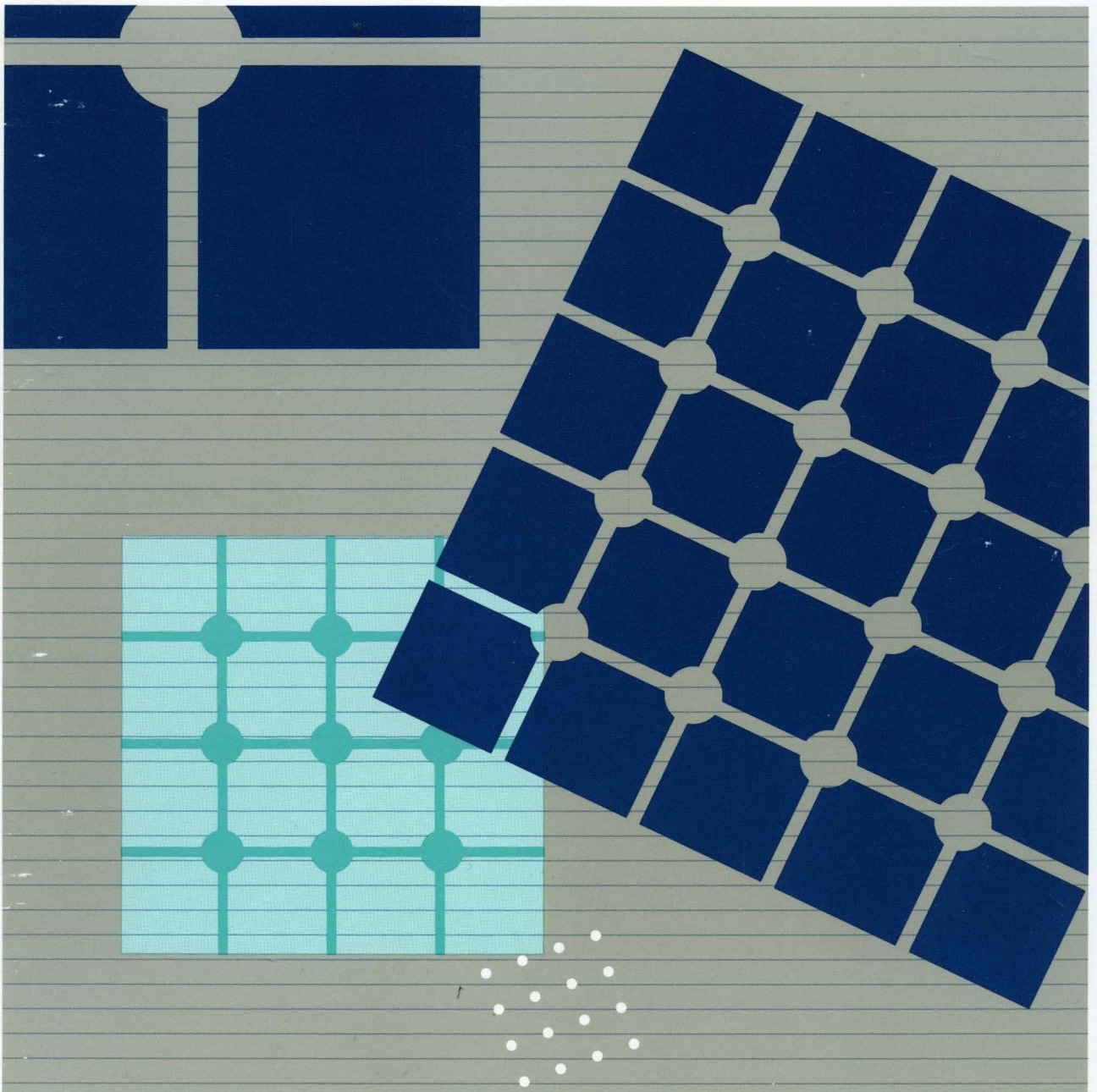




**Technical  
Reference**



5127-3710-11



Local Area Network

SC30-3383-2

**Technical  
Reference**

---

---

**Third Edition (November 1988)**

Changes are made periodically to the information herein; these changes will be incorporated in new editions of this publication.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Communication Products Information Development, Department E02, PO Box 12195, Research Triangle Park, North Carolina, U.S.A. 27709. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

---

## About This Book

This manual provides information for using IBM local area network adapters and supporting software in IBM Personal Computers and IBM PERSONAL SYSTEM/2® computers.

It is intended for use by those who are:

- Preparing programs that will use an IBM Personal Computer or Personal System/2 with IBM local area networks
- Using the IBM local area network adapters in IBM devices
- Obtaining information to understand an adapter for use in an IBM Personal Computer or IBM Personal System/2.

This manual is divided into the following chapters and appendixes.

- Chapter 1 provides an overview of the IBM interfaces provided by IBM local area network adapters and the support software available for IBM local area networks. It also includes an overview of local area networks.
- Chapter 2 describes methods of writing programs to the local area network interfaces.
- Chapter 3 describes the Command Control Blocks that can be issued to IBM adapter support software.
- Chapter 4 describes the NETBIOS<sup>1</sup> interface.
- Chapter 5 describes the NETBIOS protocol.
- Chapter 6 lists the technical references for the IBM PC Network (Broadband and Baseband) adapters.
- Chapter 7 describes the interface used by the IBM Token-Ring Network adapters.
- Chapter 8 describes the physical characteristics of the IBM Token-Ring Network adapters.
- Appendix A contains a directory of all valid commands and the related interfaces for each as well as including page references for all commands.
- Appendix B provides return codes and exception condition tables used in programming.
- Appendix C describes the program listings on the sample diskette.
- Appendix D provides information specific to the Local Area Network Support Program.
- Appendix E provides information specific to the Communications Manager of IBM Operating System/2<sup>2</sup> Extended Edition 1.1.
- After the Appendixes are a list of abbreviations and a glossary list that defines terms that are used in this manual.

---

<sup>1</sup> NETBIOS is an acronym for Network Basic Input/Output System.

<sup>2</sup> IBM Operating System/2 is a trademark of the IBM Corporation.

---

## Prerequisite Publications

Information from the following publications is necessary for use of this manual.

- The appropriate LAN adapter documentation
- *IBM Token-Ring Network Architecture Reference*, SC30-3374 \*

---

## Related Publications

This manual refers to the following IBM publications for additional information:

- *A Building Planning Guide for Communication Wiring*, G320-8059 \*
- *IBM Cabling System Planning and Installation Guide*, GA27-3361 \*
- *Using the IBM Cabling System with Communication Products*, GA27-3620 \*
- *IBM Token-Ring Network Introduction and Planning Guide*, GA27-3677 \*
- *IBM PC Network Technical Reference*
- *IBM Operating System/2 Extended Edition 1.1 Getting Started*
- *IBM Operating System/2 Extended Edition 1.1 User's Guide*
- *IBM Operating System/2 Extended Edition 1.1 System Administrator's Guide for Communications*
- *IBM Operating System/2 Extended Edition 1.1 Programming Services and Advanced Problem Determination for Communications*
- *IBM Personal Computer, Computer Language Series, Macro Assembler*
- *IBM Token-Ring Network Problem Determination Guide*, SX27-3710 \*
- *IBM Token-Ring Network Administrator's Guide*, GA27-3748 \*
- *Advanced Program-to-Program Communications for the IBM Personal Computer Programming Guide*
- *Advanced Program-to-Program Communications for the IBM Personal Computer Installation and Configuration*.

For assistance in obtaining IBM manuals, see your place of purchase. For items marked with an asterisk (\*), see your IBM representative or IBM branch office.

---

## IBM Local Area Network OEMI

The following documents make up the IBM Token-Ring Network Other Equipment Manufacture Interface.

- *IBM Cabling System Technical Interface Information*
- *IBM Local Area Network Technical Reference* (this manual)
- *IBM Token-Ring Network Architecture Reference*, SC30-3374
- *Token-Ring Access Method and Physical Layer Specification*, IEEE Std 802.5-1985.

The following documents make up the IBM PC Network Other Equipment Manufacture Interface.

- *IBM NETBIOS Application Development Guide*, S68X-2270
- *IBM PC Network Adapter Technical Reference*
- *IBM PC Network Adapter II Technical Reference*

- *IBM PC Network Adapter III/A Technical Reference*
- *IBM PC Network Adapter II - Frequency 2 Technical Reference*
- *IBM PC Network Adapter III/A - Frequency 2 Technical Reference*
- *IBM PC Network Adapter II - Frequency 3 Technical Reference*
- *IBM PC Network Adapter III/A - Frequency 3 Technical Reference*
- *IBM PC Network Baseband Adapter Technical Reference*
- *IBM PC Network Baseband Adapter/A Technical Reference*
- *IBM PC Network Baseband Extender Technical Reference*
- *IBM PC Network Translator Unit and Technical Reference*

For assistance in obtaining IBM manuals, see your place of purchase.



---

## Changes in This Edition

This is a revision of the *IBM Token-Ring Network PC Adapter Technical Reference*. It now includes new information about the following:

- Program interfaces for the Communications Manager provided with Operating System/2 Extended Edition 1.1.
- Program interfaces for the IBM PC Network
- Program interfaces for the IBM PC Network (Baseband)
- New 16/4 Mbps IBM Token-Ring Network adapters.

It also includes technical information about the network adapters for the IBM Personal System/2 and minor corrections and additions to the material in the previous editions.





---

# Contents

<b>Chapter 1. Local Area Network Overview and Interfaces</b> .....	1-1
Introduction .....	1-2
The Token-Ring Network .....	1-2
The IBM PC Network .....	1-2
The IBM PC Network (Baseband) .....	1-2
Related Software .....	1-3
OS/2 Communications Manager .....	1-3
The Dynamic Link Routine Interface and Device Driver Interface Differences .....	1-5
Why Use IEEE 802.2 or NETBIOS .....	1-6
IBM Local Area Network Adapter Interfaces .....	1-7
The Direct Interface .....	1-8
The DLC Interface .....	1-8
The APPC/PC Interface .....	1-9
NETBIOS .....	1-9
The Token-Ring Network Adapter Hardware Interface .....	1-9
Communicating on the IBM Token-Ring Network .....	1-10
Adapter Communication on a Network .....	1-10
Transmission of Data on the Token-Ring Network .....	1-10
Tokens .....	1-10
Routing Field .....	1-12
Communicating on the IBM PC Network .....	1-12
Transmission of Data on the PC Network .....	1-13
<b>Chapter 2. How to Write Programs for Local Area Networks</b> .....	2-1
Operating System/2 Extended Edition 1.1 and the Local Area Network Support Program .....	2-2
Adapter Support Software .....	2-2
Programming Conventions with DOS .....	2-3
Local Area Network Support Program (CCB1) Calling Conventions .....	2-3
Local Area Network Support Program (CCB1) Command Completion .....	2-3
Local Area Network Support Program (CCB1) Control Blocks .....	2-6
Programming Conventions with Operating System/2 Extended Edition 1.1 .....	2-6
Dynamic Link Routine Interface Programming Conventions .....	2-6
Dynamic Link Routine Interface (CCB2) Calling Conventions .....	2-8
Dynamic Link Routine Interface (CCB2) Command Completion .....	2-9
Dynamic Link Routine Interface (CCB2) Control Blocks .....	2-11
Device Driver Interface Programming Conventions .....	2-11
Device Driver Interface (CCB3) Calling Conventions .....	2-12
Device Driver Interface (CCB3) Command Completion .....	2-13
Device Driver Interface (CCB3) Control Blocks .....	2-16
Adapter Support Software Control Blocks for All CCBs .....	2-17
CCB Field Explanations .....	2-17
Addressing .....	2-23
Adapter Addresses .....	2-23
SAPs, Stations, and IDs .....	2-24
SAP Assignments .....	2-25
DLC .....	2-27
Types of Service .....	2-27
Command Sequences .....	2-27
Link Station States .....	2-29
Timers .....	2-31
Guidelines for Selecting Parameter Values .....	2-32
Transmitting, Receiving, and Buffers .....	2-34

Maximum Frame Lengths	2-35
Buffer Pools	2-35
Receive Buffers	2-36
Transmit Buffers	2-40
Calculating RAM and Work Area Usage	2-43
Calculations for the Token-Ring Network Using DOS	2-43
Calculations for the Token-Ring Network Using OS/2	2-44
Calculations for the PC Network Using DOS	2-45
Calculations for the PC Network Using OS/2	2-46
Calculations for the NETBIOS Interface	2-47
<b>Chapter 3. The Command Control Blocks</b>	<b>3-1</b>
About This Chapter	3-2
Command Descriptions	3-3
BUFFER.FREE	3-3
BUFFER.GET	3-4
DLC.CLOSE.SAP	3-5
DLC.CLOSE.STATION	3-6
DLC.CONNECT.STATION	3-7
DLC.FLOW.CONTROL	3-8
DLC.MODIFY	3-9
DLC.OPEN.SAP	3-11
DLC.OPEN.STATION	3-16
DLC.REALLOCATE	3-19
DLC.RESET	3-20
DLC.SET.TRESHOLD	3-22
DLC.STATISTICS	3-23
DIR.CLOSE.ADAPTER	3-25
DIR.CLOSE.DIRECT	3-27
DIR.DEFINE.MIF.ENVIRONMENT	3-27
DIR.INITIALIZE	3-29
DIR.INTERRUPT	3-33
DIR.MODIFY.OPEN.PARMS	3-33
DIR.OPEN.ADAPTER	3-34
DIR.OPEN.DIRECT	3-46
DIR.READ.LOG	3-47
DIR.RESTORE.OPEN.PARMS	3-50
DIR.SET.EXCEPTION.FLAGS	3-50
DIR.SET.FUNCTIONAL.ADDRESS	3-52
DIR.SET.GROUP.ADDRESS	3-52
DIR.SET.USER.APPENDAGE	3-53
DIR.STATUS	3-54
DIR.TIMER.CANCEL	3-57
DIR.TIMER.CANCEL.GROUP	3-58
DIR.TIMER.SET	3-59
PDT.TRACE.ON	3-59
PDT.TRACE.OFF	3-67
PURGE.RESOURCE	3-68
READ	3-69
READ.CANCEL	3-77
RECEIVE	3-77
RECEIVE.CANCEL	3-83
RECEIVE.MODIFY	3-84
TRANSMIT.DIR.FRAME	3-88
TRANSMIT.I.FRAME	3-88
TRANSMIT.TEST.CMD	3-89

TRANSMIT.UI.FRAME .....	3-90
TRANSMIT.XID.CMD .....	3-90
TRANSMIT.XID.RESP.FINAL .....	3-90
TRANSMIT.XID.RESP.NOT.FINAL .....	3-91
<b>Chapter 4. NETBIOS</b> .....	<b>4-1</b>
NETBIOS Overview .....	4-2
Version Information .....	4-2
The Network Control Block .....	4-3
NCB Field Explanations .....	4-3
NETBIOS Operational Parameters .....	4-7
Local Area Network Support Program Old Parameters .....	4-8
Local Area Network Support Program New Parameters .....	4-11
NETBIOS 3.0 (OS/2) .....	4-11
DLC Timers .....	4-12
NETBIOS (NCB) Calling Conventions .....	4-12
Local Area Network Support Program Calling Conventions .....	4-12
Dynamic Link Routine Interface Calling Conventions .....	4-12
Device Driver Interface Calling Conventions .....	4-13
NCB Completion .....	4-13
Wait Type Commands .....	4-13
No Wait Type Commands .....	4-13
NETBIOS Command Descriptions .....	4-14
NCB.ADD.GROUP.NAME .....	4-14
NCB.ADD.NAME .....	4-14
NCB.CALL .....	4-15
NCB.CANCEL .....	4-16
NCB.CHAIN.SEND .....	4-17
NCB.CHAIN.SEND.NO.ACK .....	4-18
NCB.DELETE.NAME .....	4-19
NCB.FIND.NAME .....	4-20
NCB.HANG.UP .....	4-21
NCB.LAN.STATUS.ALERT .....	4-22
NCB.LISTEN .....	4-22
NCB.RECEIVE .....	4-23
NCB.RECEIVE.ANY .....	4-24
NCB.RECEIVE.BROADCAST.DATAGRAM .....	4-25
NCB.RECEIVE.DATAGRAM .....	4-26
NCB.RESET .....	4-26
NCB.SEND .....	4-30
NCB.SEND.BROADCAST.DATAGRAM .....	4-31
NCB.SEND.DATAGRAM .....	4-32
NCB.SEND.NO.ACK .....	4-32
NCB.SESSION.STATUS .....	4-33
NCB.STATUS .....	4-35
NCB.TRACE .....	4-37
NCB.UNLINK .....	4-41
<b>Chapter 5. The NETBIOS Frames Protocol</b> .....	<b>5-1</b>
Overview .....	5-3
Version Information .....	5-3
Assumptions .....	5-3
Related Documents .....	5-3
Terms and Definitions .....	5-4
NETBIOS Frame Formats and Protocols .....	5-5
NETBIOS Commands .....	5-5

General Information on Remote Name Directory (RND)	5-5
New NCB Commands	5-6
Header Format Overview	5-6
NETBIOS Header	5-6
NETBIOS Frame Summary	5-8
NETBIOS Frame Formats and Descriptions	5-11
NETBIOS Protocol Examples Without RND	5-37
Name Management Examples	5-38
Remote Adapter Status Examples	5-41
Session Establishment Examples	5-44
Session Data Transfer Examples	5-47
NETBIOS Protocol Examples with RND	5-51
Session Establishment RND Examples	5-55
NETBIOS Protocol SEND.NO.ACK Examples	5-57
Session Establishment Examples	5-58
Session Data Transfer Examples	5-59
<b>Chapter 6. The PC Network and PC Network (Baseband) Adapters</b>	<b>6-1</b>
<b>Chapter 7. The Token-Ring Network Adapter Interface</b>	<b>7-1</b>
Token-Ring Network Adapters	7-3
The Adapter Interface	7-3
An Overview of the Adapter	7-3
Data Transfer	7-4
Using Interrupts	7-5
Programmed I/O (PIO) Commands	7-5
Primary Adapter Addresses	7-5
Alternate Adapter Addresses	7-6
Adapter Setup Information	7-7
BIOS/MMIO Domain	7-9
BIOS/MMIO Layout	7-10
BIOS Area	7-10
The Attachment Control Area	7-10
Communication	7-11
The Registers	7-11
Adapter Check Data	7-15
The Adapter Identification Area	7-20
Contents of the AIP	7-20
Shared RAM	7-23
Shared RAM Layout	7-23
Shared RAM Paging	7-24
Uninitialized Shared RAM	7-25
Operating the Adapter	7-26
Commands	7-29
Adapter SRB Commands	7-30
Direct Interface Commands	7-30
DIR.CLOSE.ADAPTER	7-30
DIR.INTERRUPT	7-31
DIR.MODIFY.OPEN.PARMS	7-32
DIR.OPEN.ADAPTER	7-32
DIR.READ.LOG	7-37
DIR.RESTORE.OPEN.PARMS	7-38
DIR.SET.FUNCT.ADDRESS	7-39
DIR.SET.GROUP.ADDRESS	7-39
DLC SRB Commands	7-40
DLC.CLOSE.SAP	7-41

DLC.CLOSE.STATION	7-42
DLC.CONNECT.STATION	7-43
DLC.FLOW.CONTROL	7-44
DLC.MODIFY	7-45
DLC.OPEN.SAP	7-47
DLC.OPEN.STATION	7-50
DLC.REALLOCATE	7-52
DLC.RESET	7-53
DLC.STATISTICS	7-54
Transmit Commands	7-56
Adapter-to-PC system Commands	7-58
DLC.STATUS	7-59
RECEIVED.DATA	7-60
RING.STATUS.CHANGE	7-62
TRANSMIT.DATA.REQUEST	7-62
Additional Adapter Functions	7-64
Adapter's Shared RAM	7-64
The Bridge Functions	7-64
DIR.CONFIG.BRIDGE.RAM	7-66
DIR.SET.BRIDGE.PARMS	7-67
RECEIVED.BRIDGE.DATA	7-69
POS Registers in Token-Ring Network Adapters with Micro Channel	7-75
Early Token Release	7-77
<b>Chapter 8. Token-Ring Network Adapters</b>	<b>8-1</b>
Token-Ring Network Adapters with PC I/O Bus	8-2
Power	8-2
Adapter I/O Pins	8-2
Token-Ring Network Adapters with Micro Channel	8-6
Power	8-6
Adapter I/O Pins	8-6
<b>Appendix A. Valid Commands</b>	<b>A-1</b>
DLC and Direct Interface Commands	A-1
NETBIOS Commands	A-3
<b>Appendix B. Return Codes</b>	<b>B-1</b>
CCB Return Codes Listed by Interface	B-2
CCB Return Codes Listed by Command	B-4
CCB Return Codes Cause and Action	B-12
DLC Status Codes	B-25
DLC Status Table	B-25
DLC Status Codes	B-26
Suggested Actions in Response to DLC Status	B-26
NCB Return Codes	B-27
NCB Return Codes Listed by Command	B-29
NCB Return Codes Cause and Action	B-30
Adapter Status Parameter Table	B-38
Frame Status	B-39
Exception Indications	B-40
Adapter Check	B-40
Adapter Check for CCB1	B-40
Adapter Check for CCB2	B-40
Adapter Check for CCB3	B-42
Token-Ring Network Adapter Check Reason Codes for All CCBs	B-43
PC Network Adapter Check Reason Codes for All CCBs	B-44

Network Status .....	B-44
Network Status for CCB1 .....	B-44
Network Status for CCB2 .....	B-44
Network Status for CCB3 .....	B-46
Token-Ring Network Status Codes for All CCBs .....	B-47
PC Network Status Codes for All CCBs .....	B-48
Bring-Up Errors for All CCBs .....	B-48
Bring-up Error Codes for All CCBs .....	B-48
Token-Ring Network Adapter Open Errors for All CCBs .....	B-49
Open Error Codes for All CCBs .....	B-49
Suggested Actions in Response to Open Errors .....	B-50
PC Network Adapter Open Errors for All CCBs .....	B-53
PC System Detected Errors .....	B-53
PC System Detected Errors for CCB1 .....	B-53
PC System Detected Errors for CCB2 .....	B-54
PC System Detected Errors for CCB3 .....	B-56
System Action Exceptions for OS/2 .....	B-58
System Action Exceptions for CCB2 .....	B-58
System Action Exceptions for CCB3 .....	B-60
<b>Appendix C. Local Area Network Program Listings .....</b>	<b>C-1</b>
The Sample Program Diskette .....	C-1
<b>Appendix D. Local Area Network Support Program Information .....</b>	<b>D-1</b>
16/4 Mbps Token-Ring Network Adapters .....	D-2
Token-Ring Network Adapter Failure to Open .....	D-2
PC BIOS and TIMERINT.SYS .....	D-3
Considerations When Using Two LAN Adapters .....	D-3
Using PC Network Adapters .....	D-4
NETBIOS Differences: Local Area Network Support Program Versions 1.1 and 1.0. ....	D-4
NETBIOS: Adapter Transmit/Receive Buffer Default Sizes .....	D-5
NETBIOS: Determination of Session Data Transmit Frame Size .....	D-5
Device Driver Summary .....	D-6
Token-Ring Network Device Driver Parameters .....	D-6
Token-Ring Network Adapter Addresses .....	D-6
PC Network Device Driver Parameters .....	D-9
PC Network Adapter Addresses .....	D-9
NETBIOS Device Driver Parameters .....	D-10
NETBIOS Device Driver Keyword Old Parameter Summary .....	D-10
NETBIOS Device Driver Keyword New Parameter Summary .....	D-11
The Local Area Network Support Program Interrupt Arbitrator .....	D-16
Registration Process Overview .....	D-17
<b>Appendix E. Operating System/2 Extended Edition 1.1 Information .....</b>	<b>E-1</b>
CONFIG.SYS COMMANDS .....	E-2
Adapter Parameters .....	E-3
DLC Parameters .....	E-6
NETBIOS 3.0 Parameters .....	E-9
The Operating System/2 Trace Facility .....	E-10
Trace Code Definition .....	E-11
Trace Entry Format .....	E-11
Operating System/2 NETBIOS Trace Facility .....	E-15
<b>List of Abbreviations .....</b>	<b>X-1</b>
<b>Glossary .....</b>	<b>X-3</b>

**Index** ..... X-7





# Figures

1-1.	Communications Manager Structure	1-4
1-2.	DOS Network Interfaces	1-7
1-3.	Operating System/2 Extended Edition 1.1 Communications Manager Interfaces	1-8
1-4.	Token Format	1-10
1-5.	Token-Ring Network Frame Format	1-11
1-6.	PC Network Frame Format	1-13
2-1.	SAPs and Link Stations	2-26
2-2.	Receive Buffer Formats	2-37
2-3.	Transmit Buffers	2-42
2-4.	MAC frame	2-42
2-5.	Non-MAC I frame	2-42
2-6.	Other non-MAC frame	2-43
7-1.	PC Address Bits	7-9
7-2.	Attachment Control Area Registers	7-18
7-3.	Attachment Control Area Access Summary	7-19
7-4.	Attachment Control Area Addressing Summary	7-19
7-5.	Shared RAM Paging after Initialization	7-25
7-6.	Uninitialized Shared RAM	7-26
8-1.	Adapter Cable D Connector	8-10
E-1.	CCB Trace Entry (Trace Codes X'00' - X'07')	E-12
E-2.	CCB Completion Trace Entry (Trace Code X'08')	E-12
E-3.	Receive Completion Trace Entry (Trace Code X'09')	E-13
E-4.	Status/Exception Completion Trace Entry (Trace Code X'0A')	E-13
E-5.	Interrupt Received Trace Entry (Trace Codes X'0B' - X'0C')	E-14
E-6.	Interrupt Error Trace Entry (Trace Codes X'0D')	E-15



---

# Chapter 1. Local Area Network Overview and Interfaces

Introduction	1-2
The Token-Ring Network	1-2
The IBM PC Network	1-2
The IBM PC Network (Baseband)	1-2
Related Software	1-3
OS/2 Communications Manager	1-3
The Dynamic Link Routine Interface and Device Driver Interface Differences	1-5
Why Use IEEE 802.2 or NETBIOS	1-6
IBM Local Area Network Adapter Interfaces	1-7
The Direct Interface	1-8
The DLC Interface	1-8
The APPC/PC Interface	1-9
NETBIOS	1-9
The Token-Ring Network Adapter Hardware Interface	1-9
Communicating on the IBM Token-Ring Network	1-10
Adapter Communication on a Network	1-10
Transmission of Data on the Token-Ring Network	1-10
Tokens	1-10
Frames	1-11
Routing Field	1-12
Communicating on the IBM PC Network	1-12
Transmission of Data on the PC Network	1-13
Frames	1-13

---

## Introduction

The following local area networks use the interfaces described in this manual:

- The IBM Token-Ring Network
- The IBM PC Network
- The IBM PC Network (Baseband).

IBM Personal Computers and PERSONAL SYSTEM/2® computers, hereafter referred to as *PC systems*, may use these networks with adapters and related software. This document provides technical information for the Token-Ring Network adapters and the interfaces used by all IBM network adapters on local area networks.

---

## The Token-Ring Network

The IBM Token-Ring Network, a token-ring, star-wired local area network, can accommodate on one ring up to 260 attaching devices (printers, processors, controllers). Bridges can connect multiple rings together to form a network of more than 260 devices. These attaching devices connect to one another by a series of cables, access units, and special adapters installed in the attaching devices.

Application programs running in each workstation (such as an IBM Personal Computer or Personal System/2) can direct the adapter to become a part of the ring. This manual describes the commands used by programs to control the Token-Ring Network adapter's activity on the network. Refer to the *IBM Token-Ring Network Introduction and Planning Guide* for more information about the network.

---

## The IBM PC Network

The PC Network is a bus attached, broadband, local area network that can accommodate up to 72 attaching devices with IBM components. See your dealer or place of purchase for the appropriate adapter for your PC system.

---

## The IBM PC Network (Baseband)

The PC Network (Baseband) is a bus attached, baseband, local area network that can accommodate up to 80 attaching devices with IBM components. See your dealer or place of purchase for the appropriate adapter for your PC system.

---

## Related Software

The software listed below provides the interface to allow communication on the networks using local area network adapters.

- For use with IBM Disk Operating System (DOS)
  - The adapter support software provided with the Token-Ring Network PC Adapter and Token-Ring Network PC Adapter II.
  - The Token-Ring Network NETBIOS<sup>1</sup> Program
  - The Local Area Network Support Program
  - Advanced Program-to-Program Communications for the IBM PC (APPC/PC)
- For use with IBM OPERATING SYSTEM/2 (OS/2)<sup>2</sup>
  - The Communications Manager provided with Operating System/2 Extended Edition 1.1

These interfaces are used by application programs to communicate on a local area network.

**Note:** In this book the term “adapter support software” means the software that controls the adapter and can be any of the following products or their equivalent:

- The adapter support software (TOKREUI.COM) provided with the Token-Ring Network PC Adapter and Token-Ring Network PC Adapter II
- The adapter support software provided with the Local Area Network Support Program
- The adapter support software provided with the Communications Manager provided with Operating System/2 Extended Edition 1.1.

---

## OS/2 Communications Manager

Communications Manager is a component of the Operating System/2 Extended Edition 1.1. It provides comprehensive communication capabilities for a variety of interconnections. Functions that were previously available only in various communications programs for DOS are now combined with the functions of multi-tasking and expanded memory support. Communications Manager enables users to connect to a range of computers, including IBM and non-IBM host systems and other personal computers. In addition, multiple connections can be active concurrently, giving users access to information from wherever it is located.

Communications Manager supports a wide range of communication capabilities that include:

- 3270 terminal emulation
- ASCII terminal emulation
- IBM Server-Requester Programming Interface (SRPI)
- IBM Systems Network Architecture (SNA) Advanced Program-to-Program Communication (APPC)
- IBM Asynchronous Communications Device Interface (ACDI)
- IEEE 802.2 Application Program Interface (API) (Direct interface and DLC interface)
- IBM NETBIOS API
- Emulator High-Level Language Application Programming Interface (EHLLAPI).

---

<sup>1</sup> NETBIOS is an acronym for Network Basic Input/Output System.

<sup>2</sup> IBM Operating System/2 and OS/2 are trademarks of the IBM Corporation.

Since OS/2 provides multi-tasking capability, the various communications options can usually run concurrently. In many cases, this eliminates the need to load and unload programs to communicate with different systems.

The following is a graphic representation of Communications Manager. The dark lines above some of the boxes indicate Application Programming Interfaces (APIs) provided by Communications Manager. Only the IEEE 802.2 API and the NETBIOS API are described in this book.

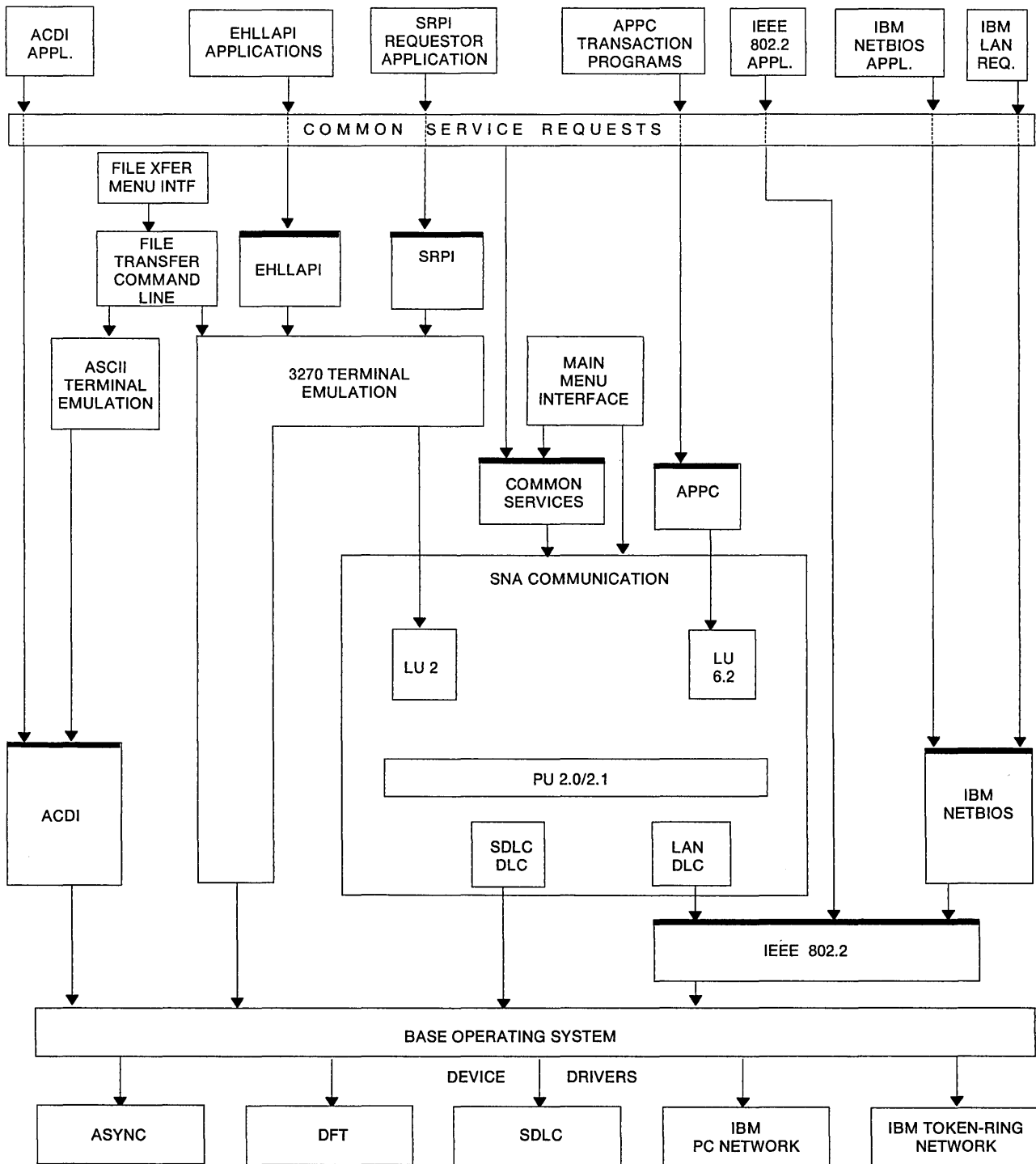


Figure 1-1. Communications Manager Structure

The IEEE 802.2 API provided by the Communications Manager supports both the direct and DLC interfaces described in this book. Application programs may choose to use the direct interface only or may use both the direct and DLC interfaces.

There are two methods available to access the IEEE 802.2 API. One method is by using the Dynamic Link Routine Interface (DLR). The other method is by using the Device Driver Interface (DD), where the adapter support software interfaces directly to the LAN device drivers provided by Communications Manager.

Application programs communicate across the IEEE 802.2 API with Command Control Blocks (CCBs). The CCBs used by the Dynamic Link Routine Interface are described by CCB2. The CCBs used by the Device Driver Interface are described by CCB3. For more information on CCBs, see Chapter 2, "How to Write Programs for Local Area Networks."

The NETBIOS API provided by Communications Manager is accessed by a Dynamic Link Routine Interface. Application programs communicate across the NETBIOS API with Network Control Blocks (NCBs). For more information on NETBIOS and the NCB commands, see Chapter 4, "NETBIOS."

To use the IEEE 802.2 API or the NETBIOS API, Communications Manager must be installed and configured. See *IBM Operating System/2 Extended Edition 1.1 Getting Started* for information on installing Communications Manager and the LAN device drivers. See *IBM Operating System/2 Extended Edition 1.1 System Administrator's Guide for Communications* for information on configuring Communications Manager.

## The Dynamic Link Routine Interface and Device Driver Interface Differences

Two levels of OS/2 interfaces exist for both the adapter support software and NETBIOS. An application program can use an interface provided by a Dynamic Link Routine Interface or one provided by a Device Driver Interface. An application program may use either type of OS/2 interface, but cannot use both interfaces at the same time if it wishes to be considered a single application. Resources provided to and resources obtained from one of the OS/2 interfaces cannot be used at the other OS/2 interface.

An application program can easily use one of the Dynamic Link Routine Interfaces by making the appropriate external references to an OS/2 Dynamic Link Routine Interface. In order for an application program to use one of the DD interfaces the application program itself must be a device driver or have a device driver as one of its components. The application program device driver must be setup to support intercommunication between device drivers. In doing this, the application program device driver can be called by the adapter support software and NETBIOS for posting of events.

Several factors may be involved in determining the best OS/2 interface for your programming needs. Some of these factors are:

- Programming language to be used for developing your application program.
  - Device driver components of application programs must be written in Assembler as registers must be accessed and processed. In addition, flags must be tested for error conditions.
- Performance
  - The DLR interfaces use semaphores and create threads for application programs in order to post events, thus task switches are involved when using the interface. When events occur that affect the application program (for example, command completions and network status changes) the application program can respond to the event after one of its threads is dispatched by a semaphore being cleared.
  - The Device Driver Interface calls the application program's device driver to post events. Thus when an event occurs that affects the application program, the application program is notified without delay and can respond immediately to the event without a task switch.



- Complexity
  - The Dynamic Link Routine Interface manages asynchronous events and allows the application program to process event information at its convenience.
  - While the Device Driver Interface provides better performance to the application program, it also requires the application program to share in some of the responsibilities associated with processing asynchronous events. When events occur, the adapter support software and NETBIOS will call the application program to post the event. No event information will be queued for later notification or retrieval by the application program. In addition, the application program device driver will be responsible for ensuring that data structures and buffers passed to the adapter support software and NETBIOS are located in valid memory segments and are locked so they will not be moved or swapped by OS/2.

---

## Why Use IEEE 802.2 or NETBIOS

There are various Application Programming Interfaces (API) supported. The interfaces will be discussed as two APIs and these are as follows:

1. IEEE 802.2 API
  - DIRECT Interface
  - DLC Interface
2. NETBIOS API
  - NETBIOS Interface

The determination of which interface to use by a programmer developing an application program to access the network depends on many things. Some of the things that need to be considered are:

- Usability of the interface

The NETBIOS interface provides a simple interface for the application program and does not require the application program to understand Data Link Control.
- The performance required by your application program.

The IEEE 802.2 interfaces provide better performance, but require the application programs to be significantly more complex. Performance advantages can be up to two times that of NETBIOS based on the amount of data transferred between the application programs.
- The interfaces used by other applications with which your application may interact.

# IBM Local Area Network Adapter Interfaces

The Local Area Network Support Program and Communications Manager with Operating System/2 Extended Edition 1.1 provide both IEEE 802.2 and NETBIOS interfaces. These programs provide at least two levels of entry to IBM networks: the data link control (DLC) interface, and the direct interface.

The adapter support software provides an application program interface to the adapter. The application program can issue Command Control Blocks (CCBs) to the adapter support software to interface with the adapter. This frees the application program from the burden of interacting directly with the adapter. For information on CCBs and communicating with the adapter support software, see Chapter 2, "How to Write Programs for Local Area Networks," and Chapter 3, "The Command Control Blocks." For information about interacting directly with the adapter, see Chapter 7, "The Token-Ring Network Adapter Interface." See Figure 1-2 for an illustration showing the relationship of application programs, the adapter support software, and the network adapter when using DOS adapter support software. See Figure 1-3 on page 1-8 for an illustration showing the relationship of application programs, the adapter support software, and different adapters when using Operating System/2 Extended Edition 1.1.

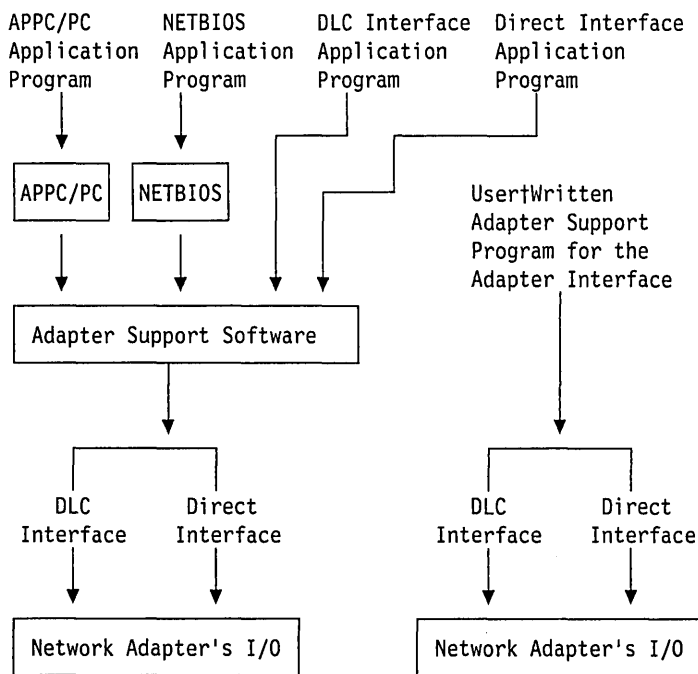


Figure 1-2. DOS Network Interfaces

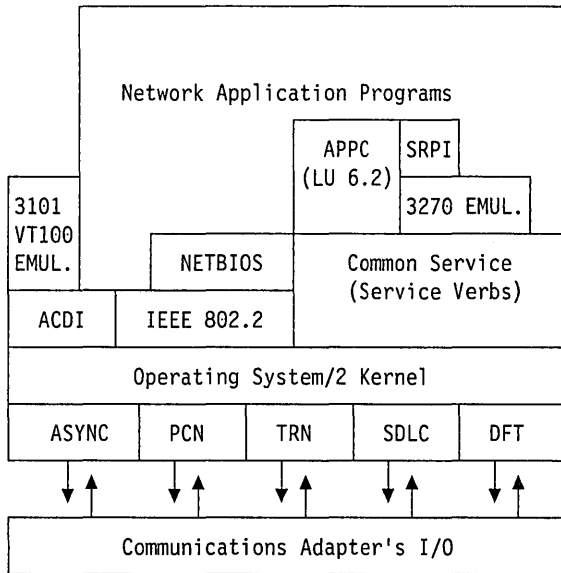


Figure 1-3. Operating System/2 Extended Edition 1.1 Communications Manager Interfaces

If adapter support software is used, a network application program assembles a control block containing a command and related information for the adapter. Control is passed to the adapter support software and the application program awaits the results.

Appendix A, "Valid Commands," contains a directory of all commands, their related interfaces, and the page where a description of each is located. The functions of the DLC interfaces of the adapter and the adapter support software are compatible with the service specifications of the Institute of Electrical and Electronic Engineers Inc. (IEEE) 802.2 Logical Link Control. Detailed information on these interfaces is contained in Chapter 7, "The Token-Ring Network Adapter Interface."

Each of the following interfaces provides a means of communicating with the adapter. Depending on which you choose, the code you provide will share the responsibility for control of the adapter with the adapter support software.

## The Direct Interface

The direct interface allows control of the adapter using standard control blocks and parameters.

This interface provides the ability to open and close an adapter, obtain error status, and set addresses. It also permits transmission and reception of frames directly without LLC protocol assistance.

Detailed use of this interface is described in Chapter 2, "How to Write Programs for Local Area Networks."

## The DLC Interface

This interface, together with the direct interface, provides an interface to applications using the Logical Link Control sublayer of Data Link Control protocol. The DLC protocol consists of LLC sublayer and the MAC layer protocol. The interface may be used in two ways.

- For IEEE type 1 communication: connectionless communication between devices providing no guarantee of delivery (through the DLC SAP interface).
- For IEEE type 2 communication: connection-oriented services (through the DLC station interface), providing point-to-point connectivity with guaranteed delivery and retry.

Much of the communication overhead function is provided by the adapter and adapter support software, which permit less complex application programming.

Detailed use of this interface is described in Chapter 2, "How to Write Programs for Local Area Networks." See the *IBM Token-Ring Network Architecture Reference* for more about communication using DLC.

## The APPC/PC Interface

The APPC/PC Program is a product that uses the adapter support software. Refer to *Advanced Program-to-Program Communications for IBM Personal Computer Installation and Configuration Guide*. How to design and write APPC/PC transaction programs is explained in *Advanced Program-to-Program Communications for the IBM Personal Computer Programming Guide*.

**Note:** The APPC/PC support for Operating System/2 is provided by the Communications Manager.

## NETBIOS

The IBM NETBIOS API provides a programming interface to the LAN so that an application program can have LAN communication without knowledge and responsibility of the Data Link Control (DLC). NETBIOS provides the necessary DLC communications for the application program. Nodes are identified on the LAN by NETBIOS names, and two types of data transfer are supported by NETBIOS. Session support provides guaranteed delivery of the data, while Datagram support does not guarantee delivery.

NETBIOS application programs require that NETBIOS and the adapter support software be used on the Token-Ring Network or the PC Network.\* Using NETBIOS is explained in Chapter 4, "NETBIOS."

## The Token-Ring Network Adapter Hardware Interface

The adapter interface provides access to the Token-Ring Network through the three interfaces mentioned earlier:

- The direct interface
- The DLC Service Access Point (SAP) interface
- The DLC Station interface.

If a program is written to use the adapter interfaces, that program will replace the adapter support software and will be totally responsible for interacting with the adapter.

Detailed information about operating the adapter is contained in Chapter 7, "The Token-Ring Network Adapter Interface."

---

\* The only exception to the requirement to use the adapter support software is when the *original* PC Network Adapter or the PC Network Protocol Driver are used.

---

## Communicating on the IBM Token-Ring Network

The IBM Token-Ring Network uses a Media Access Control (MAC) protocol to control data flow, monitor ring conditions, and encapsulate and route data for devices attached to the physical ring. The token and frame are involved in this Token-Ring Network communication. This protocol is not totally explained in this manual. See the *IBM Token-Ring Network Architecture Reference* for more detail about this protocol.

### Adapter Communication on a Network

To communicate on a Token-Ring Network, an adapter obtains a token and changes the token to a frame containing control information and data. The frame is then passed around the ring, and each adapter on the ring monitors the frame and retransmits it. When the frame reaches the adapter to which the data is addressed, the adapter copies the frame as it is retransmitting it and indicates in the frame that the data has been received. The frame continues in the same direction around the ring until it returns to the adapter that sent the frame. The sending adapter removes the frame from the ring, replacing it with a token.

### Transmission of Data on the Token-Ring Network

- When the adapter is instructed to “open” following an instruction to “initialize,” it goes through a series of diagnostics before attempting to attach to the network. The adapter verifies:
  - The correct operation of the adapter
  - That a cable between the attaching device and the 8228 Multi-Station Access Unit exists
  - That a test message sent along the lobe cable (the cable between the attaching device and the 8228) can pass through the cable and return unchanged.

**Note:** Some possible error results may be delayed until after the “open” has completed.

- The “open” command tells the adapter to send a direct current signal to the 8228 Access Unit to which it is attached. This causes the attaching device to become an active part of the ring: receiving, monitoring, and retransmitting ring traffic.
- The adapter places data on the ring by changing a token into a frame that will pass around the ring in place of the token.

### Tokens

A token is a unique sequence of bits in 3 bytes transmitted on the ring. The sequence is a start delimiter byte followed by an access control field of 1 byte and ending with an end delimiter byte. Each byte is 8 bits in length without parity. Both the start delimiter (SD) and the end delimiter (ED) are unique bit arrangements recognized by all adapters compatible with the IEEE 802.5 standard. The access control (AC) field controls access to the ring.

SD 1 Byte	AC 1 Byte	ED 1 Byte
--------------	--------------	--------------

Figure 1-4. Token Format

The same 3 bytes are included in a frame. However, a bit is set on in the AC byte to indicate a frame rather than a token. Additional control and information bytes are included between the AC byte and the ED byte.

# Frames

A frame consists of:

- A start delimiter (SD) byte
- An access control (AC) byte with the frame bit on
- A frame control (FC) byte
- Destination address bytes
- Source address bytes
- Optional routing field bytes
- Optional DLC header field
- Optional Information (data) bytes
- The frame check sequence (FCS) byte
- The end delimiter (ED) byte
- The frame status (FS) byte.

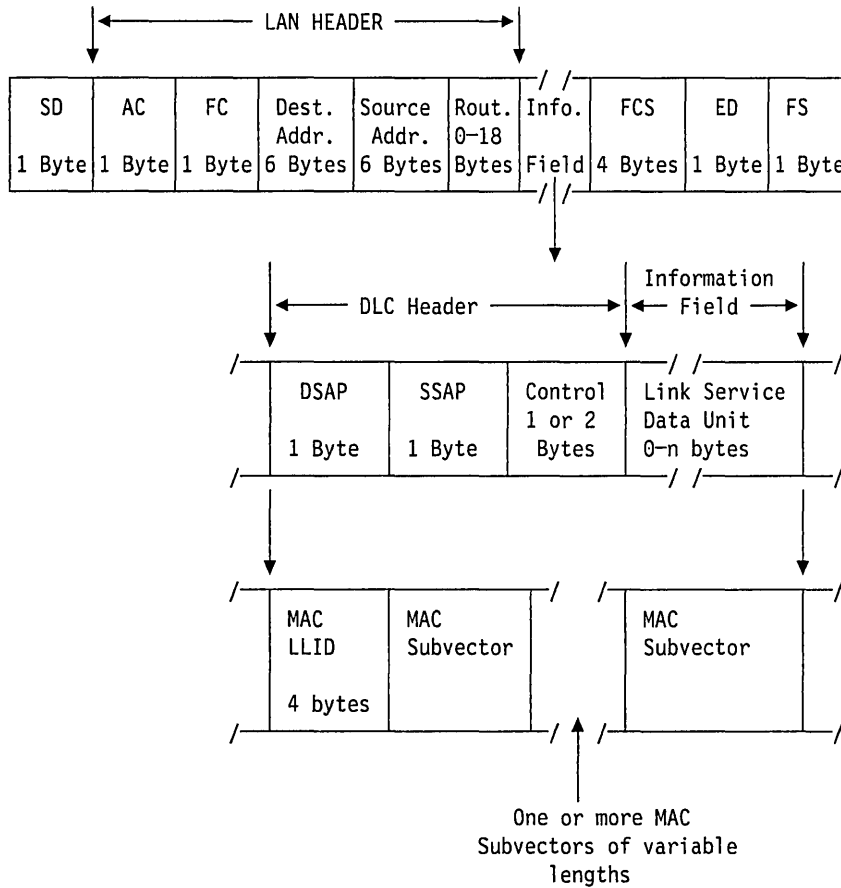


Figure 1-5. Token-Ring Network Frame Format. Bits are transmitted in bytes, most significant bit (bit 7) first.

The physical, or LAN, header consists of the AC byte, the 1-byte frame control (FC) field, 6 bytes of destination address, 6 bytes of source address, and from zero to 18 bytes of routing information. This is followed by the information field. Finally, the physical trailer (PT) is included, consisting of 4 bytes of frame check sequence (FCS) field, the ED byte, and the FS byte.

The frame may be one of two types:

- MAC frame
- Non-MAC frame.

Media Access Control (MAC) frames contain information about the status of an adapter or the ring itself.

Certain MAC frames may be received by the adapter and provided to the application program at the direct interface. Some MAC frames may be sent to the adapter for transmission on the ring using the direct interface of either the adapter support software or the adapter.

Some Non-MAC frames contain data and messages that users transmit to one another.

Some Non-MAC frames contain LLC protocol-only information transmitted by the adapter. These frames are used for type 2 protocol support.

The 2 most significant bits of the FC byte define the frame type. The types are:

**B'00'** MAC frame  
**B'01'** LLC frame (non-MAC)  
**B'10'** reserved  
**B'11'** reserved

## Routing Field

The routing field is used by bridges to forward frames to their destination. This field is optional and is needed if the frame needs to be forwarded by bridges to other rings. Further information on how to use this field and signify that it is present is given in the *IBM Token-Ring Network Architecture Reference*.

Frame formats are described in detail in the *IBM Token-Ring Network Architecture Reference*. Bit sequences in that manual and possibly other documentation may differ from IBM PC format in that the most significant bit of a byte is designated **7** for the IBM PC system and may be called **0** elsewhere. Only the representation differs. The byte's content is not altered.

---

## Communicating on the IBM PC Network

The PC Network uses carrier-sense multiple-access with collision detect (CSMA/CD) protocols to control data flow. There is no token, and there is no distinction between MAC and non-MAC frames in the sense used in the discussion of IBM Token-Ring protocols. Frames used to control the operation of the network are LLC Type 1 UI frames exchanged between management SAPs (X'E4' and X'F4').

IBM PC Network adapters implement the protocols involved in transmitting and receiving frames, and in recognizing the condition of the network. They also have RAM used to buffer frame data: its location within the PC system's memory space is determined by switches on the card. The adapter support software in the PC system provides access to the network through the following interfaces:

- The direct interface
- The DLC Service Access Point (SAP) interface
- The DLC Station interface

## Transmission of Data on the PC Network

When the adapter support software is instructed to “open” following an instruction to “initialize,” it goes through a series of diagnostics before attempting to attach to the network. The adapter support software verifies:

- The correct operation of the adapter
- That a frame can be sent on the network
- That the specific address being used for the adapter is unique on the network.

Before placing data on the network the adapter checks to see whether another adapter is already transmitting and, if so, waits for a random time interval before checking again. Once the adapter determines that the network is available it will transmit the frame data. It then monitors the network in case another adapter transmitted at the same time. If a collision is detected the adapter will wait for a random period of time before retransmitting the frame.

### Frames

A frame on the IBM PC Network consists of the following fields:

- Start delimiter (SD) - 1 byte
- Destination address - 6 bytes
- Source address - 6 bytes
- Zeroes - 2 bytes
- Optional routing field - up to 18 bytes
- Optional DLC header field - 3-4 bytes
- Optional Information (data) bytes
- The frame check sequence (FCS) - 4 bytes
- Pad characters (flags) if needed to reach the minimum frame length.

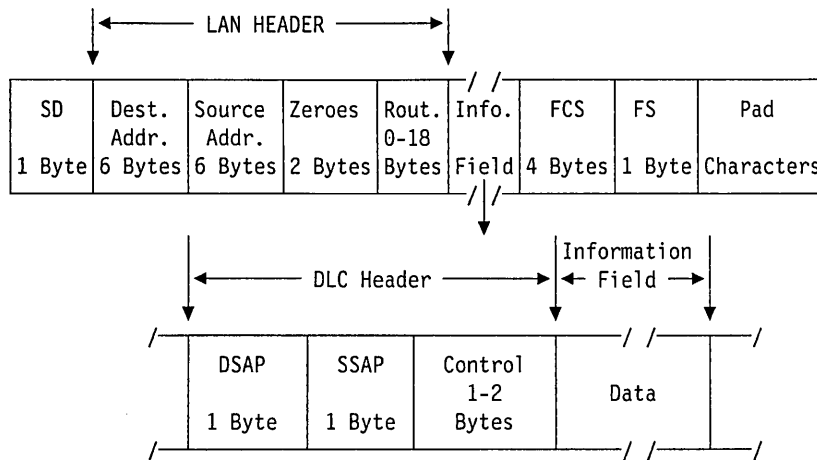


Figure 1-6. PC Network Frame Format. Bits are transmitted in bytes, most significant bit (bit 7) first.

The physical, or LAN, header consists of the destination and source addresses, two zero bytes, and from zero to 18 bytes of routing information. This is followed by the user-provided data. The LAN header on the PC Network is different from that on the Token-Ring Network, but in order to provide compatibility for application programs the difference is not reflected at the interface to the adapter support software. In cases where the application program provides the access control and frame control bytes (used for the Token-Ring Network) the adapter support software will simply omit the bytes from the transmitted frame and insert two bytes of zeroes following the source address. Note, however, that the adapter support software will check to make sure that the frame control byte specifies an LLC, not a MAC, frame. The



adapter support software will transmit the address fields with the bit sequence expected on the IBM PC Network, not that expected on the IBM Token-Ring Network.

The formats of the Routing Information, the DLC header, and the information field are identical on both networks: these formats are described in detail in the *IBM Token-Ring Network Architecture Reference*. Bit sequences in that manual and possibly other documentation may differ from IBM PC format in that the most significant bit of a byte is designated **7** for the IBM PC system and may be called **0** elsewhere. Only the representation differs. The byte's content is not altered.

---

## Chapter 2. How to Write Programs for Local Area Networks

Operating System/2 Extended Edition 1.1 and the Local Area Network Support Program	2-2
Adapter Support Software	2-2
Programming Conventions with DOS	2-3
Local Area Network Support Program (CCB1) Calling Conventions	2-3
Local Area Network Support Program (CCB1) Command Completion	2-3
Appendages	2-4
Local Area Network Support Program (CCB1) Control Blocks	2-6
Programming Conventions with Operating System/2 Extended Edition 1.1	2-6
Dynamic Link Routine Interface Programming Conventions	2-6
Execution of Multiple Commands	2-6
User's Data Segments Restrictions	2-7
User's Data Segments Guidelines	2-7
Dynamic Link Routine Interface (CCB2) Calling Conventions	2-8
Dynamic Link Routine Interface (CCB2) Command Completion	2-9
User Notification Flags	2-9
Posting of Events	2-10
Dynamic Link Routine Interface (CCB2) Control Blocks	2-11
Device Driver Interface Programming Conventions	2-11
User's Data Segments Restrictions	2-11
User's Data Segments Guide Lines	2-12
Device Driver Interface (CCB3) Calling Conventions	2-12
Device Driver Interface (CCB3) Command Completion	2-13
Posting of Events	2-13
Device Driver Interface (CCB3) Control Blocks	2-16
Adapter Support Software Control Blocks for All CCBs	2-17
CCB Field Explanations	2-17
Addressing	2-23
Adapter Addresses	2-23
SAPs, Stations, and IDs	2-24
SAP Assignments	2-25
DLC	2-27
Types of Service	2-27
Command Sequences	2-27
Link Station States	2-29
Timers	2-31
Guidelines for Selecting Parameter Values	2-32
Transmitting, Receiving, and Buffers	2-34
Maximum Frame Lengths	2-35
Buffer Pools	2-35
Receive Buffers	2-36
Receive Buffer Formats	2-36
Buffer One: Option = Not Contiguous MAC/DATA	2-37
Explanation of Certain Buffer Fields	2-39
Transmit Buffers	2-40
Transmit Buffer Formats	2-41
Calculating RAM and Work Area Usage	2-43
Calculations for the Token-Ring Network Using DOS	2-43
Calculations for the Token-Ring Network Using OS/2	2-44
Calculations for the PC Network Using DOS	2-45
Calculations for the PC Network Using OS/2	2-46
Calculations for the NETBIOS Interface	2-47

---

## Operating System/2 Extended Edition 1.1 and the Local Area Network Support Program

This chapter, as well as the rest of this Technical Reference, includes discussions about the LAN interfaces provided by the Operating System/2 Extended Edition 1.1 and the Local Area Network Support Program. These interfaces are very similar in that most of the commands are the same and the structure of the commands is very similar. Where it is possible to discuss one command that can be used for all interfaces, it will be done using the term CCB only. Where it is impossible to talk about them as one, they will be discussed separately as CCB1, CCB2, and CCB3. The major differences in the Command Control Blocks will be explained in this chapter and the different control block structures will be referred to as follows:

**CCB1** The Command Control Block for the IEEE 802.2 adapter support software provided with the original Token-Ring Network PC Adapter, Token-Ring Network PC Adapter II, and the IEEE 802.2 adapter support software supplied with the Local Area Network Support Program.

**CCB2** The Command Control Block for the Dynamic Link Routine Interface provided with the Communications Manager of Operating System/2 Extended Edition 1.1.

**CCB3** The Command Control Block for the Device Driver Interface provided with the Communications Manager of Operating System/2 Extended Edition 1.1.

**CCB** When CCB is used without a qualifier, the information refers to all three interfaces.

**Note:** The Local Area Network Support Program will not operate with Operating System/2.

---

## Adapter Support Software

Adapter support software is supplied with the original Token-Ring Network PC Adapter and the Token-Ring Network PC Adapter II. The Local Area Network Support Program and Operating System/2 Extended Edition 1.1 provide equivalent adapter support software if any of the other Token-Ring Network adapters or any of the PC Network adapters are used. If adapter support software like this is used, it must be loaded into PC system memory. The application program interface of this adapter support software provides three levels of entry to the Token-Ring Network:

- The DLC SAP interface
- The Direct Interface
- The DLC Station Interface.

If NETBIOS is also loaded into memory, an interface to NETBIOS commands is provided. See Chapter 4, "NETBIOS," for more about NETBIOS.

The adapter support software interacts with the adapter. It allows an application program to use the adapter by providing control blocks and calling the adapter support software. This frees the application program from the burden of interacting with the adapter.

---

# Programming Conventions with DOS

## Local Area Network Support Program (CCB1) Calling Conventions

To issue a request to the Local Area Network Support Program (or other network adapter support software for DOS) a network application program assembles a control block containing a command and related information for the adapter. The application program then puts the PC system's main memory address of this control block into the ES (extra segment) and BX (base) registers. At this point, the application program issues a X'5C' software interrupt. The adapter support software responds to the X'5C' interrupt by processing the control block. While processing the CCB, the adapter support software enables interrupts.

## Local Area Network Support Program (CCB1) Command Completion

CCB\_RETCODE is initially set to a value of X'FF' and once any immediate command processing is accomplished, control is returned to the application program. At that point the application program may continue with other processing, but may not disturb the CCB or associated data. (The CCB\_RETCODE is still X'FF'.) When the command is completed, the adapter support software will set the return code in both the AL register and the CCB\_RETCODE field and check the CCB\_CMD\_CMPL field. The CCB\_CMD\_CMPL is used to give the adapter support software the address of an application command completion appendage.

- If the CCB\_CMD\_CMPL field is not zero, the adapter support software transfers control to the application program at the address provided. The application program continues with the command completion appendage and returns to the adapter support software when completed.

Upon entry, the command completion appendage may obtain the final return code from either the AL register or the CCB\_RETCODE field.

- If the CCB\_CMD\_CMPL field contains X'00000000', the application program has not supplied a command completion appendage and the adapter support software performs no further action for this CCB and does not interrupt the application program. In that case, the application program must monitor the CCB\_RETCODE for a change from X'FF', indicating the adapter has completed the command and updated the return code.

If the adapter support software immediately determines that the adapter cannot execute the command, it sets the CCB\_RETCODE field with the error code.

There are some commands that execute entirely in the PC system and do not use the adapter hardware. When this is the case:

- The completion code is set when the adapter support software returns from the interrupt that initiated the command.
- If the command completion appendage is defined, it will be given control before the adapter support software returns from the interrupt.

This is an exception and is explained with the command descriptions to which it applies.

## Appendages

User-supplied appendages provide exit points from the adapter support software. These appendages should be short subroutines that may improve the application program's ability to handle information or events. See Chapter 4, "NETBIOS" on page 4-1 for routines used with NETBIOS.

To ensure the integrity of the system, the appendages should have the following characteristics:

1. The amount of code executed should be limited, as this routine is an I/O appendage. The appendage is used because a point has been reached where information should be saved for subsequent use.
2. When the appendage is entered, the keyboard and DOS timer are disabled, and no more interrupts can be serviced from this adapter until the appendage is completed.
3. When control is passed to the appendage, interrupts are disabled and it appears to the appendage that the appendage was entered via an 8086 INT instruction. The stack is established such that an 8086 IRET instruction will properly return control and restore flags.

When appendage processing is complete, the appendage code must execute the 8086 IRET as the last instruction.

The adapter support software will have saved all registers on the stack prior to giving control to the appendage. Twenty four bytes of the stack will have been used by the adapter support software when processing the adapter interrupt. When the appendage is entered, there are 242 bytes of stack space available.

4. Execution of the IRET by the appendage will return control to the adapter support software at the point at which it had transferred control to the appendage. The adapter support software will restore all registers and return control to the program that was originally interrupted.

Upon entry to the appendages:

- The CX register contains the adapter number.
- The CS register points to the appendage code (current segment).
- The SS and SP registers define the current stack.
- Other specific appendage descriptions define other registers.

The types of user appendages are:

### Command completion appendage

A per-command exit that allows asynchronous command completion. The application program may provide several Command Completion Appendages and selectively point to a specific one in each CCB.

The entry point is indicated by the address in the CCB\_CMD\_CMPL field (which should not be X'00000000', indicating no appendage) of the related CCB.

The address of the CCB that the adapter completed will be in registers ES and BX. The return code will be in CCB\_RETCODE and the AL register (AH = X'00').

### Data received appendage

This appendage is defined in the RECEIVED\_DATA field of the parameter table of the RECEIVE command. The address of the CCB is placed in registers DS and SI. The address of the first receive data buffer is placed in registers ES and BX.

### Exception or Status conditions

These appendages are a set of exit points that allow the adapter support software to report hardware and software error conditions and certain status information to the user. When any exception state occurs, all outstanding adapter commands will have the CCB\_RETCODE field of

their CCBs set for the appropriate reason, and will be queued and passed to the exception appendage. The command completion appendage will not be taken. See the CCB\_POINTER field description on page 2-18 for more about queues.

#### **PC-Detected error appendage**

This appendage is defined in the PC\_ERROR\_EXIT field of the CCB for a DIR.INITIALIZE command and a DIR.MODIFY.OPEN.PARMS command, or in the PC\_ERROR\_EXIT field of the DIRECT\_PARMS table of a DIR.OPEN.ADAPTER command.

The adapter support software passes parameters to the appendage on entry. Register CX contains the adapter number. Register AX contains the error code. Refer to "PC System Detected Errors" on page B-53 for the code meanings.

#### **Network status appendage**

This appendage is defined in the NETW\_STATUS\_EXIT field of the CCB for a DIR.INITIALIZE command and a DIR.MODIFY.OPEN.PARMS command, or in the NETW\_STATUS\_EXIT field of the DIRECT\_PARMS table of a DIR.OPEN.ADAPTER command.

The adapter support software passes parameters to the appendage on entry. Register CX contains the adapter number. Register AX contains the network status. Refer to "Network Status" on page B-44 for the code meanings.

#### **Adapter check appendage**

This appendage is defined in the ADAPTER\_CHECK\_EXIT field of a DIR.INITIALIZE command and a DIR.MODIFY.OPEN.PARMS command, or in the ADAPTER\_CHECK\_EXIT field of the DIRECT\_PARMS table of a DIR.OPEN.ADAPTER command. See pages 3-29, 3-33, and 3-34 for the description of these commands. Refer to "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43 for the reason code meanings.

Adapter open errors take the normal command completion appendage.

#### **DLC status appendage**

This appendage is defined in the DLC\_STATUS\_EXIT field of the CCB parameter table for a DLC.OPEN.SAP command.

The adapter support software passes parameters to the appendage on entry. Register CX contains the adapter number. Register AX contains the DLC status code. Register SI contains a user-defined value from the USER\_STAT\_VALUE field in the parameter table of the DLC.OPEN.SAP command. Registers ES and BX point to the DLC status table. Refer to "DLC Status Codes" on page B-25 for the code meanings and the DLC status table.

## Local Area Network Support Program (CCB1) Control Blocks

This is the description of the adapter support software CCB1 for DOS. The content of the first field indicates to the adapter support software which type of interface the application program wishes to use. If the first field contains either X'00' or X'01', the block is considered to be a CCB and either the *direct interface* or the *DLC interface* is being used. In the case where that field is less than X'03', it is considered to be the CCB adapter field. The values X'02' and X'03' cannot be used; they are reserved and the adapter support software will return an error code if they are used.

If the first field contains a byte greater than X'0F', the *NETBIOS interface* is being used and the control block is considered to be an NCB. The NCB is described under "NCB Field Explanations" on page 4-3. NETBIOS must have been loaded when a NETBIOS command is issued or the adapter support software will return an X'FB' return code.

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	Command field
2	CCB_RETCODE	1	DB	Completion code
3	CCB_WORK	1	DB	Adapter Support Software work area
4	CCB_POINTER	4	DD	Queue pointer and Adapter Support Software work area
8	CCB_CMD_CMPL	4	DD	Command completion user appendage
12	CCB_PARM_TAB	4	--	Parameters or pointer to CCB parameter table

**Note:** The above control block definition is to be used with both the direct interface and the DLC interface.

For a complete description of the fields, see "CCB Field Explanations" on page 2-17.

---

## Programming Conventions with Operating System/2 Extended Edition 1.1

### Dynamic Link Routine Interface Programming Conventions

#### Execution of Multiple Commands

To enhance performance of the adapter support software, the application program may request execution of multiple commands with a single invocation of the adapter support software. This is accomplished by allowing application programs to queue CCB requests using the CCB\_POINTER field of the CCBs. All queued CCBs will be linked with the CCB\_POINTER fields pointing to the next CCB in the queue of CCBs. All commands queued for a single invocation must be for the same adapter. If the chained commands are not for the same adapter, the processing of the queue will be terminated with all unprocessed commands returned to the user with the CCB\_RETCODE of the first CCB returned set to X'5F'.

While processing the queue of CCB requests, if an error is found the remaining CCBs in the queue will not be processed. The CCB containing the error plus a queue of the commands that have not been processed will be returned to the application program using a Bad Command Pointer. The address of this pointer is passed to the Dynamic Link Routine Interface. See "Dynamic Link Routine Interface (CCB2) Calling Conventions" on page 2-8.

Not all commands should be chained, especially commands that are dependent on the completion of other commands. For example, do not chain DLC.OPEN.SAP, DLC.OPEN.STATION, and DLC.CONNECT.STATION commands together because the later two commands are dependent upon the completion of the previous command. However, synchronous commands that execute in the PC system and commands that are independent of each other may be chained together without resulting in time related errors. Commands such as: TRANSMIT, READ, BUFFER.FREE, BUFFER.GET and commands for different STATION.IDs can be issued successfully in a chain of commands.

### **User's Data Segments Restrictions**

All application program data segments referred to in a call to the adapter support software must be accessible by the Operating System/2 process that is calling the adapter support software. That is, the CCB and all data areas pointed to by the CCB (for example, SAP buffers and transmit buffers) must be in the current local descriptor table (owned by the current Operating System/2 process) so that the adapter support software will have access to the areas in order to lock and update them.

All control blocks (for example, parameter tables) referenced with an offset must be located in the same segment as the associated CCB.

### **User's Data Segments Guidelines**

Data that will be processed by the adapter support software should be placed in separate segments. Segments supplied by application programs in requests should be separate from data segments that contain static local variables of the application programs. Allocating separate segments for data accessed by the adapter support software is not a requirement. However, using separate segments will limit the size of the data area that the adapter support software locks. The adapter support software will lock data to ensure that Operating System/2 does not move data to disk or other segments when the data is not frequently accessed. At interrupt time, the adapter support software must have immediate access to user data associated with CCB requests. By locking the segments containing data, the adapter support software ensures that Operating System/2 will not disturb the data. However, locking segments consumes RAM area, thus locking fewer segments will allow more physical RAM to be available. The following is a list of data structure types that should be assigned to a separate allocated segment.

- Receive Buffers
- Transmit Buffers
- Command Control Blocks and associated parameter tables.

To maximize overall performance, an application program should use a single segment to contain the above data structures. By providing a single segment all the associated data structures will be locked as long as one of the structures remain in the domain of adapter support software device driver. With the single segment locked, no additional locks will have to be done when requesting the adapter support software's services, unless the CCB of the request references a segment outside of the locked segment. Also, performance will be enhanced if all data structures start on an even byte boundary.



## Dynamic Link Routine Interface (CCB2) Calling Conventions

To request the Dynamic Link Routine Interface, two parameters must be placed onto the stack. These parameters are:

- CCB2 Pointer: Pointer to CCB to be processed
- Address of Bad Command Pointer: Address of a returned pointer value

This parameter is the address of a DD (Double Word) pointer (Bad Command Pointer). If an error is found with a command that is included in a chain of CCBs passed to Operating System/2 Extended Edition 1.1 on a single invocation, the address of the CCB containing the error with all unprocessed CCBs still chained to it will be returned. These commands will be pointed to by the Bad Command Pointer upon return from the Operating System/2 Extended Edition 1.1 invocation. The Bad Command Pointer will only be valid when the immediate return code in AX is set to X'0003'. If an immediate return code is set, the semaphores specified in the CCB are not cleared.

The application program must have access to the segments referenced by all pointers, for example, the CCB and all associated data structures.

When a list of commands is passed to Operating System/2 Extended Edition 1.1 on a single invocation, all chained commands must be for the same adapter.

For a given application program to make a request to Operating System/2 Extended Edition 1.1, it must:

- Push selector of a CCB2 onto the STACK.
- Push offset of a CCB2 onto the STACK.
- Push selector of address for "Bad Command Pointer" onto the STACK.
- Push offset of address for "Bad Command Pointer" onto the STACK.
- Invoke the Operating System/2 Extended Edition 1.1 Dynamic Link Routine Interface (ACSLAN module name within the ACSLAN dynamic link library), using the **far** CALL/RETURN interface. ACSLAN will remove the push parameters from the stack before returning to the caller.

Upon return the AX register will contain one of the following immediate return codes:

- X'0000'** Command accepted or command completed successfully.
- X'0001'** Invalid CCB pointer.
- X'0002'** CCB in error.
- X'0003'** CCB in error, check Bad Command Pointer.
- X'0004'** Unexpected operating system return code, adapter closed.
- X'0005'** Unexpected operating system return code.
- X'0006'** Invalid Command Pointer.

If on return the AX register is set to X'03' the Bad Command Pointer should be checked. If the Bad Command Pointer is non-zero, then it points to a queue of commands that were not processed, excluding the first CCB in this queue. The first of these commands was found to have an error and will have the CCB\_RETCODE field set. All other commands that are queued to the first CCB have not been processed, thus no return code is provided.

If on return the AX register is set to X'04' the adapter will be closed. If a READ command is outstanding, application program resource information is returned to the READ command's parameter table. See the READ command in Chapter 3, "The Command Control Blocks" on page 3-69 for more information.

If on return the AX register is set to X'05', the CCB\_WORK field should be checked for an Operating System/2 Extended Edition 1.1 function and the word at offset 24 of the CCB should be checked for the Operating System/2 Extended Edition 1.1 return code of the failing request.

## Dynamic Link Routine Interface (CCB2) Command Completion

User notification flags, semaphores and return codes are used to post events to the application programs. The choice of how each event is posted is left up to the application program. Some events are allowed to be posted differently.

User notification flags are used to post events as follows:

1. To request that information relating to an event be placed onto a completion list managed by the adapter support software. By placing an event onto the completion list, information relating to the event can be retrieved at a later time.
2. To enable notification of critical exceptions that result in the adapter closing. By enabling critical exception notification a user can be alerted of the event if a READ command is outstanding prior to the occurrence of the event.

All flags are 4 bytes in length and are preserved across invocations. The flags are "set" whenever non-zero values are used. If needed, these flags can be used to contain user specific information. However, if the flags are equal to X'00000000', then the flags are considered "not set." Nothing will be placed onto the completion list and no notification will be given for critical exceptions.

If an event is placed onto the completion list, the application program must issue a READ command to remove the event from the completion list. See the READ command description on page 3-69.

Semaphores may be provided with all commands. Upon completion of a command the semaphore will be cleared by the adapter support software to alert the application program of the command completion.

The completion return code for each command can also be polled by the application program to determine when the command has completed.

### User Notification Flags

The following user flags must be set if event information is to be placed onto the completion list.

- **CCB\_CMPL\_FLAG:** Command completion flag

For each command issued to the adapter support software, a CCB\_CMPL\_FLAG is included in the CCB. If this flag is set to a non-zero number, then upon completion of the command the address of the CCB will be queued onto a completion list.

- **RECEIVE\_FLAG:** Receive data flag

For each RECEIVE command issued to the adapter support software, a RECEIVE\_FLAG is included in the CCB's parameter table. If this flag is set to a non-zero number, then upon reception of data the first receive buffer address will be queued onto a completion list.

- **DLC\_STATUS\_FLAG:** DLC status change flag

For each DLC.OPEN.SAP command issued to the adapter support software, a DLC\_STATUS\_FLAG is included in the CCB's parameter table. If this flag is set to a non-zero number, detection of a DLC status change results in a copy of the current DLC Status table being queued onto a completion list. See "DLC Status Codes" on page B-25 for a list of DLC Status codes.

- **User Exception Flags (Non-critical exceptions)**

The user exception flags are set using the DIR.SET.EXCEPTION.FLAGS command. If these flags are set to a non-zero number, then upon detection of an exception condition the appropriate information will be queued onto a completion list. See "DIR.SET.EXCEPTION.FLAGS" on page 3-50 for a description of the DIR.SET.EXCEPTION.FLAGS command. The following is a list of the user exception flags that will enable information to be placed onto the completion list for non-critical exceptions.

- NETWORK\_STATUS\_FLAG: Network status flag
- SYSTEM\_ACTION\_FLAG: System action flag

Enabling notification of critical exceptions is done by setting the following user flags with the DIR.SET.EXCEPTION.FLAGS command.

- ADAPTER\_CHECK\_FLAG: Adapter check flag
- NETWORK\_STATUS\_FLAG: Network status flag
- PC\_ERROR\_FLAG: PC system detected error flag
- SYSTEM\_ACTION\_FLAG: System action flag

If these flags are set to a non-zero number and a READ command is outstanding requesting notification of critical exceptions, detection of a critical exception condition will result in the appropriate information being copied to the outstanding READ command's parameter table. At this time, the READ command will be posted as defined by the READ command's CCB.

**Note:** For more information on the exception information, see "Adapter Check for CCB3" on page B-42, "Network Status for CCB3" on page B-46, "PC System Detected Errors for CCB3" on page B-56, and "System Action Exceptions for CCB3" on page B-60.

## Posting of Events

All commands issued to the adapter support software can be posted using any combination of the three post mechanisms: setting the user flag along with issuing a READ command; waiting on a semaphore; or polling the return code set in the CCB. However, posting exceptions and DLC status changes must be implemented with setting the user flags and issuing a READ command.

If the associated user flag is not set, the event will not be queued to the completion list and the user must use one of the other mechanisms to post the event.

For each command, a semaphore can be passed to the adapter support software within the CCB. Upon completion of the command, the adapter support software will clear the semaphore. Thus, if an application program had a thread waiting on the semaphore, the thread would get dispatched. If neither the user flag nor the semaphore are used, the application program must poll the return code of the CCB to determine when the command has completed.

If both the user flag and a semaphore are used, the semaphore will be cleared when the command completes and the CCB of the command will also be placed onto the completion list. By clearing the semaphore, the application program has been notified of the command completion. However, the application program must still issue a READ command (if one is not already outstanding for the given event) to remove the CCB from the completion list.

There are two special cases where events are chained together in order to lessen the number of READ commands that must be issued to retrieve information from the completion list. Both RECEIVE and TRANSMIT commands can be issued specifying that event information relative to each command be linked together. If the user chooses to have receive data frames chained together and completed TRANSMIT commands' CCBs chained together, the following applies:

1. If a RECEIVE command is issued with the CMD\_CMPL\_FLAG and the RECEIVE\_FLAG set, then all receive data will be placed onto the completion list. If the RECEIVE command was issued requesting that received frames be chained and a READ command is issued with one or more frames being received that meet the READ command's requirements, the frames will be chained together using the first receive buffer of each frame.

2. If multiple TRANSMIT commands have completed that had the CMD\_CMPL\_FLAG set and requested chaining upon completion, the following will occur. Whenever a READ command is issued and more than one TRANSMIT CCB has completed that matches the READ command's requirements, the TRANSMIT CCBs will be chained together using the CCB\_POINTER of the TRANSMIT CCB and returned to the application program.

## Dynamic Link Routine Interface (CCB2) Control Blocks

This is the description of the adapter support software CCB2 for Operating System/2 Extended Edition 1.1 using the Dynamic Link Routine Interface.

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	Command field
2	CCB_RETCODE	1	DB	Completion code
3	CCB_WORK	1	DB	Adapter Support Software work area
4	CCB_POINTER	4	DD	Queue pointer and Adapter Support Software work area
8	CCB_CMPL_FLAG	4	DD	Command completion flag
12	CCB_PARM_OFFSET	2	DW	Offset to CCB2 parameter table
14	CCB_PARAMETER_1	2	DW	Parameter or reserved for an application program
16	CCB_SEMAPHORE	4	DD	Command post semaphore
20	CCB_APPL_ID	1	DB	Application ID
21	CCB_READ_FLAG	1	DB	READ chained to CCB flag
22	CCB_APPL_KEY	2	DW	Application key code
24	CCB_PARAMETER_2	2	DW	Parameter for System Key or reserved
<b>Note:</b> The above control block definition is to be used with both the direct interface and the DLC interface.				

For a complete description of the fields, see "CCB Field Explanations" on page 2-17.

## Device Driver Interface Programming Conventions

### User's Data Segments Restrictions

All application program data segments referred to in a call to the adapter support software must be accessible by the Operating System/2 process that is calling the adapter.

All segment references must be made with either GDT selectors or 32-bit physical addresses (CCB addresses, SAP buffer addresses, Transmit buffer addresses) and all segments must be locked using the Operating System/2 Device Help routine. See "RECEIVE.MODIFY" on page 3-84 and descriptions of the Transmit commands on page 3-88 for use of the 32-bit physical addresses. All other commands and data areas are mapped using the GDT selectors.

All control blocks (For example, parameter tables) referenced with an offset must be located in the same segment as the associated CCB.

## User's Data Segments Guide Lines

Data that will be processed by the adapter support software should be placed in separate segments from data segments that contain static local variables of the application programs. Allocating separate segments for data accessed by the adapter support software is not a requirement. However, using separate segments will limit the size of the data area that the adapter support software uses. The adapter support software assumes that data segments are locked prior to being called from an application program. It is therefore the application program's responsibility to lock the data areas. At interrupt time, The adapter support software must have immediate access to user data associated with CCB requests. By having the application program lock the segments containing data, The adapter support software is ensured that Operating System/2 will not disturb the data. The following is a list of data structure types that should be assigned to a separate allocated segment.

- Receive Buffers (use GDT selectors)
- Transmit Buffers (use 32-bit physical addresses for better performance)
- Command Command Blocks (CCBs) and associated parameter tables (use GDT selectors).

## Device Driver Interface (CCB3) Calling Conventions

To request the Device Driver Interface, the application program device driver must place the address of the CCB3 to be executed by the Device Driver Interface into registers ES and BX and push an invocation code of X'0000' onto the stack. The application program device driver then issues a Call Far instruction to the Operating System/2 Extended Edition 1.1 Device Driver Interface intercommunication entry point.

**Note:** The application program device driver must do an ATTACH OS/2 Device Help Function call to obtain the inter-device driver communication entry point of the Operating System/2 Extended Edition 1.1 Device Driver Interface (LANDD\$). Refer to the OS/2 command for details of call.

Upon return from the Device Driver Interface, all registers will contain their original values with the exception of the AX register. The AX register will contain the immediate return code.

For a given application program's device driver to make a request to the Device Driver Interface, it must:

- Set Register BX to the address offset of the CCB3 to be executed
- Set Register ES to the address selector of the CCB3 to be executed
- Push an invocation code of zero onto the stack
- Call Operating System/2 Extended Edition 1.1 Device Driver Interface (LANDD\$) inter-device driver communication entry point using the Call Far/Return Far interface.

Upon return, the AX register will contain one of the following immediate return codes:

- X'0000' Command accepted or command completed successfully.
- X'0001' Invalid CCB pointer.
- X'0002' CCB in error.
- X'0004' Unexpected operating system return code, adapter closed.
- X'0005' Unexpected operating system return code.
- X'0007' Invalid Invocation Code.

If on return the AX register is set to X'04', the adapter will be closed. If the user has an appendage for PC system detected errors, the function code of the Device Help request and the return code of the request that failed will be included in the information returned to the user in the PC system detected appendage's 20 byte information table.

## Device Driver Interface (CCB3) Command Completion

User appendages and return codes are used to post events to the application programs. The choice of how each event is posted is left up to the application program.

Events are posted to user appendages by one of the following methods:

1. Appendages request that information relating to an event be passed to the application program by an appendage call from the adapter support software when an event has occurred.
2. Appendages enable notification of critical exceptions that result in the adapter closing. By enabling critical exception notification, an application program can be alerted of the event by an appendage call from the adapter support software.

The application program must pass the adapter support software offsets to the appendages for these different events:

- Completion of commands
- Reception of data
- DLC Status change
- The following exceptions
  - Adapter Check
  - Network Status
  - PC Detected Error
  - System Action

The completion return code for each command can also be polled by the application program to determine when the command has completed.

### Posting of Events

If event information is to be posted to the application program, the following user appendages must be defined by passing an offset to the adapter support software through the different commands. The adapter support software will enter the application program's device driver with a Call Far instruction using the application program's device driver entry point obtained when the DIR.OPEN.ADAPTER command is issued. The application program's device driver must return using a Return Far instruction.

**Note:** For all appendage calls and the RECEIVE.MODIFY subroutine call, an invocation code of X'0001' is pushed onto the stack. It is the called device driver's responsibility to remove the invocation code from the stack.

- Command Completion

For each command issued to the adapter support software, a CCB\_APPNDG\_OFFSET is included in the CCB. The offset is a 2-byte DW field that the adapter support software uses for the address of the appendage that the adapter support software will pass to the application program in DI register when the application program device driver is called.

When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the application program.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the command completion appendage.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of the completing CCB.

- Registers DX and SI contain the address of a 12-byte information table relating to the command. These registers will be set to zero if no information is available to be returned to the application program. See Table 2-4 on page 2-20 for the information table.
- Register AX contains the CCB return code.

- Receive Data

When an application program issues a RECEIVE command to the adapter support software, a RCV\_DATA\_APPNDG is included in the CCB parameter table of the RECEIVE command. The offset is a 2-byte DW field that the adapter support software will pass to the application program's device driver when receive data is available and the application program's device driver is called.

When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the application program.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the RECEIVE appendage.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of the first SAP buffer.
- Registers AX and SI contain the address of the RECEIVE command's CCB for which receive data has been processed.

- DLC Status

When an application program issues a DLC.OPEN.SAP command to the adapter support software, a DLC\_STATUS\_OFFSET is included in the CCB parameter table of the DLC.STATUS command. The offset is a 2-byte DW field that the adapter support software will pass to the application program's device driver when DLC status data is available and the application program's device driver is called.

When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the application program.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the DLC status appendage as defined by the DLC.OPEN.SAP command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 20-byte information table. See "DLC Status Codes" on page B-25 for description of DLC Status codes.
- Register AX contains the DLC Status Code.
- SI Register contains the USER.STAT.VALUE defined with the DLC.OPEN.SAP command.

- Exception Conditions

The user appendages associated with exception conditions are set using the DIR.SET.EXCEPTION.FLAGS command. See Chapter 3, "The Command Control Blocks" on page 3-1 for an explanation of the DIR.SET.EXCEPTION.FLAGS command.

An appendage offset is included in the DIR.SET.EXCEPTION.FLAGS command for each of the conditions below. The offset is a 2-byte DW field that the adapter support software will pass to the

application program's device driver when an exception occurs and the application program's device driver is called.

When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the application program for the different exception conditions.

— Adapter Check

See "Adapter Check for CCB3" on page B-42 for more information.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the adapter check appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 20 byte information table.
- Register AX contains the reason code.

— Network Status

See "Network Status for CCB3" on page B-46 for more information.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the Network status appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 14-byte information table.
- Register AX contains the Network status.

— PC Detected Error

See "PC System Detected Errors for CCB3" on page B-56 for more information.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the PC system detected error appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 20-byte information table.
- Register AX contains the error code.

— System Action

See "System Action Exceptions for CCB3" on page B-60 for more information.



- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the offset of the system action appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 14-byte information table.
- Register AL contains the System Action ID.
- Register AH contains the SAP value associated with the System Action ID.

If the associated user appendage is not defined, the event will not be posted to the user.

## Device Driver Interface (CCB3) Control Blocks

This is the description of the adapter support software CCB3 for Operating System/2 Extended Edition 1.1 using the Device Driver Interface.

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	Command field
2	CCB_RETCODE	1	DB	Completion code
3	CCB_WORK	1	DB	Adapter Support Software work area
4	CCB_POINTER	4	DD	Queue pointer and Adapter Support Software work area
8	CCB_APPNDG_OFFSET	2	DW	Offset to CCB3 completion appendage
	- reserved -	2	DW	Reserved for application program
12	CCB_PARM_OFFSET	2	DW	Offset to CCB3 parameter table
14	CCB_PARAMETER_1	2	DW	Parameter or reserved for an application program
16	CCB_RESOURCE_ID	2	DW	Resource ID of application program process
	- reserved -	2	DW	Reserved for application program
20	CCB_APPL_ID	1	DB	Application ID
21		1	DB	Reserved for application program
22	CCB_APPL_KEY	2	DW	Application key code
24	CCB_PARAMETER_2	2	DW	Parameter for System Key or reserved
<b>Note:</b> The above control block definition is to be used with both the direct interface and the DLC interface.				

For a complete description of the fields, see "CCB Field Explanations" on page 2-17.

---

## Adapter Support Software Control Blocks for All CCBs

The application program must prepare a control block to request an activity from the adapter when the adapter support software is used. When the adapter support software analyzes the control block, it can determine which interface is needed by the content of the first bytes.

The content of the control blocks is explained in “Local Area Network Support Program (CCB1) Control Blocks” on page 2-6, “Dynamic Link Routine Interface (CCB2) Control Blocks” on page 2-11, and “Device Driver Interface (CCB3) Control Blocks” on page 2-16.

### CCB Field Explanations

The following CCB field descriptions apply to all three CCBs (CCB1, CCB2, and CCB3) unless otherwise stated.

---

#### CCB\_ADAPTER

**Explanation:** Defines which adapter is to be used. It must be either X'00' to use the primary adapter or X'01' for the alternate, adapter. The adapter number has the following binary format.

B'0000ceaa'

- Where 0000 (bits 4-7) are always zero.
- Where c (bit 3) is the common storage bit. If this bit is on, the CCB was issued from common memory.
- Where e (bit 2) is the extended CCB bit. If this bit is on, the CCB is extended two bytes, with the last two bytes indicating bank switching information.
- Where aa (bits 0 and 1) are the adapter number, B'00' - B'11' (only B'00' and B'01' are current valid adapter numbers).

If bank switching is not applicable, the extended bank switch information in the CCB is set to a null value: X'FFFF'.

The following list describes “common” and “extended” bits.

- If the common and extended bits are both zero (B'00'), the CCB has been issued by an application program in one of the memory banks.
- If the common and extended bits are zero and one (B'01'), the CCB is a “pseudo-CCB.” This means that a CCB issued by an application program in one of the memory banks has been substituted with an internally generated CCB, which contains the extended memory bank information.
- If the common and extended bits are one and zero (B'10'), the CCB has been issued by code in the common storage area. NETBIOS issues this bit value for all CCBs except transmits of user-defined data. Common storage can be accessed by the adapter support software independent of the currently active bank.
- If the common and extended bits are one and one (B'11'), the CCB has been issued by code in the common storage area. The CCB has extended memory bank information. NETBIOS issues this bit value for CCBs that transmit user-defined data.

If the value is greater than X'0F', the control block is an NCB.

**For CCB1:** Values of X'02' and X'03' are reserved, and if used, a CCB\_RETCODE of X'1D' is returned. Values greater than X'03' indicate that the NETBIOS interface is to be used. See Chapter 4, “NETBIOS” on page 4-1.

When used with a 3270 PC, the adapter number of a CCB has two additional bits defined and has the format described above.

**For CCB2 and CCB3:** Values from X'02' to X'FF' are reserved.

---

### **CCB\_COMMAND**

**Explanation:** Indicates the command to perform. A value of X'FF' is a permanently defined invalid command code. See Appendix A, "Valid Commands" for reserved and valid commands.

---

### **CCB\_RETCODE**

**Explanation:** The completion code as provided by the adapter support software. For all commands, this field is set to X'FF' by the adapter support software when the CCB is received. While the field is X'FF', the application program must not alter the CCB or any associated data. When the adapter completes the command, the adapter support software sets this field to the appropriate completion code. For all commands, X'00' means successful completion. See "CCB Return Codes Listed by Interface" on page B-2 for descriptions of all return codes.

---

### **CCB\_WORK**

**Explanation:** A work area field for the adapter support software to use.

---

### **CCB\_POINTER**

**Explanation:**

**For CCB1:** While the CCB\_RETCODE is X'FF', the adapter support software uses this field for command processing.

The application program uses this field as follows:

- When the adapter is closed, the application program interrogates this field to find the next command (CCB) in a queue of outstanding commands.
- When a DLC link station is sending "I" frames, multiple transmissions may be acknowledged at one time. All acknowledged "I" frames are queued and presented at one time to the application program. That is, the adapter support software issues an interrupt providing a return code in one CCB. In the CCB\_POINTER field of that CCB is the address of a queue of CCBs that contains an appropriate return code. This continues until a CCB\_POINTER field is zero ending the queue.

**For CCB2:**

- While CCB\_RETCODE is X'FF', this area may be used by the Operating System/2 Extended Edition 1.1 Dynamic Link Routine Interface for command processing.
- Application programs use this field under the following circumstances:
  1. When the application program wishes to request that multiple commands be processed as a result of a single invocation, the CCB\_POINTER is used to chain CCB2 requests.
  2. When the application program wishes to have a READ chained to the CCB2 to be used for its completion, the READ CCB2 address is placed in CCB\_POINTER and the CCB\_READ\_FLAG is set to a non-zero value.
  3. When the adapter is closed, a chained list of outstanding commands is presented to the user via the CCB\_POINTER of:
    - a. A CCB who's address is placed into the READ command's CCB parameter table.
    - b. A DIR.CLOSE.ADAPTER command that has completed successfully.

4. If transmissions are done specifying that completed transmission request's CCB2s be chained, they are linked together upon completion using the CCB\_POINTER. For this case the READ command must be used to retrieve the completed command's CCB2. The address of the first CCB is placed into the READ command's CCB2 parameter table.

**For CCB3:** Queue pointer and adapter support software work area

- While CCB\_RETCODE is X'FF' this area may be used by the adapter support software for command processing.
- When the adapter is closed, the application program interrogates this field to find the next command (CCB) in a queue of outstanding commands.

---

### CCB\_CMD\_CMPL

**Explanation:** This field is the address of a user appendage that the adapter support software will go to upon command completion. The appendage allows the user to obtain control after a command has been completed. See "Appendages" on page 2-4. When the user's appendage receives control at this point, the address of the CCB that was completed will be in registers ES and BX and the CCB\_RETCODE will also be in register AL. Register AH will be X'00'. See "Local Area Network Support Program (CCB1) Command Completion" on page 2-3.

---

### CCB\_CMPL\_FLAG

**Explanation:** This flag indicates whether or not a completed command should be posted using the READ command. The adapter support software will check this field after command completion. If the flag is not zero, the completion will be posted to the application program with a READ command. If a READ for command completions is already outstanding, it will be posted immediately. If there is no outstanding READ, the command completion will be queued internally to the adapter support software that is waiting for a READ for command completions. If the flag is zero, the completion will not be posted to the application program with a READ command. After completion the return code is set.

If the CCB\_CMPL\_FLAG is not set, the application program will either use the CCB\_SEMAPHORE or poll the CCB\_RETCODE field for notification of the command completion. It will be the application program's responsibility to poll the CCB\_RETCODE field for a value other than X'FF'. The value X'FF' is used to signify that the command is in progress and that the CCB and its associated data should not be altered.

**Notes:**

1. See the READ command for details on posting command completions.
2. As soon as any immediate command processing has been done by the adapter support software, the command is queued and control is returned to the application program that is using the adapter support software (CCB\_RETCODE is set to X'FF'). At that point, the application program may continue with other processing (not disturbing the CCB or any associated data). When the command completes, the return code will be set and the application program will be posted if a READ is outstanding.

---

### CCB\_APPNDG\_OFFSET

**Explanation:** The offset of a user appendage within the application program device driver's code segment that is going to handle the command's completion. When the command completes, the adapter support software calls the application program device driver at its intercommunication entry point, pushing the invocation code of X'0001' onto the stack. The appendage offset is passed in the DI register. The application program device driver has the responsibility of calling the appendage located at the address offset specified in the DI register.

Application Calls:

When the adapter support software calls the application program device driver at the intercommunication entry point, the following information is provided.

- An invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software, the application program must remove the invocation code from the stack.
- Register DI contains the appendage offset as defined by each individual command.
- Register DS contains the application program's device driver protect mode data segment selector.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of the completing CCB.
- Registers DX and SI contain the address of a 12-byte information table pertaining to the command which has completed. If no information is available for the command, these registers will contain zero.

OFF-SET	FIELD NAME	BYTE LEN	DESCRIPTION
0	CCB_COUNT	2	DW Count of CCBs chained to EVENT_CCB_POINTER
2	EVENT_CCB_POINTER	4	DD Pointer to CCB terminated as a result of the CCB addressed with the ES and BX registers
6	BUFFER_COUNT	2	DW Count of buffers chained to FIRST_BUFFER_ADDR
8	FIRST_BUFFER_ADDR	4	DD Address of first receive buffer for buffer pool

- Register AX contains the CCB3 Return code.

---

### CCB\_PARM\_TAB

**Explanation:** This field points to additional parameters that are command-specific. These parameters are explained with the related command descriptions.

If the parameters required are 4 or fewer bytes in length, they are provided in the CCB\_PARM\_TAB field instead of in an area pointed to by the field.

---

### CCB\_PARM\_OFFSET

**Explanation:** This field points to additional parameters that are command-specific. These parameters are explained with the related command descriptions.

If the parameters require 2 or fewer bytes, they are provided in the CCB\_PARM\_OFFSET field. If the parameters require more than two bytes, the field contains the offset within the selector of the parameter table for this command.

---

### CCB\_PARAMETER\_1

**Explanation:** This field can contain another 2 bytes of parameter data to combine with the CCB\_PARM\_OFFSET field for a total of 4 bytes, or it can contain user specific data.

---

## CCB\_SEMAPHORE

**Explanation:** A system semaphore can be used to notify an application program of a command completion. When the command completes, the adapter support software will clear the CCB\_SEMAPHORE to alert the application program that the command has completed. To specify a system semaphore, the CCB\_SEMAPHORE field must contain a handle of a system semaphore that is returned from Operating System/2 Extended Edition 1.1 when the system semaphore is created or opened. System semaphore handles provided on the DIR.OPEN.ADAPTER command should be used to obtain optimum performance.

Only the Operating System/2 Extended Edition 1.1 process that issues the DIR.OPEN.ADAPTER command and provides system semaphore handles can use these handles. All other processes associated with the application program must provide a handle returned from Operating System/2 Extended Edition 1.1 for the given process when the system semaphore is created or opened.

If the application program does not want to use a semaphore, the CCB\_SEMAPHORE field should be coded as zero. If the CCB\_SEMAPHORE field contains an invalid SYSTEM semaphore handle (excluding zeros), the process will be terminated by Operating System/2 Extended Edition 1.1 whenever the handle is used in an Operating System/2 Extended Edition 1.1 call.

### Notes:

1. The system semaphore must be created with option set to exclusive ownership not required.
2. The adapter support software does not set the CCB\_SEMAPHORE; it only clears it.

---

## CCB\_RESOURCE\_ID

**Explanation:** A Resource ID must be used to allow proper clean-up of resources owned by terminating processes of an application program. All control blocks that have been passed to the adapter support software can have a resource ID associated with them. This allows the adapter support software to know which resources are associated with which application programs. This ID is required when an application program consists of more than one Operating System/2 Extended Edition 1.1 process. When a process terminates, the application program's device driver must clean up the resources associated with the process by issuing the PURGE.RESOURCES command with a PURGE\_RESOURCE\_ID. All control blocks that have a resource ID matching the PURGE\_RESOURCE\_ID will be freed by the adapter support software.

Memory passed to LANDD\$ in the form of CCBs or buffers can be owned by different processes of an application program or by an application program's device driver. Since the LAN device drivers cannot guarantee that the active process is the owner of memory being passed to it, all control blocks and buffers will be associated with the Resource ID. The Resource ID will be passed as a parameter with the CCB and will be associated with the CCB and all other control blocks (logs, buffers) referenced by the CCB.

If an application program consists of more than one Operating System/2 Extended Edition 1.1 process, it is the application program's responsibility to manage the clean-up of each process' resources. For example, if an application program has two Operating System/2 Extended Edition 1.1 processes that have both been allocated memory that is being used in the application program's SAP buffer pool, then when one of the processes terminates the application program should notify LANDD\$ with the PURGE\_RESOURCE command. The Resource ID will be passed as a parameter to specify what control blocks should be removed (cleaned up) from the LAN device driver's internal queues.

---

## CCB\_APPL\_ID

**Explanation:** The ID of the application program issuing the command. The CCB\_APPL\_ID is returned on the DIR.OPEN.ADAPTER request and must be used by the application program for all following commands that the application program issues. Some of the CCBs can be issued with System Key, such as DIR.SET.GROUP.ADDRESS, DIR.READ.LOG and others, then the CCB\_APPL\_ID field will not be needed. Otherwise, all of the CCBs (with the exception of the DIR.STATUS command) are required to use CCB\_APPL\_ID, which is returned on DIR.OPEN.ADAPTER CCB.

**Note:** System Key is for system administrator usage only.

---

### **CCB\_READ\_FLAG**

**Explanation:** An application program can specify that a READ command is chained to this CCB using the CCB\_POINTER field. This READ command will be used to process the completion of this CCB only, and not other commands that may have previously completed.

---

### **CCB\_APPL\_KEY**

**Explanation:** A key code used to provide resource security for application programs. The CCB\_APPL\_KEY parameter must match the CCB\_APPL\_KEY code provided by the user on the DIR.OPEN.ADAPTER request for the given command to succeed. If the user chooses not to use a key code (key code is set to zero) when he issues the DIR.OPEN.ADAPTER command, then the CCB\_APPL\_KEY parameter is not checked by the adapter support software when a request is made. Some commands can be issued with the System Key, such as DIR.SET.GROUP.ADDRESS, DIR.READ.LOG, and others. When this is done the CCB\_APPL\_KEY field will not be checked. If the command is issued without the System Key, the CCB is required to have the same CCB\_APPL\_KEY as the application program did in the DIR.OPEN.ADAPTER command.

**Note:** The System Key is for system administrator usage only.

---

### **CCB\_PARAMETER\_2**

**Explanation:** Command parameters (2 bytes of parameter data). Usually used for the System Key parameter.

**For System Key:** This parameter is used to enable only a system administrator to perform operations that could stop ring communication for application programs.

This key code is used for the following:

- Change functional address
- Change group addresses
- Reset selected SAPs and stations or all SAPs and all stations
- Relinquish ownership of direct stations
- Force a physical close for an adapter
- Force the adapter to initialize
- Read and reset adapter error and direct interface logs.

The System Key is not typically used by application programs, but for maintenance and problem determination.

This command may be issued by a system administrator with the SYSTEM KEY as defined by configuration parameters. If the adapter has not been opened by the system administrator, only polling the of the CCB\_RETCODE field can be used for posting of this command completion. CCB2 can post the command completion also using an Operating System/2 system semaphore. If the adapter has been opened and an application program ID has been returned, this command can be posted like any other command. See "System Action Exceptions for OS/2" on page B-58 for more information on the System Key.

If the System Key is not used, it should be coded as X'0000'.

---

## Addressing

Each adapter using the network has an address called the `NODE_ADDRESS`. When frames are sent on the network by adapters, the frame contains two of these addresses: a source address and a destination address. The frame is sent to the destination address adapter by the source address adapter.

Additional address and link information to be used in other transmission layers may be included in the frame following the LAN addresses. Additional addressing is used in the implementation of both the LLC and NETBIOS. The LLC sub-layer uses an address known as a service access point (SAP), described in the next section. NETBIOS addressing is described further in Chapter 4, "NETBIOS."

An adapter is provided with a permanent universally administered address. Additionally, the application program has the capability to provide a temporary replacement for this address and to provide a group address for the adapter.

**Note:** The `NODE_ADDRESS` can only be changed by the configuration parameters when using CCB2 or CCB3.

Group and functional addresses can be set and used as destination addresses; they may not be used as source addresses.

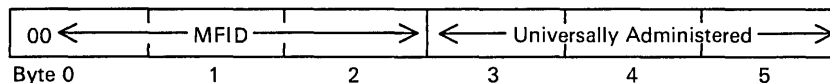
See the *IBM Token-Ring Network Architecture Reference* for uses and restrictions for these types of addresses.

---

## Adapter Addresses

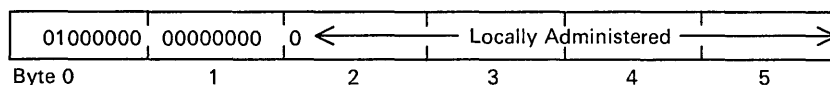
Adapters are able to identify the intended recipient of any frame because each adapter has a unique address. There are two types of addresses: universally administered and locally administered.

All PC system network adapters manufactured by IBM have universally administered addresses encoded on them. These addresses use the following format:



The first two bits (B'00') indicate that the address is a universally administered address. The "MFID" field contains the manufacturer's identification. The IEEE assures that every universally administered address is unique.

The application program may assign locally administered addresses. A locally administered address overrides the universally administered address encoded on the adapter. These addresses use the following format:



The bit 0 and 1 in byte 0 identify the address as locally administered. Because of restrictions placed on addresses by certain networking protocols, we recommend that you assign addresses in the range 00000001 - 79999999 (decimal). Your network administrator is responsible for preserving the uniqueness of these addresses.



For additional information about maintaining addresses, see Chapter 2 of the *IBM Token-Ring Network Administrator's Guide*.

## SAPs, Stations, and IDs

There are several types of stations that can be defined to the adapter. They are referred to by the STATION\_ID field in command descriptions. The direct station is automatically assigned by the adapter when the adapter is opened. The direct station is referred to by three STATION\_IDs. This station is automatically prepared to receive frames from the network when the DIR.OPEN.ADAPTER command (for DOS) or the DIR.OPEN.DIRECT command (for OS/2) is issued, but the application program must issue a RECEIVE command to make the information available at the Direct interface. The three STATION\_IDs are described below and differ only in how they handle receiving frames.

The direct station's STATION\_IDs are:

- X'0000'** This STATION\_ID of the direct station receives all frames (MAC & NON-MAC) not directed to other defined stations. This station may transmit MAC and non-MAC (data) frames.
- X'0001'** This STATION\_ID of the direct station receives MAC frames and transmits either MAC or non-MAC frames. The PC Network does not use MAC frames.
- X'0002'** This STATION\_ID of the direct station receives non-MAC frames and transmits either MAC or non-MAC frames.

Additionally, other stations can be opened by the application program. Service access points (SAPs) can be opened for communications with service access points in other devices connected to the network.

Both uses of the DLC interface, connectionless and connection operations, use service access points (SAPs) for communication on the network. An application program can open several SAPs for an adapter and each SAP can have several link stations opened that are associated with it. These link stations can then be directed to connect to link stations in other adapters (or even the same adapter). A SAP can be established to operate in one of two ways:

1. XID command frames will be handled by the LLC sublayer.
2. XID command frames will be passed to the application program for handling.

An option is set when the SAP is opened defining the handling of received XID commands. XID responses are always passed to the application program. See "Transmitting, Receiving, and Buffers" on page 2-34.

When an application program opens a SAP, the application program assigns a SAP\_VALUE and the adapter or adapter support software assigns a STATION\_ID. Communication between the application program and the adapter support software refers to a SAP by the 2-byte STATION\_ID. For SAPs, the first byte of the STATION\_ID identifies the SAP and the second byte is zero. When a link station is associated with a SAP, a new STATION\_ID is assigned. That STATION\_ID is two bytes, the first byte identifies the associated SAP and the link station number is the second byte. All link station numbers are unique for a given adapter, even if they are associated with different SAPs. Both SAPs and link stations are referenced by using the STATION\_ID field. For example, X'0100' represents a SAP and X'0108' represents a link assigned to that SAP.

When a SAP is used to communicate with another SAP, the application program provides the STATION\_ID to identify the local SAP and provides a destination address and SAP value (SAP\_VALUE) to identify the remote SAP. The same information is needed to open a link station. When both devices have a SAP and link station opened, a connect command actually initiates establishing the link connection.

The LLC header part of a frame contains two one-byte SAP values: the destination SAP (DSAP), and the source or sending SAP (SSAP). The SAP value actually uses only seven of the eight bits. One bit of the

SSAP is used to indicate whether the frame is an LLC command or response, and one bit of the DSAP is used to identify the target SAP as a group or individual SAP. The bit used is the low order bit of the SAP value supplied by the user in the various SAP commands. An individual SAP value is always even, and a group SAP value always odd.

A group SAP is a set of open individual SAPs. The global SAP is a special case of a group SAP for which the set consists of all currently open individual SAPs. When a frame is sent to a group SAP, a copy of the frame is passed to each individual SAP that is a member of the group. Note that frames cannot be sent from a group SAP as the bit that indicates group or individual has a different meaning in SSAP.

A SAP may be opened as an individual SAP, or a group SAP, or both. This is done using the option bits and the SAP value provided in the parameter list of the DLC.OPEN.SAP command. If the individual option is chosen, frames containing a DSAP equal to the SAP value with the low order bit off will be accepted. If the group option is chosen, frames containing a DSAP equal to the SAP value with the low order bit set on will be accepted. If both options are selected, both odd and even DSAP values will be recognized.

An additional option bit is used to specify group membership. A SAP opened with the individual option may be designated to be a member of one or more group SAPs, provided that the group member option is also selected. The group SAPs to which it will belong may be specified in the DLC.OPEN.SAP and DLC.MODIFY commands. Membership is deleted using the DLC.MODIFY command before closing the SAP. The only restriction is that all members of a particular group must have selected the same XID handling option.

Refer to Figures 1-5 and 1-6 starting on page 1-11. For a transmitted frame, the destination address in the LAN header is the remote NODE\_ADDRESS. The source address in the LAN header is the local NODE\_ADDRESS. The DSAP is the destination SAP\_VALUE (RSAP\_VALUE), and the SSAP is the local SAP\_VALUE. At the receiving end the interpreting of local and remote fields will be exchanged. For example, the destination address field is the local NODE\_ADDRESS of the receiving adapter.

The maximum number of user assigned group and individual SAPs possible is 254. The maximum number of link stations per adapter is 255 (all of which may be assigned to the same SAP). However, RAM and memory constraints will limit the number of SAPs and link stations that may be open at one time. See "Calculating RAM and Work Area Usage" on page 2-43 to calculate the number of SAPs and link stations.

More information about these SAPs and links is included with related command descriptions.

## **SAP Assignments**

The following SAPs are opened automatically:

- Null SAP X'00'
- Global SAP X'FF'

The Null SAP is opened automatically (with a SAP value of X'00'). It represents the LLC as a whole. The Null SAP provides the ability to respond to remote nodes even when no SAP has been activated. This SAP supports only connectionless service and responds only to XID and Test Command frames. The Null SAP is not accessible to the local application program.

The Global SAP is opened automatically (with a SAP value of X'FF'). It is a group SAP with all open individual SAPs as members. XID, TEST, and UI frames directed to the Global SAP will be passed to each open SAP in turn, with the DSAP field in the received frame buffer set equal to the receiving individual SAP\_VALUE, where they will be handled according to frame type.

SAP X'E4' is opened automatically for the PC Network.

If NETBIOS is to be used, it will use SAP X'F0'. SAPs X'F1' to X'FF' are reserved.

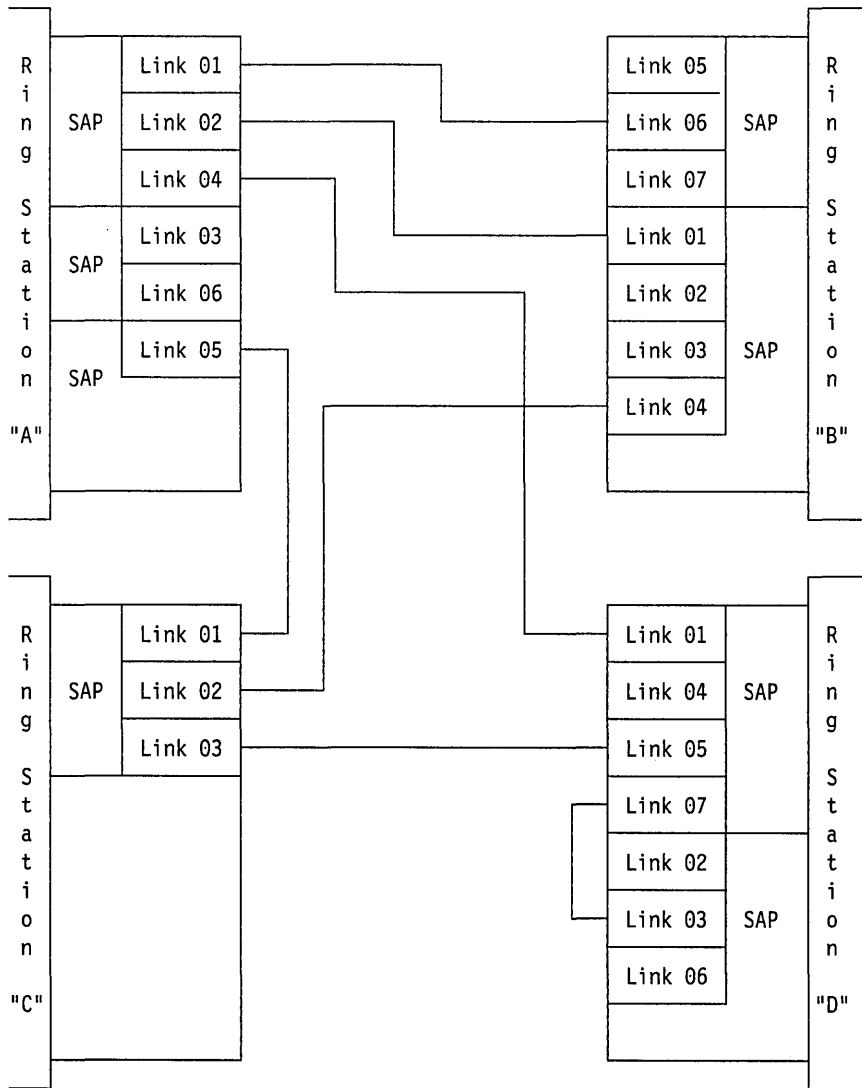


Figure 2-1. SAPs and Link Stations

---

## DLC

The DLC interface provides an interface to application programs using the Logical Link Control sublayer of Data Link Control protocol. The interface may be used in two ways.

- For IEEE type 1 communication: connectionless communication between devices providing no guarantee of delivery.
- For IEEE type 2 communication: connection-oriented services.

Much of the communication overhead function is provided by the adapter and the adapter support software, which permit simple programming. See the *IBM Token-Ring Network Architecture Reference* for more about communication

### Types of Service

The IBM Token-Ring Network and IBM PC Network support IEEE 802.2 type 1 and type 2 service as described in *IBM Token-Ring Network Architecture Reference*. Type 1 is connectionless service allowing transmission and receipt of Unnumbered Information (UI) frames, Exchange ID (XID) frames, and TEST frames. Type 1 uses unnumbered LLC Protocol Data Units (LPDUs). Frames sent using this type of service are not followed by a transmission from the receiving device verifying correct receipt and sequence of events unless provided by an application program in that device. Recovery and retry actions must be controlled by the application program. Type 2 is connection-oriented service providing guaranteed delivery and using numbered LLC Protocol Data Units (LPDUs).

### Command Sequences

When Logical Link Control protocols are used, commands must be issued in certain sequences to obtain the desired result.

In all cases, the adapter must be initialized and opened prior to the use of any transmit and receive commands.

A possible sequence could be:

Table 2-5. CCB1 and CCB3 Command Sequence

CCB1 and CCB3 Command	Comments
DIR.INITIALIZE	Select, clear, and test adapter
DIR.OPEN.ADAPTER	Make ready, set parameters, and connect the adapter to the ring <b>For CCB3:</b> Performs a logical open and returns parameters
DIR.SET.EXCEPTION.FLAGS	<b>For CCB3:</b> Enable exception notification
DIR.OPEN.DIRECT	<b>For CCB3:</b> Open the direct station for one application program
RECEIVE	Prepare for received data for direct station
DLC.OPEN.SAP	Allocate a Service Access Point
RECEIVE	Prepare for received data for this SAP
DLC.OPEN.STATION	Prepare a link station
RECEIVE	Prepare for received data for this link station
DLC.CONNECT.STATION	Initiate the communication link with remote station

**For CCB1 only:** The DIR.INITIALIZE command should be issued only if the adapter is known to be dedicated to the application program. The DIR.INTERRUPT command can be issued to determine if a DIR.INITIALIZE command is needed.

Table 2-6. CCB2 Command Sequence

CCB2 Command	Comments
DIR.OPEN.ADAPTER	Performs a logical open and returns parameters
DIR.STATUS	Obtain the current status of the Network
DIR.SET.EXCEPTION.FLAGS	Enable exception notifications
READ	Allow for posting of exception events
DIR.OPEN.DIRECT	Open the direct station for one application program
RECEIVE	Receive for direct stations
READ	Allow posting for direct stations receive data
DLC.OPEN.SAP	Allocate a Service Access Point
READ	Allow posting for DLC status change
READ	Allow posting for SAP station and its link station receive data
DLC.OPEN.STATION	Prepare a link station
DLC.CONNECT.STATION	Initiate the communication link with remote station

After this sequence has been completed, the application program can transmit and receive data on a link station in the following manner:

- **Receiving data**

Check the receive return code if no appendage was used, or check to see if the appendage routine has received data. After moving data from the receive buffer, issue a BUFFER.FREE command to return the buffer to the pool.

- **Transmitting data**

Any buffer can be used or the application program can issue a BUFFER.GET command to obtain enough buffers to contain the transmit data, move the data to the buffers, and issue a TRANSMIT.I.FRAME command. Issue a BUFFER.FREE command when the transmit is completed to return buffers that were originally retrieved from the buffer pool, with the exception of buffers referenced by XMIT\_QUEUE\_TWO transmit commands fields.

When preparing to leave the application program or network communication is no longer required, the following commands should be issued:

Command	Comments
DLC.CLOSE.STATION	Close the link station
DLC.CLOSE.SAP	Close the Service Access Point
DIR.CLOSE.DIRECT	<b>For CCB2 and CCB3:</b> Close the direct station
DIR.CLOSE.ADAPTER	Remove from the ring

## Link Station States

LLC type 2 protocol maintains primary and secondary states for each link station. Only one of the primary states may be active at a time. If the application program issues a command to a link station that is not valid for the current state, the command will be rejected with a return code of X'41'. DLC.MODIFY and DLC.FLOW.CONTROL commands are accepted in all states. If a link station is not established, there is no control block and no primary and secondary states exist. Then the link station is "non-existent."

Changes in the DLC status of the link station are reported to the interface. See "DLC Status Codes" on page B-25 and "Suggested Actions in Response to DLC Status" on page B-26 along with the following state information.

The primary states are:

- **Link Closed**

All received frames will be ignored in this state. The state is entered when:

- A DM response to a SABME or DISC has been queued for transmission. The close command that caused the transmission will be completed when the transmission is completed.
- A DM or UA response to a DISC has been received. The close command that caused the transmission will be completed when the transmission is completed.
- A reset command has been received, but a transmission has already been queued or is in process and must be completed before the link station can be released.

- **Disconnected**

All received frames will be ignored in this state except DLC frames with the poll bit set, and for which a DM is transmitted, and a SABME which is reported to the PC system. A DLC.CLOSE.STATION or DLC.CONNECT.STATION command will be accepted when this state is active. The state is entered when:

- A DLC.OPEN.STATION command has been accepted.
- A SABME for a previously non-existent station has been accepted.
- A DM response or DISC command from the paired station has been received.
- The retry count has been exhausted due to timeouts.

- **Disconnecting**

This state is normally entered when:

- The initial in-process return code is supplied after receipt of a DLC.CLOSE.STATION command. This state will be maintained until one of the following occurs:
  - Either a UA or DM response to the transmitted DISC command is received
  - A SABME command is received and a DM response has been successfully transmitted
  - The retry count expires.

Exit from this state is normally to link-non-existent or link-closed state. Since the DLC.CLOSE.STATION command remains in process while the link is in disconnecting state, no other commands will be accepted. All received frames other than SABME, DISC, UA, and DM will be ignored while in this state.

This disconnecting state may also be entered upon expiration of the retry count in FRMR received. In this case, exit is to the disconnected state.

- **Link Opening**

Unexpected received frames are ignored in this state. The state is entered by:

- A DLC.CONNECT.STATION command being issued by the PC system. Before entering this state, the adapter will transmit either a SABME command, or a UA response if a SABME has been received from the remote station.

If a SABME was transmitted, the adapter expects a UA response. On receipt of the UA response, it will transmit an RR command-poll and change to the link-opened (checkpointing) state.

If a UA was transmitted, the adapter expects either a supervisory command or an information frame, on receipt of which it will change state to link opened state (possibly with remote busy).

If the expected frame is not received and the retry count is exhausted, the link will be returned to the disconnected state unless a SABME has been received.

The DLC.CONNECT.STATION command will be returned with a successful return code or with an indication that the remote station failed to respond.

- **Resetting**

All received frames except DISC, DM, FRMR, and SABME will be ignored. Only DLC.CLOSE.STATION and DLC.CONNECT.STATION commands will be accepted by the adapter when in this state. The state is entered when a SABME command frame is received from the remote station when the link is open and not in disconnected state or link closed state.

- **Frame Reject Sent**

All received frames except DISC, DM, FRMR, and SABME will be ignored. Only DLC.CLOSE.STATION and DLC.CONNECT.STATION commands will be accepted by the adapter when in this state. The state is entered when an illegal frame is received and a FRMR frame has been transmitted.

- **Frame Reject Received**

All received frames except DISC, DM, and SABME will be ignored. Only DLC.CLOSE.STATION and DLC.CONNECT.STATION commands will be accepted by the adapter when in this state. The state is entered when a FRMR has been received.

- **Link Opened**

This is the only state in which information transfer is allowed and in which TRANSMIT commands will be accepted. In this state the adapter will handle sequential delivery and acknowledgment of information frames, together with retransmission if required. The state is entered when The adapter passes from the link opening state after the SABME-UA exchange, which completes the connection protocol.

The link station secondary states are:

- **Checkpointing**

A poll is outstanding, I frame transmission is suspended.

- **Local Busy (user)**

A DLC.FLOW.CONTROL command with a set-local-busy option has been accepted. I frame reception is suspended until a DLC.FLOW.CONTROL command with a reset-local-busy (user) option has been accepted.

- **Local Busy (buffer)**

An out-of-buffer return code has been set by the PC system in response to a request for data service on a receive. I frame reception is suspended until a DLC.FLOW.CONTROL command with a reset-local-busy (buffer) option has been accepted.

- **Remote Busy**

An RNR frame has been received from the remote station. I frame transmission is suspended until a receive ready or reject response, or a SABME command, or an in-sequence I response frame with the F bit set to B'1' has been received.

- **Rejection**

An out-of-sequence I frame has been received from the remote station and an REJ transmitted. I-frame reception is suspended until an in-sequence I frame or a SABME has been received.

- **Clearing**

A poll is outstanding and a confirmation of clearing local busy is required after the response is received.

- **Dynamic Window**

The remote station is on a different ring, and there appears to be congestion through the bridge(s).

## Timers

The DLC functions use three timers:

**T1** Response timer

**T2** Acknowledgment timer

**Ti** Inactivity timer.

See the *IBM Token-Ring Network Architecture Reference* for details about the timers.

The rate at which each of these timers is stepped and the value at which they time out are selectable by parameters. The rate of stepping is referred to as the “tick” and is defined by the following.

**For CCB1:** Define the “tick” with fields in the DLC open parameters provided to the adapter with the DIR.OPEN.ADAPTER command.

**For CCB2 and CCB3:** Define the “tick” with the configuration parameters at system initialization time.

Each timer requires a short timer tick (TICK\_ONE) and a long timer tick (TICK\_TWO). The period between timer ticks will be some number of 40 millisecond intervals.

The timer value, or count at which it expires and interrupts the adapter, is selected with parameters provided to the adapter when a DLC.OPEN.SAP, DLC.OPEN.STATION, or DLC.MODIFY command is issued.

A timer value is selected by using a number between 1 and 10. Each timer is divided into two groups of possible values.

1. If the number selected is between 1 and 5, the short timer tick (TICK\_ONE) will be used and is referred to as group 1. The timer value will be equal to the number selected multiplied by the short timer tick value (number selected x short tick value).
2. If the number selected is between 6 and 10, the long timer tick (TICK\_TWO) will be used and is referred to as group 2. The timer value will be equal to number selected minus 5 multiplied by the long timer tick value ((number selected - 5) x long tick value).

Therefore, there are three timers with two rates selectable for each, or six parameters to be selected.

Each DLC.OPEN.SAP command will set the values for the three timers for that specific SAP using the rates selected for the entire adapter. For example, if the value of the T1 timer in one SAP is 4 and the value for the T1 timer in another SAP is 7, the short rate of stepping is selected for the Response timer on the one SAP and the long rate of stepping is selected for the Response timer in the other SAP. The group-2 timer values should be used when longer delays are expected, such as when in a multi-ring environment.



The time of expiration is not exact, but falls into a range starting with the calculated time.

For example, if a given timer chose the following tick values:

Group-one tick: 200 milliseconds

Group-two tick: 1 second

then the following timer values would be available:

Group one (milliseconds)			Group two (seconds)		
number	value	actual range	number	value	actual range
1	200	200- 400	6	1	1-2
2	400	400- 600	7	2	2-3
3	600	600- 800	8	3	3-4
4	800	800-1000	9	4	4-5
5	1000	1000-1200	10	5	5-6

The next section includes guidelines for selecting timer values.

## Guidelines for Selecting Parameter Values

Following are some basic guidelines to consider when selecting parameter values for the network adapter. There are several basic parameters that may affect the performance obtained when using the Data Link Control (DLC) functions of the adapter. In most cases the default values will provide efficient operation. See "Timers" on page 2-31 and the parameter fields of the DLC.OPEN.SAP, DLC.MODIFY and DLC.OPEN.STATION commands. The parameters that are outlined here are:

Parameter	Pseudo Parameter
Response Timer (T1)	The TIMER_T1 parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Inactivity Timer (Ti)	The TIMER_Ti parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Receiver Acknowledgment Timer (T2)	The TIMER_T2 parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Maximum Length I-Field (N1)	The MAX_I_FIELD parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Maximum Number of Retransmissions (N2)	The MAX_RETRY_CNT parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Number of I-Format LPDUs Received before Sending Acknowledgment (N3)	The MAX_IN parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Number of Acknowledgments Needed to Increment Ww (Nw)	The MAXOUT_INCR parameter of a DLC.OPEN.SAP, DLC.MODIFY or, DLC.OPEN.STATION command.
Maximum Number of Outstanding I-Format LPDUs (TW)	The MAXOUT parameter of a DLC.OPEN.SAP, DLC.MODIFY, or DLC.OPEN.STATION command.

### Response Timer (T1)

The Response Timer (T1) is maintained by the sending adapter whenever an I-format LPDU or a command LPDU with the poll bit set to B'1' is sent. Should this timer expire before a response is received, the sending adapter solicits remote link station status by sending a supervisory command LPDU with the poll bit set to B'1'. The T1 timer should therefore be greater than the

total delays that the frame might encounter within the sending node, the network, and the receiving node. Normal settings for the T1 parameter should be in the range of 1 to 2 seconds. For instance, a setting above 2 seconds may result in noticeable delays to those responses that must be retransmitted (typically less than 3 percent of the total frames).

**Inactivity Timer (Ti)**

The Inactivity Timer (Ti) runs whenever the Response Timer (T1) is not running. Expiration of this timer suggests that the link may have been lost. The Inactivity Timer (Ti) should be five to ten times greater than T1 and it is recommended that the minimum be 30 seconds.

**Receiver Acknowledgment Timer (T2)**

A link station starts T2 when an I-format LPDU is received into PC system memory. T2 is stopped when an acknowledgment is sent either with an outgoing frame or when the number of I-format LPDUs received before sending acknowledgment (N3) value is reached. If T2 expires, the link station must send an acknowledgment as soon as possible. The value of T2 must be less than that of T1 to ensure that the remote link station will receive the delayed acknowledgment before T1 expires. Typical values for T2 will normally be 80 to 256 milliseconds.

**Maximum Length of I-field (N1)**

The Maximum Length of I-field (N1) parameter is used primarily to enable a pair of stations to establish the maximum size frame that can be received by either station. For example, one station may be able to transmit and receive frames up to 2KB each while the other can only send and receive frames of 1KB or smaller. Under no circumstance should the N1 value exceed the total amount of receive memory available. Also, since the N1 value implies that a station may transmit frames of length N1 bytes, N1 should not exceed the Data Hold Buffer (DHB) size (For CCB1: the DHB\_BUF\_LEN parameter of the DIR.OPEN.ADAPTER command) for the adapter in that station.

A key factor in selecting the N1 value is the receive buffer capacity of the destination adapter. Server devices, for example, may support several sessions concurrently, and therefore have a more limited buffer capacity than a workstation. (See "Calculating RAM and Work Area Usage" on page 2-43.)

Table 2-7. Maximum I-field Length for Network Adapters		
Token-Ring Network		PC Network Adapter
Data Rate of 4 Mbps	Data Rate of 16 Mbps	
4905 Bytes	18000 Bytes	4905 Bytes

N1 should never exceed 2042 bytes with the Token-Ring Network PC Adapter or with PC Network adapters. N1 values smaller than 512 bytes may result in a perceived decrease in station-to-station response times.

**Maximum Number of Retransmissions (N2)**

The Maximum Number of Retransmissions (N2), or MAX\_RETRY\_CNT, defines the maximum number of attempts in which a sending adapter will perform the checkpoint procedure following the expiration of the T1 timer. The combination of T1 and N2 should be great enough to allow for error detection and recovery on the network. This count also prevents continual retransmission of the same I frame.

Typical values for N2 will be 10 or less.

**Maximum Number of Outstanding I-Format LPDUs (TW) and Number of I-format LPDUs Received before Sending Acknowledgment (N3)**

The TW and N3 counts should be considered together since they establish the ratio of acknowledgment frames to I-format LPDU frames. However, the N3 value should be compared only with the TW value of the remote link station, not the local station. The values of TW and N3 can affect the response perceived by the user in some cases. However, in most instances, the

default values will provide the best general performance. The following guidelines should be considered:

- The TW count will allow the sender to transmit TW frames before it is forced to halt and wait for an acknowledgment. Therefore, the receiver should be able to absorb that number of frames, either in its SAP buffers or within the buffers in PC system memory. A small value of TW will reduce the chances that frames will have to be retransmitted due to buffer congestion at the receiver. The TW/N3 ratio thus provides a flow control mechanism to prevent overruns at the receiver.
- The TW value should always be at least twice N3 value. Network response can be severely degraded if N3 exceeds TW.
- Very little network overhead or adapter processing is required to send or receive an acknowledgment frame. Therefore, every frame can be acknowledged without a perceptible degradation in performance.
- Even though the maximum values allowed for TW and N3 are 127 each, practical values should not exceed 8 for TW or 4 for N3.

**Note:** For more information, see the *IBM Token-Ring Network Architecture Reference*.

### **Working window (Ww), and Window increment (Nw)**

There are two counts associated with the dynamic window algorithm for flow control. The purpose of the dynamic window algorithm is to allow the sending station to temporarily reduce the transmit window (Tw) whenever network or receive adapter congestion is resulting in lost frames. By temporarily reducing the window size, the flow of frames over that link is reduced, thus permitting the congested node to recover from the temporary overload.

The DLC interface provides an interface to application programs using the Logical Link Control sublayer of Data Link Control protocol. The interface may be used in two ways.

- For IEEE type 1 communication: connectionless communication between devices providing no guarantee of delivery.
- For IEEE type 2 communication: connection-oriented services.

Much of the communication overhead function is provided by the adapter and the adapter support software, which permit simple programming. See the *IBM Token-Ring Network Architecture Reference* for more about communication using DLC and LLC.

---

## **Transmitting, Receiving, and Buffers**

Data exchanged between application programs is sent on the network in frames. A frame consists of headers and data. All frames have a LAN header, although the format is slightly different depending on whether the IBM Token-Ring Network or IBM PC Network is being used. MAC frames, transmitted only on the IBM Token-Ring Network, consist of the LAN header and a data field. In non-MAC frames the LAN header is followed by the DLC header. The data field that follows the LAN header in MAC frames, and the DLC header in non-MAC frames, is the data provided by the application program. (The data field itself may contain further headers, in a format defined by the application programs exchanging the data.)

The length of the LAN header varies depending on the length of the Routing Information field (if present). The length of the DLC header varies depending on the frame type: the I frames used to exchange data in Type 2 protocols have a four-byte DLC header; the frames used for Type 1 protocols have a three byte header.

The type of frame being transmitted also affects the amount of information that the application program must provide, and how it is provided. There are three cases:

- MAC frames:** The application program provides the complete LAN header and the data field. The Source Address in the LAN header will be overwritten with the address being used by the adapter.
- I frames:** The application program provides only the data when it wishes to transmit a frame. The information required to build the headers is provided in the DLC.OPEN.STATION and DLC.CONNECT.STATION commands.
- Other frames:** The application program provides the LAN header and the data in its own buffers. It provides the information needed to build the DLC header with the transmit command (the Remote SAP value and the command type). The Source Address field in the LAN header will be overwritten with the address in use by the adapter.

The application program must have the data, and, if required, the LAN header in buffers prepared in a format understood by the adapter support software. These buffers are in PC system memory belonging to the application program. This memory may be entirely controlled by the application program, or may be given to the adapter support software to manage, in which case it is part of the buffer pool discussed in "Buffer Pools."

Frames are transmitted onto the network from an area known as a Data Holding Buffer, or adapter transmit buffer. For both the IBM Token-Ring and PC Networks this is an area of the RAM on the adapter. The adapter transmit buffer is used differently for Type 2 I frames than it is for other frames: for I frames it holds only the user data part of the frame, the LAN and DLC headers are built in another area of adapter (or adapter support software) memory. For other frames the headers as well as the data have to fit in the transmit buffer. This difference affects the maximum length of a frame that can be transmitted.

Frames are received from the network into adapter memory. The adapter support software moves frames from adapter memory to application program memory, provided that there is application program buffer space to hold the incoming frame, and an active RECEIVE command to use to pass the buffers to the application program. The application program provides the buffer space in the form of a buffer pool, described below. Once the adapter support software has moved the data to PC system memory, it will use the RECEIVE command (and optionally the Read command for CCB2 users) to tell the application program that it has received a frame. See page 3-69 for the RECEIVE command and page 3-77 for the READ command.

## Maximum Frame Lengths

### Buffer Pools

A buffer pool is an area of PC system memory provided by the application program to the adapter support software. Each buffer pool is divided into buffers. Received frames are put into buffers from the buffer pool by the adapter support software. These buffers must be returned to the pool (using BUFFER.FREE commands) after the application program has finished with the frame data. When the application program transmits a frame it may use buffers from a separate area of memory, or buffers from the buffer pool (obtained by issuing a BUFFER.GET command). The following commands are associated with generating, defining, and handling buffer pools:

<b>DIR.OPEN.ADAPTER</b>	Allocates direct interface buffer pool (for CCB1 only)
<b>DIR.OPEN.DIRECT</b>	Allocates direct interface buffer pool (for CCB2 and CCB3 only)
<b>DIR.MODIFY.OPEN.PARMS</b>	Changes direct interface buffer pool allocation (for CCB1 only)
<b>DLC.OPEN.SAP</b>	Allocates DLC interface buffer pool for a specific SAP
<b>BUFFER.GET</b>	Gets one or more buffers from a SAP pool and DIRECT pool
<b>BUFFER.FREE</b>	Returns one or more buffers back into a pool
<b>RECEIVE</b>	Receives data into buffers
<b>RECEIVE.MODIFY</b>	Receives data into optional buffers
<b>TRANSMIT</b>	Sends data from buffers

Buffer pools can be allocated for every SAP defined to the adapter and for the direct interface direct station at STATION\_IDs X'0000', X'0001', and X'0002'. STATION X'0001' is not used on the PC Network. Every SAP defined may have one pool of buffers defined for its use.

Each buffer pool is independent of the others and has the following characteristics:

- The adapter support software uses these buffers to satisfy the RECEIVE command. Their use is optional for transmit commands.
- All link stations associated with a specific SAP use the same buffer pool.
- All buffers in a pool have the same length.
- Every buffer has a 12-byte overhead, to contain a forward pointer and length information controlled by the adapter support software.
- When defining a buffer pool:
  - The buffer length defined must be a multiple of 16 bytes.
  - The user-defined length includes the 12-byte overhead.
  - The minimum user-defined length is 80 bytes (68 data bytes plus 12 bytes of overhead).
- The application program may allow the adapter support software to prepare the buffer pool, or it may take that responsibility itself. Buffer pools are controlled by the adapter support software and individual buffers are obtained and returned by BUFFER.GET and BUFFER.FREE commands. If the application program controls the buffers, it must prepare the control fields in the prescribed format. Since the buffers controlled by the adapter support software must be used for receives, either buffers prepared by the adapter support software are used or the application program provides a prepared buffer to the adapter support software by issuing a BUFFER.FREE command.

**Note:** The impact of a SAP having no more available buffers may vary depending upon the data being received. See the RECEIVE command description.

## Receive Buffers

Data is received from the network into adapter buffers. If there is a RECEIVE command outstanding for the SAP or link station, the data is then moved by the adapter support software to the appropriate buffer pool in PC system memory. The application program then processes the data and issues a BUFFER.FREE command to return the buffer to the pool. More than one buffer can be returned by an application program simultaneously.

### Receive Buffer Formats

The USER\_OFFSET field, in bytes 8-9 of each buffer, allows all buffers to be handled similarly regardless of the amount of information in the buffer prior to the actual received data. When more than one buffer from the buffer pool is used to receive a frame, the format for the first buffer (BUFFER\_ONE) is different than the other buffers used to contain this frame. By interrogating the contents of the USER\_OFFSET field, the format of any buffer can be determined. See the RECEIVE command and the RECEIVE.MODIFY command descriptions for buffer assignment.

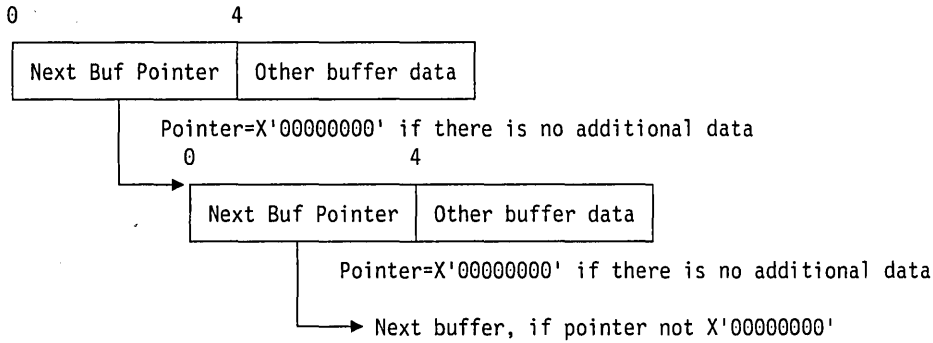


Figure 2-2. Receive Buffer Formats

### Buffer One: Option = Not Contiguous MAC/DATA

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	BUF_POINTER	4	DD	Pointer to the next buffer or X'00000000' if no additional buffers
4	RCV_LEN	2	DW	Length of entire receive frame
6	LENGTH_IN_BUFFER	2	DW	Length of data in buffer beginning at byte X (received data)
<b>For CCB1:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer segment-selector + offset.
<b>For CCB2 and CCB3:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer selector-selector + offset.
<b>For all CCBs:</b>				
10	USER_LENGTH	2	DW	The length of the USER_SPACE field defined by the USER_OFFSET parameter
12	STATION_ID	2	DW	Receiving station ID
14	OPTIONS	1	DB	Option byte from RECEIVE parameter table
15	MESSAGE_TYPE	1	DB	Type of message received
16	BUFFERS_LEFT	2	DW	The number of buffers left in the SAP buffer pool
18	RCV_FS	1	DB	Received Frame Status field
19	ADAPTER_NUM	1	DB	Adapter number (0 or 1)
<b>For CCB1:</b>				
20	LAN_HEADER_LENGTH	1	DB	The length of the LAN Header field (bytes 22-53)
21	DLC_HEADER_LENGTH	1	DB	The length of the DLC_HEADER (bytes 54-57); If the value is X'00' this is for the direct interface.
22	LAN_HEADER	32	DB	The LAN header received with the frame. The actual length is defined by LAN_HEADER_LENGTH.
54	DLC_HEADER	4	DB	The DLC header received with the frame, if applicable. The actual length is defined by DLC_HEADER_LENGTH. (Contents undefined if DLC LENGTH = 0.)
58	USER_SPACE	-	--	An area in the buffer for use by the application program. The length is defined by USER_LENGTH (bytes 10-11).
X	RCVD_DATA	-	DB	The data received following the DLC Header in the frame.
<b>For CCB2:</b>				
20	NEXT_FRAME	4	DD	A pointer to the next receive frame
<b>For CCB3:</b>				

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
20		4	DD	Reserved for the application program
<b>For CCB2 and CCB3:</b>				
24	LAN_HEADER_LENGTH	1	DB	The length of the LAN Header field (bytes 26-57)
25	DLC_HEADER_LENGTH	1	DB	The length of the DLC_HEADER (bytes 58-61): If the value is X'00' this is for the direct interface.
26	LAN_HEADER	32	DB	The LAN header received with the frame. The actual length is defined by LAN_HEADER_LENGTH.
58	DLC_HEADER	4	DB	The DLC header received with the frame, if applicable. The actual length is defined by DLC_HEADER_LENGTH. (Contents undefined if DLC LENGTH = 0.)
62	USER_SPACE	-	--	An area in the buffer for use by the application program. The length is defined by USER_LENGTH (bytes 10-11).
X	RCVD_DATA	-	DB	The data received following the DLC Header in the frame.

### Buffer One: Option = Contiguous MAC/DATA

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	BUF_POINTER	4	DD	Pointer to the next buffer or X'00000000' if no additional buffers
4	RCV_LEN	2	DW	Length of entire receive frame
6	LENGTH_IN_BUFFER	2	DW	Length of data in buffer beginning at byte X (received data)
<b>For CCB1:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer segment-selector + offset.
<b>For CCB2 and CCB3:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer selector-selector + offset.
<b>For all CCBs:</b>				
10	USER_LENGTH	2	DW	The length of the USER_SPACE field defined by the USER_OFFSET parameter
12	STATION_ID	2	DW	Receiving station ID
14	OPTIONS	1	DB	Option byte from RECEIVE parameter table
15	MESSAGE_TYPE	1	DB	Type of message received
16	BUFFERS_LEFT	2	DW	The number of buffers left in the SAP buffer pool
18	RCV_FS	1	DB	Received Frame Status field
19	ADAPTER_NUM	1	DB	Adapter number (0 or 1)
<b>For CCB1:</b>				
20	USER_SPACE	-	--	An area in the buffer for use by the application program. The length is defined by USER_LENGTH (bytes 10-11).
X	RCVD_DATA	-	DB	The data received in the frame including the LAN Header and the DLC Header.
<b>For CCB2:</b>				
20	NEXT_FRAME	4	DD	A pointer to the next receive frame
<b>For CCB3:</b>				
20		4	DD	Reserved for the application program
<b>For CCB2 and CCB3:</b>				

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
24	USER_SPACE	-	--	An area in the buffer for use by the application program. The length is defined by USER_LENGTH (bytes 10-11).
X	RCVD_DATA	-	DB	The data received in the frame including the LAN Header and the DLC Header.

## Explanation of Certain Buffer Fields

---

### MESSAGE\_TYPE

**Explanation:** (byte 15) Type of message received

- X'02' MAC frame (Direct Station on the Token-Ring Network only)
- X'04' I frame (Information frame - application program data - link stations only)
- X'06' UI frame
- X'08' XID command (poll bit)
- X'0A' XID command (not poll bit)
- X'0C' XID response (final bit)
- X'0E' XID response (not final bit)
- X'10' TEST response (final bit)
- X'12' TEST response (not final bit)
- X'14' Other, Used for non-MAC frame (Direct Station only)

---

### RCVD\_FS

**Explanation:** Frame Status (FS) byte (byte 18)

**Note:** Only valid on the Token-Ring Network.

#### **BIT MEANING**

- 7 Address recognized indicator (A)
- 6 Frame copied indicator (C)
- 5 Reserved
- 4 Reserved
- 3 Address recognized indicator (A)
- 2 Frame copied indicator (C)
- 1-0 Reserved

---

### NEXT\_FRAME

**Explanation:** Pointer to next frame in chain.

When the application program specifies that received frames are to be chained, the NEXT\_FRAME field of the first buffer of each frame is used to point to the next frame that was received.

---

### USER\_SPACE

**Explanation:** This space may be loaded by the application program. It is not altered by the adapter support software or by the received frame data.



---

## RCVD\_DATA

**Explanation:** (bytes X to end of buffer)

If option is CONTIGUOUS:

This data begins with the LAN Header from the received frame.

If option is NOT CONTIGUOUS:

- If MESSAGE\_TYPE is X'02' or X'14', this is the data immediately following the LAN Header from the received frame.
- If MESSAGE\_TYPE is not X'02' or X'14', this is the data immediately following the DLC Header from the received frame.

Additional data that will not fit into this buffer is placed in buffer two and subsequent buffers.

### Buffer Two and Subsequent Buffers

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	BUF_POINTER	4	DD	Pointer to the next buffer or X'00000000' if no additional buffers
4	RCV_LEN	2	DW	Length of entire receive frame
6	LENGTH_IN_BUFFER	2	DW	Length of data in buffer beginning at byte X (received data)
<b>For CCB1:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer segment-selector + offset.
<b>For CCB2 and CCB3:</b>				
8	USER_OFFSET	2	DW	Offset from the beginning of the buffer to the USER_SPACE field. Use this value with the buffer selector-selector + offset.
<b>For all CCBs:</b>				
10	USER_LENGTH	2	DW	The length of the USER_SPACE field defined by the USER_OFFSET parameter
12	USER_SPACE	-	--	An area in the buffer for use by the application program. The length is defined by USER_LENGTH (bytes 10-11).
X	RCVD_DATA	-	DB	A continuation of the data received in the frame

## Transmit Buffers

The application program issues a BUFFER.GET command, or creates a buffer, moves data into the assigned buffer and adds necessary header information, and issues the transmit command. The adapter support software moves the contents of the buffer into shared RAM and interrupts the adapter to proceed with the transmission. When the Transmit command has completed, the application program issues a BUFFER.FREE for all buffers originally obtained from the buffer pool except the XMIT\_QUEUE\_TWO buffer which is freed by the adapter support software when the transmission is successful (return code is zero).

The total amount of data in all buffers of one issued command must fit into one adapter transmit buffer in shared RAM.

**For CCB1:** The adapter transmit buffer size is defined by the DIR.OPEN.ADAPTER command DHB\_BUFFER\_LEN parameter, with the maximum being dependent on the type of adapter being used.

**For CCB2 and CCB3:** The adapter transmit buffer size is defined by configuration parameters, with the maximum being dependent on the type of adapter being used.

6 bytes are used as overhead

14 bytes are used for the access control (AC) byte, the frame control (FC) byte, and the LAN Header source and destination address fields.

The remaining length will be reduced if routing information is used (up to 18 bytes) and if a DLC Header is included (up to 4 bytes). For more information on buffers space, see Figures 2-4, 2-5, and 2-6 starting on page 2-42.

**Note:** The LAN and DLC headers are not placed in the transmit buffer for an I frame transmission as a result of a TRANSMIT.I.FRAME command making an additional 36 bytes available.

## Transmit Buffer Formats

The transmit buffers must be formatted as defined here.

Four groups of buffers are definable by transmit commands. See "Transmit Command Specifics" on page 3-91. They are:

- XMIT\_QUEUE\_ONE
- XMIT\_QUEUE\_TWO
- BUFFER\_ONE
- BUFFER\_TWO

XMIT\_QUEUE\_ONE and XMIT\_QUEUE\_TWO may each consist of one or more buffers.

Most combinations of XMIT\_QUEUE\_ONE, XMIT\_QUEUE\_TWO, BUFFER\_ONE, and BUFFER\_TWO can be selected for use. BUFFER\_TWO may be used only if BUFFER\_ONE is also being used. However, they will be transmitted sequentially beginning with XMIT\_QUEUE\_ONE and ending with BUFFER\_TWO whenever two or more are selected. XMIT\_QUEUE\_ONE could contain header information that seldom or never needs modifying. XMIT\_QUEUE\_TWO could contain data or device-specific header information. BUFFER\_ONE could contain the actual data to be transmitted. BUFFER\_TWO, if used, might contain additional data. Buffers in XMIT\_QUEUE\_TWO are freed by the adapter support software if the transmission is successful (return code is zero). These buffers are always returned when using Operating System/2, regardless of the return code.

BUFFER\_ONE and BUFFER\_TWO are user-defined and may contain any type of information. Any buffer group may be excluded by providing a buffer length of zero in the transmit command CCB.

The buffers defined by XMIT\_QUEUE\_ONE and XMIT\_QUEUE\_TWO are as follows:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	BUF_POINTER	4	DD	Pointer to the next buffer or X'00000000' if there are no additional buffers.
4		2	--	Reserved
6	LENGTH_IN_BUFFER	2	DW	Length of data in buffer beginning at byte 12 plus the USER_LENGTH.
8	USER_DATA	2	DW	Available for user
10	USER_LENGTH	2	DW	Length of the USER_SPACE starting at byte 12
12	USER_SPACE	2	DW	USER_SPACE followed by data to be transmitted. If there is no user data, the transmit data starts at byte 12.

The USER\_SPACE may be loaded by the application program. The USER\_SPACE information is not transmitted.

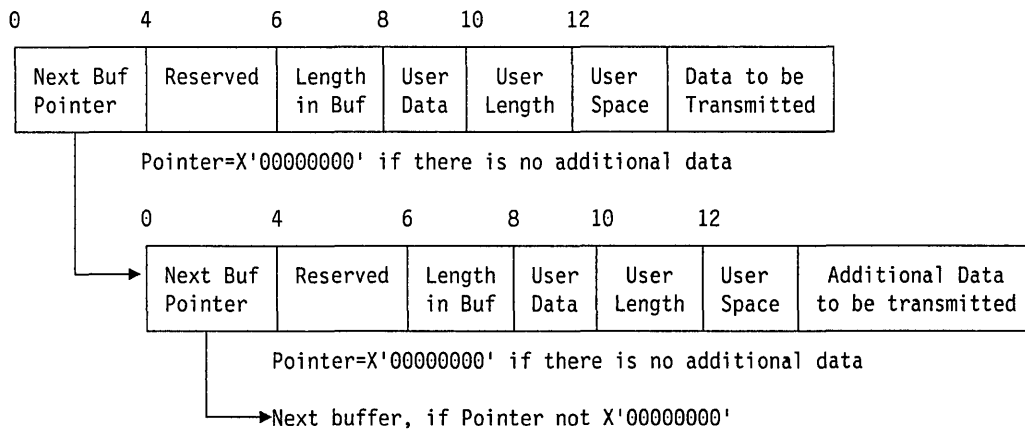


Figure 2-3. Transmit Buffers

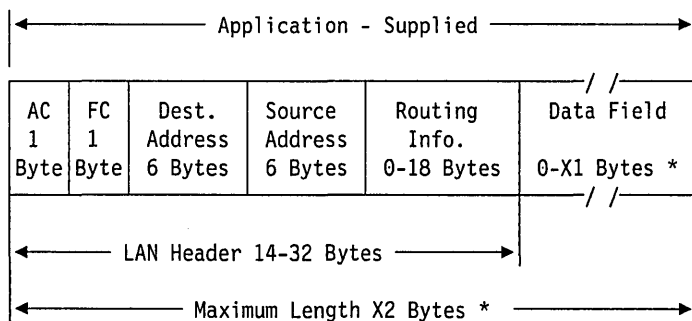


Figure 2-4. MAC frame.

\* See Table 2-8 on page 2-43 for the values of X1 and X2.

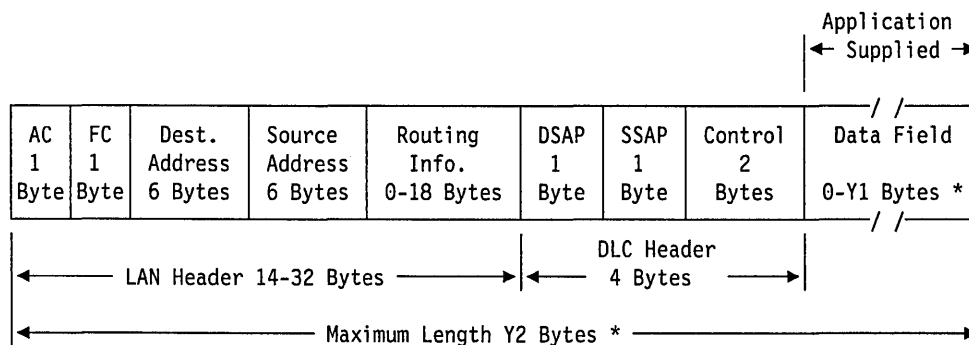


Figure 2-5. Non-MAC I frame.

\* See Table 2-8 on page 2-43 for the values of Y1 and Y2.

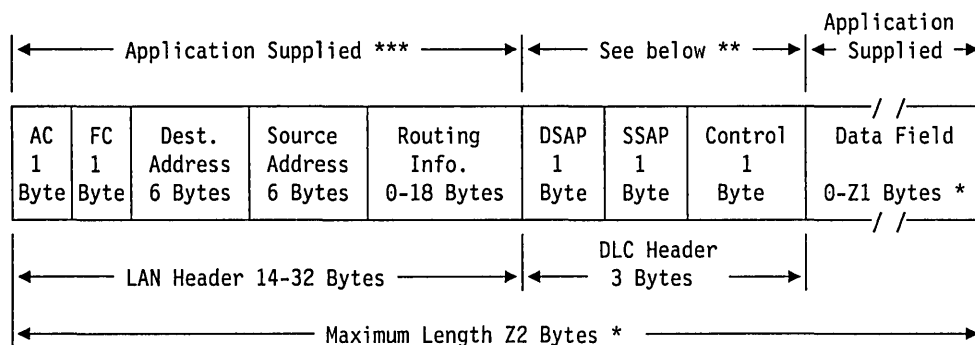


Figure 2-6. Other non-MAC frame.

\* See Table 2-8 on page 2-43 for the values of Z1 and Z2.

\*\* The adapter places the DLC header values in 3 bytes of the adapter transmit buffer. None of the application program's buffer space is needed for the DLC header.

\*\*\* The LAN header space, including the destination address and routing information fields, is provided by the application program in the first buffer. The adapter fills in the source address field values.

Value	Token-Ring Network			PC Network Adapter
	PC Adapter, PC Adapter II Adapter/A	16/4 adapters at 4 Mbps	16/4 adapters at 16 Mbps	
X1	2028	4444	17940	2028
X2	2042	4458	17954	2042
Y1	2042	4458	17954	2042
Y2	2078	4494	17990	2078
Z1	2025	4441	17937	2028
Z2	2042	4458	17954	2045

## Calculating RAM and Work Area Usage

The descriptions for calculating RAM and work area usage are subdivided under the following headings: Token-Ring Network Using DOS; Token-Ring Network Using Operating System/2 Extended Edition 1.1; PC Network; and NETBIOS.

### Calculations for the Token-Ring Network Using DOS

**Note:** The adapter support software discussed in this section refers to the adapter support device driver provided with the Local Area Network Support Program.

The adapter support software provides 1KB of work area for each adapter. During initialization it returns to the operating system (DOS) 1KB of memory used for initialization. If only one adapter is present, it returns the 1KB that would have been assigned to the additional adapter as work area.

**Note:** In a PC3270 Work Station Program environment, the adapter support software provides 4KB of work area for each adapter.

The assignment of adapter support software work area is as follows:

Bytes	Function
48	The direct interface
36	For each SAP
6	For each link station

**Note:** In a PC3270 Work Station Program environment, each SAP requires 110 bytes rather than 36.

The application program can assign work area by setting parameters in the DIR.OPEN.ADAPTER command.

The DIR.OPEN.ADAPTER command defines functions that require space in shared RAM. The remainder of shared RAM is configured for receive buffers.

The assignment of shared RAM is as follows:

Bytes	Function
1588	For adapter work and communication area
64	For each SAP
14*	For each group SAP
144	For each link station
96 to 17960	For each adapter transmit buffer defined by the application program
* Plus 2 times (the maximum group SAP members)	

For example, 16KB of shared RAM on a Token-Ring Network PC Adapter II would support a configuration of:

- 64 link stations with 8 SAPs
- A default selection of 600 bytes for the DHB.

This leaves 4468 bytes for receive buffers.

Similar calculations can be made for 8K, 32K, and 64KB of Shared RAM.

## Calculations for the Token-Ring Network Using OS/2

**Note:** The adapter support software discussed in this section refers to the Communications Manager provided with the Operating System/2 Extended Edition 1.1.

The adapter support software can provide up to 60KB of work area for either a single adapter or two adapters. The size of these work areas is dependent on configuration data and, when combined, can take up to 60KB. The work areas are for user application program, control blocks, SAPs, and link stations defined by the configuration parameters. See Appendix E, "Operating System/2 Extended Edition 1.1 Information" on page E-1 for more information on configuration parameters. In addition, a single pool of queue elements is also created from the 60K byte work area. The work area described is located in the data segment of the adapter support softwares device driver. If all of the data segment is not needed, the adapter support software will adjust its data segment to accommodate a smaller work area.

The assignment of adapter support software work area is as follows:

Bytes	Function
210	For each adapter
104	For each application program
106	For direct stations
62	For each SAP
22	For each link station

The application program can assign work areas by setting the configuration parameters at initialization time.

The configuration parameters define resources that require space in shared RAM. The remainder of shared RAM is configured for receive buffers.

The assignment of shared RAM is as follows:

Bytes	Function
1588	For adapter work and communication area
64	For each SAP
14*	For each group SAP
144	For each link station
96 to 17960	For each adapter transmit buffer defined by the application program
* Plus 2 times (the maximum group SAP members)	

For example, 16KB of shared RAM on a Token-Ring Network PC Adapter II would support a configuration of:

- 64 link stations with 8 SAPs
- A default selection of 600 bytes for the DHB.

This leaves 4468 bytes for receive buffers.

Similar calculations can be made for 8K, 32K, and 64KB of Shared RAM.

## Calculations for the PC Network Using DOS

The Local Area Network Support Program software contains a dynamic work area. The amount of dynamic work area can be specified by the user at load time up to a maximum of 64KB per adapter. If the user does not specify a work area size, the Local Area Network Support Program will allocate a default work area of 8KB.

This work area is used by the Local Area Network Support Program to maintain SAPs and link stations as follows:

Bytes	Function
278	Static Overhead
108	Each Individual SAP
14*	Each Group SAP
150	Each Link Station
* Add 2 times (Maximum group SAP members)	

**Note:** In a PC3270 environment, each SAP requires 110 bytes rather than 108.

This work area is configured by the Local Area Network Support Program as a result of a DIR.OPEN.ADAPTER command. The remainder of this work area is configured for receive buffers. There must be enough memory available for a minimum of 2 receive buffers.

## Calculations for the PC Network Using OS/2

The adapter support software includes a common work area (60 KB). In addition, two other work areas (maximum of 64 KB each) area created. The size of these work areas is dependent on configuration data and when combined with other work areas can take up to 188 KB. The adapter support software requests Operating System/2 to allocate RAM for each work space to contain the Link Station Control Blocks, SAP Control Blocks and Group SAP Control Blocks. The number of bytes to allocate is optionally specified by configuration parameters. If this parameter is omitted, 8KB (default) per adapter is allocated. During initialization, if only one adapter is present, memory for only one work area will be requested from Operating System/2, while other work areas will be returned.

This work area is used by the adapter support software to maintain SAPs and link stations as follows:

Bytes	Function
744	Static Overhead
72	Each Individual SAP
14*	Each Group SAP
144	Each Link Station
* Add 2 (Maximum group SAP members)	

## Calculations for the NETBIOS Interface

NETBIOS provides 9KB of work area for each adapter. When NETBIOS is loaded, assignment of this work area may be altered. As with the adapter support software, the application program can assign work area by setting parameters in the DIR.OPEN.ADAPTER command.

The assignment of NETBIOS work area is as follows:

Table 2-9. NETBIOS Data Area Storage Requirements - DOS	
Bytes	Function
4400	NETBIOS work area
20	Each name defined at load time
50	Each link station defined at load time
50	Each session defined at load time
100	Each NCB defined at load time
70	Each RND entry defined at load time

Table 2-10. NETBIOS Data Area Storage Requirements - OS/2	
Bytes	Function
5000	NETBIOS work area
25	Each name is defined by configuration parameters
50	Each link station is defined by configuration parameters
50	Each session is defined by configuration parameters
200	Each NCB is defined by configuration parameters
70	Each RND entry is defined by configuration parameters





---

## Chapter 3. The Command Control Blocks

About This Chapter	3-2
Command Descriptions	3-3
BUFFER.FREE	3-3
BUFFER.GET	3-4
DLC.CLOSE.SAP	3-5
DLC.CLOSE.STATION	3-6
DLC.CONNECT.STATION	3-7
DLC.FLOW.CONTROL	3-8
DLC.MODIFY	3-9
DLC.OPEN.SAP	3-11
DLC.OPEN.STATION	3-16
DLC.REALLOCATE	3-19
DLC.RESET	3-20
DLC.SET.TRESHOLD	3-22
DLC.STATISTICS	3-23
Log Formats	3-24
DIR.CLOSE.ADAPTER	3-25
DIR.CLOSE.DIRECT	3-27
DIR.DEFINE.MIF.ENVIRONMENT	3-27
DIR.INITIALIZE	3-29
DIR.INTERRUPT	3-33
DIR.MODIFY.OPEN.PARMS	3-33
DIR.OPEN.ADAPTER	3-34
DIR.OPEN.DIRECT	3-46
DIR.READ.LOG	3-47
DIR.RESTORE.OPEN.PARMS	3-50
DIR.SET.EXCEPTION.FLAGS	3-50
DIR.SET.FUNCTIONAL.ADDRESS	3-52
DIR.SET.GROUP.ADDRESS	3-52
DIR.SET.USER.APPENDAGE	3-53
DIR.STATUS	3-54
DIR.TIMER.CANCEL	3-57
DIR.TIMER.CANCEL.GROUP	3-58
DIR.TIMER.SET	3-59
PDT.TRACE.ON	3-59
Trace Table Formats for Token-Ring Network	3-61
PDT.TRACE.OFF	3-67
PURGE.RESOURCE	3-68
READ	3-69
DLC Status Change Events	3-76
READ.CANCEL	3-77
RECEIVE	3-77
RECEIVE.CANCEL	3-83
RECEIVE.MODIFY	3-84
TRANSMIT.DIR.FRAME	3-88
TRANSMIT.I.FRAME	3-88
TRANSMIT.TEST.CMD	3-89
TRANSMIT.UI.FRAME	3-90
TRANSMIT.XID.CMD	3-90
TRANSMIT.XID.RESP.FINAL	3-90
TRANSMIT.XID.RESP.NOT.FINAL	3-91
Transmit Command Specifics	3-91

---

## About This Chapter

This chapter describes all the commands that can be issued to the adapter support software. The commands are listed alphabetically, so that a particular command can be found easily.

The commands can be regarded as falling into one of three groups based on which version of the adapter support software is being used.

- CCB1** The Command Control Block for the IEEE 802.2 adapter support software provided with the original Token-Ring Network PC Adapter, Token-Ring Network PC Adapter II, and the IEEE 802.2 adapter support software supplied with the Local Area Network Support Program.
- CCB2** The Command Control Block for the Dynamic Link Routine Interface provided with the Operating System/2 Extended Edition 1.1.
- CCB3** The Command Control Block for the Device Driver Interface provided with the Operating System/2 Extended Edition 1.1.

Throughout this chapter the term **CCB** will be used when information is common to all three groups. Detailed information on using CCBs with the different versions of the adapter support software may be found in "Adapter Support Software Control Blocks for All CCBs" on page 2-17.

The commands can also be considered to fall into four functional groups:

- Commands used at the direct interface
- Commands used at the DLC (IEEE 802.2) Interface - both SAP and station interfaces
- Commands used for transmitting and receiving frames
- Commands used for problem determination (CCB1 only)

The direct interface permits control functions to be performed on the adapter using standard control blocks and parameters. It provides the ability to open and close an adapter, obtain error status, and set addresses.

**Note:** When using Operating System/2 Extended Edition 1.1, the direct interface commands DIR.OPEN.ADAPTER and DIR.CLOSE.ADAPTER logically open and close an adapter on an application program basis. A System Key must be used to physically close an adapter.

This interface also permits transmission of frames directly with no protocol assistance. When using the direct interface, an application program can communicate with another application program without links and link stations. The direct interface supports three direct stations as discussed in "SAPs, Stations, and IDs" on page 2-24. All received frames not directed to an active SAP or link station default to the direct station.

The DLC interface is provided for application programs using the Logical Link Control sublayer of Data Link Control protocol. It provides both connectionless and connection-oriented services. See "DLC" on page 2-27 for more details.

Information on the use of the transmit and receive commands, buffer pools, and buffer formats may be found in "Transmitting, Receiving, and Buffers" on page 2-34.

---

## Command Descriptions

All the commands use a control block, as described in Chapter 2, "How to Write Programs for Local Area Networks" on page 2-1. All differing uses of variables in the control block and additional control information, such as parameter tables, are included with these command descriptions. The possible return codes for each command are listed in tables in Appendix B, "Return Codes," see "CCB Return Codes Listed by Command" on page B-4.

Each command description begins with a box containing the command name. The hexadecimal number at the top of the box is the command code value. Whenever parameter tables are included, descriptions of the parameters follow the table.

### BUFFER.FREE

Hex 27

**BUFFER.FREE**

**Command Description:** This command returns one or more buffers to the SAP's buffer pool or the DIRECT station buffer pool.

**Command Specifics:** When the buffer is placed back in the buffer pool, bytes 4 and 5 (buffer length) of the buffer are set to zero. This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**For CCB1 and CCB3:** The command completion appendage will be taken if provided.

**For CCB2:** Either a semaphore or a READ command can be used for command completion.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	SAP/DIRECT station ID: defines the buffer pool
2	BUFFER_LEFT	2	DW	Number of buffers left in the pool *
4		4	DB	Reserved
8	FIRST_BUFFER	4	DD	Address of the first buffer to be added to the pool. The value is set to zero on return.
* Indicates a returned value.				

---

#### STATION\_ID

**Explanation:** Defines the SAP that the buffer is currently assigned to. The SAP\_NUMBER portion of the STATION\_ID must identify a valid opened SAP or X'00' (DIRECT STATION), the STATION\_NUMBER portion is ignored.

---

#### BUFFER\_LEFT

**Explanation:** Defines the number of buffers in the pool after the command has been completed. The adapter support software will return the value when the command is completed.

## BUFFER.GET

---

### FIRST\_BUFFER

**Explanation:** The address of the first buffer to be added to the pool. If this value is zero, no buffer will be freed and the command will be completed with a CCB\_RETCODE of X'00'.

## BUFFER.GET

Hex 26

### BUFFER.GET

**Command Description:** This command gets one or more buffers from the SAP's buffer pool or the DIRECT station buffer pool.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**For CCB1 and CCB3:** The command completion appendage will be taken if provided.

**For CCB2:** Either a semaphore or a READ command can be used for command completion.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	SAP/DIRECT station ID: defines the buffer pool
2	BUFFER_LEFT	2	DW	Number of buffers left in the pool *
4	BUFFER_GET	1	DB	Number of buffers to get
5		3	DB	Reserved
8	FIRST_BUFFER	4	DD	Address of first buffer obtained *
* Indicates a returned value.				

---

### STATION\_ID

**Explanation:** Defines the SAP buffer pool that the buffer is to be taken from. The SAP\_NUMBER portion of the STATION\_ID must identify a valid opened SAP or X'00' (DIRECT STATION), the STATION\_NUMBER portion is ignored.

---

### BUFFER\_LEFT

**Explanation:** Defines the number of buffers in the pool after the command has been completed. The adapter support software will return the value when the command is completed.

---

### BUFFER\_GET

**Explanation:** This defines the number of buffers to get from the pool. If there is an inadequate number of buffers in the pool, the command will terminate with a CCB\_RETCODE of X'19'. If the value is set to 0, the default of 1 is used.

**Note:** This command could cause a link station to go into a "local busy" condition, if too many buffers are taken.

**FIRST\_BUFFER**

**Explanation:** The address of the first buffer that was obtained. The adapter will return the value when the command is completed. If no buffers are obtained, this field will be set to X'00000000'.

**DLC.CLOSE.SAP**

Hex 16

**DLC.CLOSE.SAP**

**Command Description:** This command closes (deactivate) a Service Access Point (SAP).

**Command Specifics:** If any station associated with the SAP is open, the command will terminate with a CCB\_RETCODE of X'47', SAP cannot close unless all link stations are closed.

If a 47 error code results when a DLC.CLOSE.SAP command closely follows a DLC.CLOSE.STATION command for the last open station for that SAP, reissue the DLC.CLOSE.SAP command.

If a RECEIVE command is outstanding for the SAP, it will terminate with an X'0A' return code and the RECEIVE command's CCB address will be placed in the CCB\_POINTER field of this command CCB.

**For CCB1:** The STATION\_ID of the SAP to be closed is placed in the first 2 bytes of the CCB\_PARM\_TAB field and the second two bytes are reserved.

If a RECEIVE command is outstanding, the command completion appendage of the RECEIVE command will not be taken, but the CCB of the RECEIVE command will be returned to the application program in the CCB\_POINTER field of the DLC.CLOSE.SAP command.

**For CCB2:** The STATION ID of the SAP to be closed is placed in the CCB\_PARM\_OFFSET field. If a LINK STATION is still open for this SAP, the STATION\_ID (of the LINK STATION not closed) is returned in the CCB\_PARAMETER1 field.

If a RECEIVE command is outstanding for the SAP, it is not put on the completion list. A semaphore is not cleared to post its command completion. If the application program wants to receive pointers to all outstanding CCBs for this SAP, the CCB\_CMPL\_FLAG of the DLC.CLOSE.SAP command must be set. When a READ command is issued or if one is outstanding requesting notification of command completions, it will post the completed DLC.CLOSE.SAP command. The outstanding data area pointers will be returned in the parameter table of the READ command. If a READ command is not already outstanding or if one is not chained to the CCB\_POINTER field to post the DLC.CLOSE.SAP when it completes, the DLC.CLOSE.SAP command completion event is placed on a completion queue. Since the SAP has been closed due to this command completing successfully, the only way to retrieve the event is by issuing a READ command with the OPTION\_INDICATOR field set to match on all events. This event cannot be retrieved by posting a READ with the OPTION\_INDICATOR set to match on the SAP that was closed, since it is no longer valid.

**For CCB3:** The STATION ID of the SAP to be closed is placed in the CCB\_PARM\_OFFSET field. If a LINK STATION is still open for this SAP, the STATION\_ID (of the LINK STATION not closed) is returned in the CCB\_PARAMETER1 field.

If a RECEIVE command is outstanding, the command completion appendage of the RECEIVE command will not be taken, but the CCB of the RECEIVE command will be returned to the application program in the CCB\_POINTER field of the DLC.CLOSE.SAP command. In addition to the RECEIVE command, buffers from the SAP buffer pool will also be return in the information table.

## DLC.CLOSE.STATION

Valid Return Codes: see "CCB Return Codes Listed by Command" on page B-4.

## DLC.CLOSE.STATION

Hex 1A

DLC.CLOSE.STATION

**Command Description:** This command closes (deactivate) a link station.

**Command Specifics:** If this command is issued while the ring is beaoning, it may not complete until the ring is no longer beaoning. See the *IBM Token-Ring Network Architecture Reference* for a description of ring beaoning.

**For CCB1:** The STATION\_ID of the link station to be closed is placed in the first 2 bytes of the CCB\_PARM\_TAB field and the second two bytes are reserved.

If a receive command is outstanding for this link station, it will be terminated with a CCB\_RETCODE of X'0A' and its address will be placed in the CCB\_POINTER of the DLC.CLOSE.STATION command.

The CCB\_CMD\_CMPL appendage of the receive command will not be taken. Any pending transmit commands will be aborted immediately.

**For CCB2:** The STATION ID of the link station to be closed is placed in the CCB\_PARM\_OFFSET field.

If a receive command is outstanding for this link station, it will be terminated with a CCB\_RETCODE of X'0A' and its address will be placed in the CCB\_POINTER of the DLC.CLOSE.STATION command.

There will be no command completion notification done for the RECEIVE other than setting the return code. If the application program wants to receive pointers to all outstanding CCBs for this station, the CCB\_CMPL\_FLAG of the DLC.CLOSE.STATION command must be set. When a READ command is issued or if one is outstanding requesting notification of command completions, it will post the completed DLC.CLOSE.STATION command. The outstanding data area pointers will be returned in the parameter table of the READ command. If a READ command is not already outstanding or if one is not chained to the CCB\_POINTER field to post the DLC.CLOSE.STATION when it completes, the DLC.CLOSE.STATION command completion event is placed on a completion list. Since the link station has been closed due to this command completing successfully, the only way to retrieve the event is by issuing a READ command with the OPTION\_INDICATOR field set to match on all events or to match on the SAP that owned the link station. This event cannot be retrieved by posting a READ with the OPTION\_INDICATOR set to match on the link station that was closed, since it is no longer valid.

**For CCB3:** The STATION ID of the link station to be closed is placed in the CCB\_PARM\_OFFSET field.

If a RECEIVE command is outstanding, the command completion appendage of the RECEIVE command will not be taken, but the CCB of the RECEIVE command will be returned to the application program in the CCB\_POINTER field of the DLC.CLOSE.SAP command. When a DLC.CLOSE.STATION command is issued, the RECEIVE command associated with that station will be terminated with a CCB\_RETCODE of X'0A' and the CCB of the RECEIVE command will be returned to the application program in the information table passed to the completion appendage of the DLC.CLOSE.STATION command.

Valid Return Codes: see "CCB Return Codes Listed by Command" on page B-4.

## DLC.CONNECT.STATION

Hex 1B

DLC.CONNECT.STATION
---------------------

**Command Description:** This command starts or completes a SABME-UA exchange to place both the local and remote link stations in a data transfer state.

**Command Specifics:**

**For CCB1:** The CCB\_PARM\_TAB points to the parameter table.

**For CCB2 and CCB3:** The CCB\_PARM\_OFFSET is the offset to the parameter table.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

**CCB Parameter Table**

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	Link station ID to be connected
2		2	DB	Reserved
<b>For CCB1:</b>				
4	ROUTING_ADDR	4	DD	Address in PC system memory of 18 bytes of network routing information
<b>For CCB2 and CCB3:</b>				
4	ROUTING_OFFSET	2	DW	Offset in PC system memory of 18 bytes of network routing information
6		2	DW	Reserved for the application program

---

### STATION\_ID

**Explanation:** The station ID of the link station.

---

### ROUTING\_ADDR and ROUTING\_OFFSET

**Explanation:** This is the address or offset to a memory location where the routing information is found. If the remote partner for this link station is on a different ring, routing information is necessary for frames to be exchanged. If the link station has been established because of a DLC.OPEN.STATION command, the routing information must be provided with this command. If the link station was established because of receipt of a SABME from the remote partner, the adapter will obtain the routing information from the received frame and ignore any provided with this command. The DLC.CONNECT.STATION command may also be used to provide new routing information if there is a link failure. The information must be provided in the format in which it will be used in transmitted frames. If this field is set to zero, or if the length field of the routing information field is zero and no SABME is outstanding, the remote partner will be assumed to be on the same ring. See the description of bit 10 in “DLC Status Codes” on page B-25. Refer to the *IBM Token-Ring Network Architecture Reference* for more about routing information and XID. Also, refer to any documentation related to implementation by bridges in your network.



## DLC.FLOW.CONTROL

Hex 1D

### DLC.FLOW.CONTROL

**Command Description:** This command controls the flow of data across a specified link station on a SAP, or every link station on a SAP by setting and resetting a local busy status. See "Link Station States" on page 2-29.

**Command Specifics:**

**For CCB1:** The CCB\_PARM\_TAB field contains the STATION\_ID and the FLOW\_CONTROL option byte. The STATION\_ID is the first two bytes, the FLOW\_CONTROL (option byte) is the third byte, and the last byte is reserved.

**For CCB2 and CCB3:** The CCB\_PARM\_OFFSET field contains the STATION\_ID. The CCB\_PARAMETER\_1 field contains the FLOW\_CONTROL (option byte); the FLOW\_CONTROL (option byte) is the first byte, the second byte is reserved.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

---

### STATION\_ID

**Explanation:** The first 2 bytes of the CCB\_PARM\_TAB parameter contain either the STATION\_ID of a specific link station of a SAP or the STATION\_ID of a SAP. If the ID is a SAP ID, all the link stations on the SAP are affected. If the ID is a link station ID, only that specific station is controlled.

---

### FLOW\_CONTROL

**Explanation:** Contains bits that define options. Bit 7 is the high-order bit (leftmost bit position).

- Bit 7: Used to set/reset a local busy state
  - If this bit is off (0), then the link station will enter the "local busy" state (Bit 6 is ignored).
  - If this bit is on (1), then the local busy state is reset based on the condition of bit 6.
- Bit 6: Used to indicate the type of "local busy" state which is being reset (bit 7 = 1).
  - If this bit is off (0), it indicates a "user-set" local busy state is to be reset.
  - If this bit is on (1), it indicates a local busy state caused by either an "out-of-receive-buffers" state, or "no receive command outstanding" state, will be reset.
- Bits 0 - 5: Reserved.

Should be set to zeros, but are not checked by the adapter.

## DLC.MODIFY

Hex 1C

DLC.MODIFY

**Command Description:** This command modifies certain work values of an open link station or the default values of a SAP.

**Command Specifics:** This command allows altering the values without the need to close and reestablish the SAP and links. The values to be modified are contained in the parameter table referenced by the following field.

**For CCB1:** The CCB\_PARM\_TAB field points to the parameter table.

**For CCB2 and CCB3:** The CCB\_PARM\_OFFSET field points to the parameter table.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0		2	DW	Reserved
2	STATION_ID	2	DW	SAP station or link station ID
4	TIMER_T1	1	DB	T1 value (response timer)
5	TIMER_T2	1	DB	T2 value (acknowledgment timer)
6	TIMER_TI	1	DB	Ti value (inactivity timer)
7	MAXOUT	1	DB	Maximum transmits without a receive acknowledgment
8	MAXIN	1	DB	Maximum receives without a transmit acknowledgment
9	MAXOUT_INCR	1	DB	Dynamic window increment value
10	MAX_RETRY_CNT	1	DB	N2 value
11		3	DB	Reserved
14	ACCESS_PRIORITY	1	DB	Ring access priority. Access priority is not recognized on the PC Network.
15		4	DB	Reserved
19	GROUP_COUNT	1	DB	Length of data in GROUP_LIST or GROUP_LIST_OFFSET
<b>For CCB1:</b>				
20	GROUP_LIST	4	DD	Address of a list of Group SAP values
<b>For CCB2 and CCB3:</b>				
20	GROUP_LIST_OFFSET	2	DW	Offset to a list of Group SAP values
22		2	DW	Reserved for the application program

### STATION\_ID

**Explanation:** The link station ID whose working values are to be changed or the SAP ID whose defaults are to be changed.

### TIMER\_T1

**Explanation:** Specifies the time period between 1 and 10 used to determine an inoperative condition on a link. The time intervals are based on the tick values specified by the DIR.OPEN.ADAPTER command or by configuration parameters. If the value is zero, the current value will remain in effect. See "Timers" on page 2-31 for a complete description of timers.

---

### TIMER\_T2

**Explanation:** Specifies the timer value for the T2 timer used to delay transmission of an acknowledgment for a received I-LPDU for a link station being modified in this SAP. The time intervals are based on the tick values specified by the DIR.OPEN.ADAPTER command or by configuration parameters. If the value is zero, the current value will remain in effect. If the value is greater than 10, the timer will not be used. See "Timers" on page 2-31 for a complete description of timers.

---

### TIMER\_Ti

**Explanation:** Specifies the time period between 1 and 10 used to determine an inactive condition on a link. The time intervals are defined by the DIR.OPEN.ADAPTER command or by configuration parameters. If the value is zero, the current value will remain in effect.

---

### MAXOUT

**Explanation:** Specifies the maximum number of sequentially numbered transmitted I-LPDUs that a link station associated with this SAP may have outstanding at any one time. The maximum valid value is 127. If the value is zero, the current value will remain in effect.

---

### MAXIN

**Explanation:** Specifies the maximum number of sequentially numbered received I-LPDUs that a link station associated with this SAP may receive prior to sending an acknowledgment. The maximum valid value is 127. If the value is zero, the current value will remain in effect.

---

### MAXOUT\_INCR

**Explanation:** This dynamic window increment value is used to reduce bridge congestion. If the two end points of the link are on different rings, and the adapter detects an error condition requiring retransmission, the MAXOUT parameter will be set to 1. It will then be incremented by 1 each time MAXOUT\_INCR frames are acknowledged by the remote station, until it reaches the application program-requested value. For more details, see the *IBM Token-Ring Network Architecture Reference*. If the value is zero, the current value will remain in effect.

---

### MAX\_RETRY\_CNT

**Explanation:** Specifies the number of retries for an unacknowledged command LPDU, or in the case of an I-LPDU timeout, the number of times that the non-responding remote link station will be polled with an RR/RNR command LPDU. This count is used in conjunction with the Response Timer and should be great enough to ensure time for ring error detection and recovery. This parameter also prevents continual retransmission of the same I frame. The maximum valid value is 255. If the value is zero, the current value will remain in effect.

---

---

**ACCESS\_PRIORITY**

**Explanation:** The transmit access priority value to be placed in the AC byte of all transmissions from the SAP or link station. The format is B'nnn00000', where 'nnn' is the access priority value. If the access priority is higher than authorized for the adapter, the command will terminate with a CCB\_RETCODE of X'08'. Valid values for the access priority are from 0 to 3. Access priority is not recognized on the PC Network.

---

**GROUP\_COUNT**

**Explanation:** The number, from 0 to 13, of group SAPs as defined by the GROUP\_LIST or GROUP\_LIST\_OFFSET field.

---

**GROUP\_LIST or GROUP\_LIST\_OFFSET**

**Explanation:** This field may be used either to request membership in additional group SAPs for an individual SAP, or to request that membership be canceled. The GROUP\_COUNT parameter indicates the number of valid values in this field. If the low-order bit of a SAP value is zero, membership in the corresponding group is requested. If the low-order bit of a SAP value is one, membership is canceled.

This field is ignored if the GROUP\_COUNT parameter is zero.

**DLC.OPEN.SAP**

Hex 15

DLC.OPEN.SAP

**Command Description:** This command activates a SAP and reserves a number of link stations for the SAP.

**Command Specifics:** This command can be used to define:

- An individual SAP
- A group SAP
- A SAP as a member of a group.

The application program is responsible for checking that the parameters are reasonable.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

If the return code is X'45' or X'49', the SAP has been opened, but there has been some problem with the GROUP\_LIST or GROUP\_LIST\_OFFSET parameter.

An open for an individual SAP requesting Group SAP membership for a group SAP can cause a return code of X'06' if the membership is requested in group SAPs with different types. See the bits described in the OPTIONS\_PRIORITY field. If this happens, the SAP will open with membership into the group SAPs specified until the SAP type of the Group SAP listed changes. All subsequent group SAPs listed after a SAP type change has been detected will not be used for membership.

CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	SAP station ID (X'nn00') *
2	USER_STAT_VALUE	2	DW	User value passed back on DLC status
4	TIMER_T1	1	DB	T1 value (response timer)
5	TIMER_T2	1	DB	T2 value (acknowledgment timer)
6	TIMER_TI	1	DB	Ti value (inactivity timer)
7	MAXOUT	1	DB	Maximum transmits without a receive acknowledgment
8	MAXIN	1	DB	Maximum receives without a transmit acknowledgment
9	MAXOUT_INCR	1	DB	Dynamic window increment value
10	MAX_RETRY_CNT	1	DB	N2 value
11	MAX_MEMBERS	1	DB	Maximum SAPs for a group SAP
12	MAX_I_FIELD	2	DW	Maximum received information field
14	SAP_VALUE	1	DB	SAP value to be assigned
15	OPTIONS_PRIORITY	1	DB	SAP options and ring access priority. Access priority is not recognized on the PC Network.
16	STATION_COUNT	1	DB	Number of link stations to reserve
17		2	DB	Reserved
19	GROUP_COUNT	1	DB	Length of data in GROUP_LIST or GROUP_LIST_OFFSET
<b>For CCB1:</b>				
20	GROUP_LIST	4	DD	Address of a list of Group SAP values
24	DLC_STATUS_EXIT	4	DD	I/O appendage exit, DLC status change
<b>For CCB2 and CCB3:</b>				
20	GROUP_LIST_OFFSET	2	DW	Offset to a list of Group SAP values
22		2	DW	Reserved for the application program
<b>For CCB2:</b>				
24	DLC_STATUS_FLAG	4	DD	User notification flag for DLC status changes
<b>For CCB3:</b>				
24	DLC_STATUS_OFFSET	2	DW	Offset of DLC status appendage
26		2	DW	Reserved for the application program
<b>For all CCBs:</b>				
28	DLC_BUF_SIZE	2	DW	Size of buffers in pool
30	DLC_POOL_LEN	2	DW	Length of pool buffer
32	DLC_POOL_ADDR	4	DD	Starting address of buffer pool
<b>For CCB2 and CCB3:</b>				
36	ADAPTER_STNS_AVAIL	1	DB	Number of link stations available *
* Indicates a returned value.				

**STATION\_ID**

**Explanation:** The station ID returned by the adapter or the adapter support software. This value is used to identify this SAP in subsequent commands.

---

**USER\_STAT\_VALUE****Explanation:**

- For CCB1:** On entry to the DLC status appendage (defined by parameter `DLC_STATUS_EXIT`), this value is passed back to the user in register SI.
- For CCB2:** User value, passed back on DLC status change notification using the READ command's CCB parameter table.
- For CCB3:** User value, passed back in register SI when calling application program device driver entry point for the DLC status change appendage.
- 

**TIMER\_T1**

**Explanation:** Specifies the time period between 1 and 10 used to determine an inoperative condition on a link. If the value is zero, the default of 5 is used. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by configuration parameters.

---

**TIMER\_T2**

**Explanation:** Specifies the time period between 1 and 10 used to delay transmission of an acknowledgment for a received I-LPDU for a link station in this SAP. If the value is zero, the default of 2 is used. If the value is greater than 10, the timer will not be used. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by configuration parameters.

---

**TIMER\_Ti**

**Explanation:** Specifies the time period between 1 and 10 used to determine an inactive condition on a link. If the value is zero, the default of 3 is used. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by configuration parameters.

---

**MAXOUT**

**Explanation:** Specifies the maximum number of sequentially numbered transmitted I-LPDUs that the link station(s) using this SAP may have outstanding at any time. The maximum valid value is 127. If the value is zero, the default of 2 is used.

---

**MAXIN**

**Explanation:** Specifies the maximum number of sequentially numbered received I-LPDUs that the link station using this SAP may have prior to sending an acknowledgment. The maximum valid value is 127. If the value is zero, the default of 1 is used.

---

**MAXOUT\_INCR**

**Explanation:** This dynamic window increment value is used to reduce bridge congestion. If the two end points of the link are on different rings, and the adapter detects an error condition requiring retransmission, the MAXOUT parameter will be set to 1. It will then be incremented by 1 each time MAXOUT\_INCR frames are acknowledged by the remote station, until it reaches the application program-requested value. For more details, see the *IBM Token-Ring Network Architecture Reference*. If the value is zero, the default of 1 is used.

---

### MAX\_RETRY\_CNT

**Explanation:** Specifies the number of retries for an unacknowledged command LPDU, or in the case of an I-LPDU timeout, the number of times that the non-responding remote link station will be polled with an RR/RNR command LPDU. This count is used in conjunction with the Response Timer and should be great enough to ensure time for ring error detection and recovery. This parameter also prevents continual retransmission of the same I frame. The maximum valid value is 255. If the value is zero, the default of 8 is used.

---

### MAX\_MEMBERS

**Explanation:** The maximum number of individual SAPs that may be assigned membership in the group SAP if this SAP is to be a group SAP as well as an individual SAP. Membership in the group SAP is assigned as the member SAPs are opened. This parameter may not exceed the similar parameter provided with the DIR.OPEN.ADAPTER command or configuration parameters and will default to that value if this parameter is zero.

---

### MAX\_I\_FIELD

**Explanation:** This parameter is the maximum size for an I frame that can be received by this SAP's link station. It applies to the information field in received I frames for link stations, and will be ignored if STATION\_COUNT is zero. If the value is zero, the default of 600 is used.

---

### SAP\_VALUE

**Explanation:** This is the value of the SAP to be assigned. The value must not be zero and the low-order bit is ignored. (In this example, "x" is the low order bit: B'nnnnnnx'.) X'00' is invalid and X'FE' cannot be used as a group SAP.

This is the SSAP for transmitted messages and the DSAP for received messages. See Figure 1-5 on page 1-11 or Figure 1-6 on page 1-13.

---

### OPTIONS\_PRIORITY

**Explanation:** Various SAP options, each represented by a bit. The bit being on (value of B'1') indicates taking the option. The high-order bit is the leftmost bit, 7.

- Bits 7-5 are ring access priority. Access priority is not recognized on the PC Network.

The transmit access priority to be placed in the AC byte of all transmissions from the SAP. If the access priority is too high, the command will terminate with the CCB\_RETCODE set to X'08'. This value is typically B'000'.

- Bit 4 is reserved. It should be zero.

- Bit 3 is the XID handling option.

If this bit is zero, the XID command frames are handled for this SAP by the DLC function of the adapter.

If this bit is 1, the XID command frames for this SAP are passed to the application program.

- Bit 2 is the individual SAP bit.

If this bit is 1, the SAP is an individual SAP.

The STATION\_COUNT parameter must be zero if bit 2 is not 1.

- Bit 1 is the group SAP bit.

If this bit is 1, the SAP is a group SAP.

- Bit 0 is the member of a group SAP bit.

If this bit is 1, the SAP may be a member of a group SAP. See the GROUP\_COUNT and GROUP\_LIST parameters.

At least one of the bits 0, 1, and 2 must be on. Bit 0 can be on only if bit 2 is on.

### STATION\_COUNT

**Explanation:** The number of link stations to reserve. This parameter is to provide link station resources so that subsequent DLC.OPEN.STATION commands may be issued.

If the requested number of stations is not available, the command will terminate with a CCB\_RETCODE of X'46'.

If the value is zero, no station may be opened for the SAP.

### GROUP\_COUNT

**Explanation:** The number of group SAPs defined in the GROUP\_LIST or GROUP\_LIST\_OFFSET field. If additional memberships are needed, use the DLC.MODIFY command. Valid values for the Token-Ring Network are from 0 to 8. Valid values for the PC Network are from 0 to 255.

### GROUP\_LIST or GROUP\_LIST\_OFFSET

**Explanation:** This field points to a list of group SAP values. The GROUP\_COUNT parameter indicates the number of valid values in this field. This field is ignored if the GROUP\_COUNT parameter is zero.

**For CCB1:** The GROUP\_LIST is the address of the list.

**For CCB2 and CCB3:** The GROUP\_LIST\_OFFSET is an offset to the list.

### DLC\_STATUS\_EXIT

**Explanation:** This field points to the beginning of an appendage routine provided by the application program. This routine will receive control whenever DLC status changes for this SAP. See "DLC Status Codes" on page B-25.

### DLC\_STATUS\_FLAG

**Explanation:** This flag must be set to a non-zero value if notification is wanted when any DLC status changes for the specified SAP and associated link stations. See "DLC Status Codes" on page B-25.

### DLC\_STATUS\_OFFSET

**Explanation:** This offset identifies the DLC status appendage so that the application program can be notified of any changes for the specified SAP and associated link stations. This appendage offset is passed to the application program device driver entry point in register DI when the Device Driver Interface call the application program for notification of any DLC status change. See "DLC Status Codes" on page B-25.

### DLC\_BUF\_SIZE

**Explanation:** The size of the buffers in the SAP buffer pool. This is the size of the entire buffer including all adapter overhead. The size must be a multiple of 16 with a minimum size of 80. If this value is zero, the default of 160 is used.



## DLC.OPEN.STATION

---

### DLC\_POOL\_LEN

**Explanation:** The number of 16-byte blocks in the SAP buffer pool.

If this value is zero, the default of 256 (4096 bytes) is used. If the DLC\_POOL\_ADDR is zero, this parameter is ignored.

---

### DLC\_POOL\_ADDR

**Explanation:** The starting address in PC system memory where the adapter is to build the SAP buffer pool. If this value is zero, the application program has the responsibility of building the buffer pool, and the DLC\_BUF\_SIZE parameter must still indicate the size of each buffer.

---

### ADAPTER\_STNS\_AVAIL

**Explanation:** If the command terminates with a return code of X'46', this field will contain the number of link stations that are currently available and have not been reserved. This SAP is not opened and no link stations are reserved for this SAP.

## DLC.OPEN.STATION

Hex 19

### DLC.OPEN.STATION

**Command Description:** This command allocates resources for a link station.

**Command Specifics:** The adapter support software performs functions to set up the link station in the adapter, but no ring communication takes place. A DLC.CONNECT.STATION command must be issued to either the local or remote link station by its application program to initiate connection-oriented communications. Thereafter, a DLC.CONNECT.STATION command must be issued at the other station to complete establishing the link.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	SAP_STATION_ID	2	DW	SAP station ID
2	LINK_STATION_ID	2	DW	Link station ID (X'nns') *
4	TIMER_T1	1	DB	T1 value (response timer)
5	TIMER_T2	1	DB	T2 value (acknowledgment timer)
6	TIMER_TI	1	DB	Ti value (inactivity timer)
7	MAXOUT	1	DB	Maximum transmits without a receive acknowledgment
8	MAXIN	1	DB	Maximum receives without a transmit acknowledgment
9	MAXOUT_INCR	1	DB	Dynamic window increment value
10	MAX_RETRY_CNT	1	DB	N2 value
11	RSAP_VALUE	1	DB	The remote SAP value
12	MAX_I_FIELD	2	DW	Maximum received information field

\* Indicates a returned value.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
14	ACCESS_PRIORITY	1	DB	Ring access priority. Access priority is not recognized on the PC Network.
15		1	DB	Reserved
<b>For CCB1:</b>				
16	DESTINATION	4	DD	Pointer to remote station address
<b>For CCB2 and CCB3:</b>				
16	DESTINATION_OFFSET	2	DW	Offset to remote station address
18		2	DW	Reserved for the application program
* Indicates a returned value.				

---

### SAP\_STATION\_ID

**Explanation:** The SAP ID value passed to the adapter.

---

### LINK\_STATION\_ID

**Explanation:** The link station ID value assigned by the adapter.

---

### TIMER\_T1

**Explanation:** Specifies the time period between 1 and 10 used to determine an inoperative condition on a link. If the value is zero, the default is as defined by the DLC.OPEN.SAP command. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by the configuration parameters.

---

### TIMER\_T2

**Explanation:** Specifies the time period between 1 and 10 used to delay transmission of an acknowledgment for a received I-LPDU for this SAP. If the value is zero, the default is as defined by the DLC.OPEN.SAP. If the value is greater than 10, the timer will not be used. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by the configuration parameters.

---

### TIMER\_Ti

**Explanation:** Specifies the time period between 1 and 10 used to determine an inactive condition on a link. If the value is zero, the default as defined by the DLC.OPEN.SAP command is used. See "Timers" on page 2-31. The time intervals are defined by the DIR.OPEN.ADAPTER command or by the configuration parameters.

---

### MAXOUT

**Explanation:** Specifies the maximum number of sequentially numbered transmitted I-LPDUs that the link station may have outstanding at any time. The maximum valid value is 127. If the value is zero, the default as defined by the DLC.OPEN.SAP command is used.

---

### MAXIN

**Explanation:** Specifies the maximum number of sequentially numbered received I-LPDUs that the link station may have prior to sending an acknowledgment. The maximum valid value is 127. If the value is zero, the default as defined by the DLC.OPEN.SAP command is used.

### MAXOUT\_INCR

**Explanation:** This dynamic window increment value is used to reduce bridge congestion. If the two end points of the link are on different rings, and an error condition requiring retransmission is detected, the MAXOUT counter will be set to one. It will then be incremented by one each time MAXOUT\_INCR frames are acknowledged by the remote station, until it reaches the value requested by the application program in the MAXOUT parameter. For more details see the *IBM Token-Ring Network Architecture Reference*. If the value is zero, the default as defined by the DLC.OPEN.SAP command is used.

---

### MAX\_RETRY\_CNT

**Explanation:** Specifies the number of retries for an unacknowledged command LPDU, or in the case of an I-LPDU timeout, the number of times that the non-responding remote link station will be polled with an RR/RNR command LPDU. The maximum valid value is 255. If the value is zero, the default as defined by the DLC.OPEN.SAP command is used.

---

### RSAP\_VALUE

**Explanation:** This is the value of the remote SAP partner. It must follow the same guidelines as the SAP\_VALUE parameter of the DLC.OPEN.SAP command, and it must be an individual SAP. A group SAP may not have a link station.

---

### MAX\_I\_FIELD

**Explanation:** This parameter applies to the information field in received I frames for this link station. If the value is zero, the default is as defined by the DLC.OPEN.SAP command.

---

### ACCESS\_PRIORITY

**Explanation:** The transmit access priority value to be placed in the AC byte of all transmissions from the link station. The format is B'nnn00000', where 'nnn' is the access priority value. If the access priority is higher than authorized for the adapter, the command will terminate with a CCB\_RETCODE of X'08'. Valid values for the access priority are from 0 to 3. Access priority is not recognized on the PC Network.

---

### DESTINATION

**Explanation:** This field points to a six-byte destination NODE\_ADDRESS of the remote adapter. The high-order bit of the NODE\_ADDRESS must be zero. Any routing information required for the frame header is supplied when the DLC.CONNECT.STATION command is issued.

The value must not be all ones. The two high-order (leftmost) bits must be B'01'. For other restrictions and details about addresses, see the *IBM Token-Ring Network Architecture Reference*.

---

### DESTINATION\_OFFSET

**Explanation:** This field is the offset to the destination six-byte NODE\_ADDRESS of the remote adapter. The high-order bit of the NODE\_ADDRESS must be zero. Any routing information required for the frame header is supplied when the DLC.CONNECT.STATION command is issued.

The value must not be all ones. The two high-order (leftmost) bits must be B'01'. For other restrictions and details about addresses, see the *IBM Token-Ring Network Architecture Reference*.

---

## DLC.REALLOCATE

Hex 17

### DLC.REALLOCATE

**Command Description:** This command adjusts (reallocates) the number of link stations allocated to a SAP without closing and re-opening the SAP.

**Command Specifics:** If this command is used to reduce the number of allocated link stations, then these de-allocated link stations are returned to the adapter's available pool.

If this command increases the number of allocated link stations, the link stations are assigned from the adapter's available pool.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	ID of affected SAP
2	OPTION_BYTE	1	DB	Increase or decrease indicator
3	STATION_COUNT	1	DB	Number of link stations to be reallocated
4	ADAPTER_AVAILABLE_STNS	1	DB	Number of link stations available for this adapter *
5	SAP_AVAILABLE_STNS	1	DB	Number of link stations available for this SAP *
<b>For CCB2 and CCB3:</b>				
6	ADAPTER_STATION_COUNT	1	DB	Number of link stations configured for this adapter *
7	SAP_STATION_COUNT	1	DB	Number of link stations reserved for this SAP *
* Indicates a returned value.				

#### STATION\_ID

**Explanation:** The SAP station ID affected (X'nn00').

#### OPTION\_BYTE

**Explanation:** The high order bit (bit 7) indicates whether the command increases or decreases the number of link stations. If bit 7 is 0, increase the number of link stations. If bit 7 is 1, decrease the number of link stations. Bits 6 through 0 are reserved.

#### STATION\_COUNT

**Explanation:** This parameter is the number of link stations to be reallocated as indicated by the OPTION\_BYTE. If this number is greater than the available link stations (either to increase or decrease), those that are available will be reallocated and the command will complete with a return code of X'57'. If this value is zero the command will complete and return the indicated values in the parameter table.

## DLC.RESET

---

### ADAPTER\_AVAILABLE\_STNS

**Explanation:** Number of available link stations (not allocated to a SAP) for this adapter (after this command completes). This value is only valid for the following return codes: X'00', X'40', X'57'.

---

### SAP\_AVAILABLE\_STNS

**Explanation:** Number of available link stations allocated to this SAP that are not currently in use (after this command completes). This value is only valid for the following return codes: X'00', X'57'.

---

### ADAPTER\_STATION\_COUNT

**Explanation:** Number of link stations configured for this adapter when it was initialized. This value is only valid for the following return codes: X'00', X'40', X'57'.

---

### SAP\_STATION\_COUNT

**Explanation:** Number of stations currently reserved for this SAP (after this command completes). This value is only valid for the following return codes: X'00', X'57'.

## DLC.RESET

Hex 14

DLC.RESET

**Command Description:** This command resets the following SAPs and link stations:

- One SAP and all associated link stations
- All SAPs and all associated link stations
- **For CCB2 and CCB3:** All SAPs and all associated link stations for one application program.

**Command Specifics:** This is an abortive situation. The affected station(s) will be closed. The command will not complete until all affected resources are freed. Transmit commands already queued for transmission will complete. If this command is issued while the ring is beaconing, it may not complete until the ring is no longer beaconing. See the *IBM Token-Ring Network Architecture Reference* for a description of ring beaconing.

All the outstanding commands will be terminated for the SAP(s) and station(s). All communications will cease and the SAP(s) and station(s) resources will be released. They must be reopened to be used further.

**For CCB1:** The CCB\_PARM\_TAB field contains the STATION\_ID value in the 2 high-order bytes. A STATION\_ID value of X'0000' defines all SAPs and all link stations. A STATION\_ID value of X'nn00' defines SAP 'nn' and all its link stations. The 2 remaining bytes are reserved.

When this command is completed, all outstanding commands that were terminated may be located using the CCB\_POINTER.

**For CCB2:** The CCB\_PARM\_OFFSET field contains the STATION\_ID value. A STATION\_ID value of X'0000' defines all SAPs and all link stations. A STATION\_ID value of X'nn00' defines SAP 'nn' and all its link stations.

This command may be issued by a system administrator using the System Key as defined by the configuration parameters. The System Key should be placed in the CCB\_PARAMETER\_2 field, if defined.

If an application program wishes to have pointers to the following returned, then the CCB\_CMPL\_FLAG of the DLC.RESET command must be set.

- Outstanding commands
- List of buffers remaining in the SAP buffer pools
- The outstanding receive frames

When a READ command is issued requesting command completions that post the completed DLC.RESET command, the pointers will be returned in the READ command's parameter table. When this command completes, all outstanding commands that were terminated will be queued on the CCB\_POINTER of the DLC.RESET command.

When the System Key is used by a system administrator with this command, only the SAP buffer pool buffers and outstanding receive frames for SAPs opened by the system administrator are returned in the READ command's parameter table. In addition, only outstanding commands issued by the system administrator will be returned to the system administrator with the CCB\_POINTER of the DLC.RESET command.

Application programs affected by a DLC.RESET issued with a System Key must have a READ command outstanding requesting notification of a system action to be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

**For CCB3:** The CCB\_PARM\_OFFSET field contains the STATION\_ID value. A STATION\_ID value of X'0000' defines all SAPs and all link stations. A STATION\_ID value of X'nn00' defines SAP 'nn' and all its link stations.

This command may be issued by a system administrator using the the System Key as defined by the configuration parameters. The System Key should be placed in the CCB\_PARAMETER\_2 field if defined.

If an application program wishes to have buffers from the SAP buffer pools returned, then the CCB\_CMPL\_APPNDG of the DLC.RESET command must be set. When this command completes, all outstanding commands that were terminated will be queued on the CCB\_POINTER of the DLC.RESET command.

When the System Key is used by a system administrator with this command, only the buffers from buffer pools of SAPs opened by the system administrator are returned when the application is called for command completion posting.

Application programs affected by a DLC.RESET issued with the System Key can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

This command may be issued by a system administrator using the System Key as defined by the configuration parameters. The System Key should be placed in the CCB\_PARAMETER\_2 field, if defined.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DLC.SET.THRESHOLD

## DLC.SET.TRESHOLD

Hex 33

### DLC.SET.THRESHOLD

#### Command Description:

#### This command is for CCB2 only:

This command sets a threshold for the SAP or direct station's buffer pool. Whenever the number of buffers in a SAP buffer pool falls below the specified threshold, the application program using the adapter support software will be notified.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
For CCB2 only:				
0	STATION_ID	2	DW	SAP station ID
2	BUFFER_THRESHOLD	2	DW	SAP buffer pool threshold number
4	ALERT_SEMAPHORE	4	DD	Alert semaphore

---

#### STATION\_ID

**Explanation:** The station ID of the SAP or direct station's buffer pool.

Use the SAP station ID to identify the buffer pool for which the threshold is being set.

---

#### BUFFER\_THRESHOLD

**Explanation:** Number of buffers that will define the threshold. If the number of buffers in the SAP buffer pool is less than this threshold, the ALERT\_SEMAPHORE will be cleared to notify the application program using the buffers. If a buffer is removed from the pool and the pool contains less buffers than the threshold, the ALERT\_SEMAPHORE will be cleared.

---

#### ALERT\_SEMAPHORE

**Explanation:** A system semaphore can be used to alert the application program of the threshold exceeded condition. When the condition exists, the adapter support software will clear the ALERT\_SEMAPHORE to notify the application program.

To specify a system semaphore, the application program places the semaphore handle into the ALERT\_SEMAPHORE field. The semaphore handle is returned from Operating System/2 when the system semaphore is created or opened.

## DLC.STATISTICS

Hex 1E

### DLC.STATISTICS

**Command Description:** This command reads and optionally resets the DLC logs specified. The log is transferred to the buffer indicated by the parameter table.

**Command Specifics:** For "Adapter" and "DIRECT INTERFACE" logs, see the DIR.READ.LOG command.

If the STATION.ID indicates a SAP and no link station data, this command executes totally in the PC system. The return code is available to the application program upon return from the adapter support software.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	SAP_STATION_ID	2	DW	Identify the log to read
2	LOG_BUF_LENGTH	2	DW	Size of buffer at LOG_BUFFER_ADDR and LOG_BUFFER_OFFSET
<b>For CCB1:</b>				
4	LOG_BUFFER_ADDR	4	DD	Address to place the log data.
<b>For CCB2 and CCB3:</b>				
4	LOG_BUFFER_OFFSET	2	DW	Offset of buffer for log data
6		2	DW	Reserved
<b>For all CCBs:</b>				
8	LOG_ACT_LENGTH	2	DW	Actual length of log *
10	OPTIONS	1	DB	Command options
* Indicates a returned value.				

---

#### STATION\_ID

**Explanation:** The station ID of the SAP or link station that the statistics are read from or optionally reset.

---

#### LOG\_BUF\_LENGTH

**Explanation:** The length of the buffer defined by LOG\_BUFFER\_ADDR or LOG\_BUFFER\_OFFSET.

---

#### LOG\_BUFFER\_ADDR

**Explanation:** The address of the buffer where the log data is to be placed.

---

#### LOG\_BUFFER\_OFFSET

**Explanation:** The offset of the buffer where the log data is to be placed.



## DLC.STATISTICS

---

### LOG\_ACT\_LENGTH

**Explanation:** The returned value of the actual length of the log that was requested. If this value is greater than the value of LOG\_BUF\_LENGTH, not all of the log has been transferred. CCB\_RETCODE will contain X'15'.

**Note:** On a "Lost Data" condition, the log will be cleared if the options indicate reset.

---

### OPTIONS

**Explanation:** Command options

- Bit 7 : If this bit is on, the counters will be reset to zero where appropriate.
- Bits 0 - 6 : Reserved. Should be zero, but not checked.

### Log Formats

**SAP log** (maintained in PC system memory)

If the DLC.STATISTICS command request is for a SAP log (X'nn00'), or a DIRECT station log (X'00ss') for CCB1, the format of the data placed in the buffer is shown in the following table. The counters are reset after they are read.

Bytes	8086 Type	Meaning
0-3	DD	Number of frames transmitted
4-7	DD	Number of frames received
8-11	DD	Number of frames discarded (no receive command)
12-15	DD	Number of times data was lost
16-17	DW	Number of buffers available in the SAP buffer pool

**LINK STATION log** (maintained in the adapter)

If the DLC.STATISTICS command request is for a link station log (X'nnss'), the format of the data placed in the buffer is:

Bytes	8086 Type	Meaning
0-1	DW	Number of I frames transmitted
2-3	DW	Number of I frames received
4	DB	Number of I frame receive errors
5	DB	Number of I frame transmission errors
6-7	DW	Number of times T1 expired (other than in data-transfer mode)
8	DB	Last command/response received
9	DB	Last command/response sent
10	DB	Link Primary State
11	DB	Link Secondary State
12	DB	Send state variable
13	DB	Receive state variable
14	DB	Last received NR
15	DB	Length of network header used in station transmissions
16-47	DB	Network header used in station transmissions

**Notes:**

1. See "Link Station States" on page 2-29 and 7-55 for more about bytes 10 and 11.
2. For the SAP counters that are 4 bytes long, an overflow condition is not reported. An overflow indication is given when the counters reach half their maximum value (X'80' or X'8000') by a DLC status change notification.
3. Some parameters are not counters, and these parameters are not reset when the reset option is selected.

**DIR.CLOSE.ADAPTER**

Hex 04

**DIR.CLOSE.ADAPTER**

**Command Description:** This command closes the adapter and terminates all network communications or terminates the "open wrap test."

**Command Specifics:** The command will force an immediate shutdown of network communications, and all outstanding commands will have the control block field CCB\_RETCODE set with X'0B'.

**For CCB1:** If the adapter was opened with a lock code, this command must have the same hexadecimal value in the first 2 bytes of the CCB\_PARM\_TAB field in order to close the adapter. If the key code is not provided or is not correct, the DIR.CLOSE.ADAPTER command will be rejected with a CCB\_RETCODE of X'05' (required parameters not provided).

**For CCB2:** For this interface the close works on a per application program basis. The adapter does not close physically.

The CCB\_POINTER field will be set with the address of a queue of CCBs that have been terminated by this command.

If the application program has any SAP stations or link stations open, they will be reset (closed) prior to the command completing.

Operating System/2 is a multi-tasking operating system, therefore multiple application programs can interface with the adapter support software. This command may be issued by a system administrator using the System Key as defined by the configuration parameters. A physical close resulting from the DIR.CLOSE.ADAPTER command being issued will only occur if this command is issued by the system administrator with the System Key.

Application programs affected by a DIR.CLOSE.ADAPTER issued with the System Key can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information. The System Key should be placed in the CCB\_PARAMETER\_2 field, if defined.

Whenever a physical close occurs resulting from the use of the System Key, all outstanding commands for all application programs will be set with the CCB\_RETCODE of X'62'. Commands on a per application program basis will be chained to the first command cancelled for each application program.

If the application program has previously set bits of the functional address with the DIR.SET.FUNCTIONAL.ADDRESS command, the adapter support software will reset the bits. Bits defined by the configuration parameters, however, will not be affected.

If the CCB\_CMPL\_FLAG is non-zero, an attempt will be made to update a READ command's parameter table with pointers to outstanding CCBs, freed SAP buffers, and outstanding receive frames. If there is a READ command chained to the DIR.CLOSE.ADAPTER command (CCB\_READ\_FLAG is non-zero and the CCB\_POINTER contains the address of a READ CCB),

## DIR.CLOSE.ADAPTER

that READ command will be used to return the outstanding data areas of the application program. If there is no READ command chained to the DIR.CLOSE.ADAPTER command, but a READ command is outstanding specifying notifications of Command Completions, it will be used to return outstanding data areas of the application program. If no READ is outstanding specifying notifications of Command Completions, free SAP buffers, and outstanding receive frames will not be posted to the user but the outstanding CCBs will be chained to the CCB\_POINTER field of this command.

**For CCB3:** For this interface the close works on a per application program basis. The adapter does not close physically.

The CCB\_POINTER field will be set with the address of a queue of CCBs that have been terminated by this command.

If the application program has any SAP stations or link stations open, they will be reset (closed) prior to the command completing.

Operating System/2 is a multi-tasking operating system, therefore multiple application programs can interface with the adapter support software. This command may be issued by a system administrator using the System Key as defined by the configuration parameters. A physical close resulting from the DIR.CLOSE.ADAPTER command being issued will only occur if this command is issued by the system administrator with the System Key.

Application programs affected by a DIR.CLOSE.ADAPTER issued with the System Key can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information. The System Key should be placed in the CCB\_PARAMETER\_2 field if defined.

Whenever a physical close occurs resulting from the use of the System Key, all outstanding commands for all application programs will be set with the CCB\_RETCODE of X'62'. Commands on a per application program basis will be chained to the first command cancelled for each application program.

If the application program has previously set bits of the functional address with the DIR.SET.FUNCTIONAL.ADDRESS command, the adapter support software will reset the bits. Bits defined by the configuration parameters, however, will not be affected.

The cancelled commands and freed SAP buffers will be returned to the application program device driver entry point for the system action event if the SYSTEM\_ACTION\_APPNDG is defined.

An application program using the Device Driver Interface must issue the DIR.CLOSE.ADAPTER command when it no longer requires the services provided by the adapter support software, since the adapter support software cannot determine when an application program using the Device Driver Interface terminates. If the DIR.CLOSE.ADAPTER command is not issued, the internal control blocks used by the adapter support software to support an application program will not be available for other application programs.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

**DIR.CLOSE.DIRECT**

Hex 34

**DIR.CLOSE.DIRECT****Command Description:****This command is for CCB2 and CCB3 only:**

This command releases ownership of all direct stations.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software. This command may be issued by a system administrator with the System Key as defined by the configuration parameters. The System Key code can be used and is located in the CCB\_PARAMETER\_2 field. This key code is used to force the availability of the DIRECT stations. If the System Key is used all outstanding commands and buffers from the DIRECT buffer pool can be returned to the application program. See "System Action Exceptions for OS/2" on page B-58 for more information. This function is not typically used by an application program.

**For CCB2:** If a READ command is used to post this command, the buffers from the DIRECT buffer pool will be returned in the READ command parameter table. If the System Key is used, all outstanding commands and buffers from the DIRECT buffer pool will be returned to the application program affected when a READ command issued by the application program completes for a system action event.

**For CCB3:** This command completion can be posted to the completion appendage specified in the CCB of the command. When the completion appendage is defined, all outstanding commands and the direct buffer pool will be returned in the information table when this command completes. If the System Key is used, all outstanding commands and the direct buffer pool will be returned to the application program if a SYSTEM\_ACTION\_APPNDG is defined.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

**DIR.DEFINE.MIF.ENVIRONMENT**

Hex 2B

**DIR.DEFINE.MIF.ENVIRONMENT****Command Description:****This command is for CCB1 only:**

This command defines the environment required for a NETBIOS emulation program to operate with the adapter support software.

**Command Specifics:** This command informs the adapter support software of the interactive routines to be provided by the NETBIOS emulation program. The adapter number in the CCB must be a value from X'00' to X'03', but the environment will be defined for all Token-Ring Network and all PC Network adapters if they are installed in the PC system. This command will not have any effect on the original PC Network Adapter if one is installed and the Local Area Network Support Program is not being used. The command may be issued when an adapter is either open or closed. This command is executed entirely in the adapter support software in the PC system. Therefore, the command completion appendage is not required, as the command is complete upon return. However, the command completion appendage will be taken if provided.

## DIR.DEFINE.MIF.ENVIRONMENT

**Note:** A NETBIOS emulation program must at some time post a completion code to the NCB presented to it by an application program. If no command completion appendage (NCB\_POST@) has been provided, the emulation program should end with an IRET instruction to return to the adapter support software which will return to the application program.

If an appendage has been defined, the emulation program should end with the following instruction sequence to cause the adapter support software to call the appendage.

CLI

---

LES BX,NCBADDR      ES and BX Point to the NCB

---

STC                      Set carry flag to indicate POST

---

RET FAR 2              Return (around flags on stack)

This special handling of the flags is the indication to the adapter support software that the appendage (NCB\_POST@) is to be called. That appendage should end with an IRET instruction.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
For CCB1 only:				
0	NCB_INPUT@	4	DD	The address of NCB module
4	NCB_OPEN@	4	DD	The address of the open module
8	NCB_CLOSE@	4	DD	The address of the close module
12	NCB_ENABLE@	4	DD	The address of the interrupt module *
* Indicates a returned value.				

---

### NCB\_INPUT

**Explanation:** This field must have a value other than zero. It must contain the address of a module or routine that the adapter support software can jump to when it has determined that the control block is an NCB rather than a CCB. Registers ES and BX will point to the NCB. Register AL contains flags as defined in the PDT.TRACE CCB entry byte 1. See page 3-61.

The specified module or routine must end with an IRET instruction back to the application program that issued the NCB. It does not return to the adapter support software.

The module will be entered with the same stack used by the application program that issued the NCB. It is the responsibility of the module to return the stack and registers as they were when the module was entered. Only the return address and flags are on the stack when entered.

---

### NCB\_OPEN

**Explanation:** This field must have a value other than zero. It must contain the address of a module or routine that the adapter support software can call when it has opened an adapter. It does this to inform the NETBIOS emulator that the adapter is open. Registers ES and BX will point to the CCB used to open the adapter. Register CX contains the adapter number.

The specified module or routine must end with a Far Return instruction back to the adapter support software, with register AL set to indicate the return code. If the AL register is set to zero, the NETBIOS emulator indicates a good return. If AL is not zero, the DIR.OPEN command will be completed with a return code of X'10'.

---

**NCB\_CLOSE**

**Explanation:** This field must have a value other than zero. It must contain the address of a module or routine that the adapter support software can call when it has closed an adapter for any reason. Register CX contains the adapter number.

The specified module or routine must end with a Far Return instruction back to the adapter support software.

---

**NCB\_ENABLE**

**Explanation:** The adapter support software will return the address of a routine that is to be called when interrupts are to be enabled.

**DIR.INITIALIZE**

Hex 20

DIR.INITIALIZE

**Command Description:** This command initializes the adapter support software areas in the PC system, resets all adapter tables and buffers, and directs the adapter to run the bring-up tests. Bring-up tests are for the Token-Ring Network only. The adapter's programmable timer is started and set to interrupt the PC system at 100-millisecond intervals.

For purposes of system integrity, the DIR.INITIALIZE command loops with interrupts enabled (except in the hardware interrupt appendage for CCB1 as explained below) until one of the following occurs.

1. The adapter interrupts, indicating completion.
2. Time expires (approximately 12 seconds for a Token-Ring Network adapter).

The 12 seconds that may elapse is due to the implementation of a three second time out and retry function. The initialization attempt is not aborted until at least four attempts have been made to initialize the adapter and execute diagnostics.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**For CCB1:** Because the command loops until complete, no command completion appendage needs to be defined. The CCB\_RETCODE is available upon return.

The command may be issued at any time. It resets the adapter and any CCBs in process will not be returned. No attempt is made to recover outstanding CCBs.

This command must be issued before any other command may be issued. This command will also preempt any other command queued in the adapter support software, it will execute immediately.

**For CCB2 and CCB3:** This command may be issued at any time by a system administrator with the System Key as defined by the configuration parameters. The System Key should be placed in the CCB\_PARAMETER\_2 field if defined.

When issued, DIR.INITIALIZE resets the adapter. Also, outstanding CCBs and buffer resources will be returned to the application program that issued them as system action exception information. That is, the returned commands CCB\_RETCODE will be set to X'62'.

Application programs affected by a DIR.INITIALIZE can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

## DIR.INITIALIZE

Valid Return Codes: see "CCB Return Codes Listed by Command" on page B-4.

CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	BRING_UPS	2	DW	Bring-up error code. * This parameter is not recognized on the PC Network.
<b>For CCB1:</b>				
2	SRAM_ADDRESS	2	DW	Segment value of adapter shared RAM. * This parameter is not recognized on the PC Network.
4		4	--	Reserved
8	ADPTR_CHK_EXIT	4	DD	I/O appendage exit, adapter check
12	NETW_STATUS_EXIT	4	DD	I/O appendage exit, network status change
16	PC_ERROR_EXIT	4	DD	I/O appendage exit, error in the PC system. This parameter is not recognized on the PC Network.
<b>For CCB2 and CCB3:</b>				
2	SRAM_ADDRESS	2	DW	Configured address of shared RAM. * This parameter is not recognized on the PC Network.
4		1	DB	Reserved *
5		15	DB	Reserved
<b>For CCB2:</b>				
20	VIRTUAL_SRAM_ADDRESS	4	DD	Virtual address of shared RAM * For the PC Network, this is the address of RAM containing link stations, SAPs, etc.
24	VIRTUAL_MMIO_ADDRESS	4	DD	Virtual address of MMIO * This is not set on the PC Network.
28	DATA_SEG_ADDRESS	4	DD	Virtual address of device driver's data segment *
<b>For CCB3:</b>				
20	VIRTUAL_SRAM_ADDRESS	4	DD	Virtual address with GDT selector of shared RAM * For the PC Network, this is the address of RAM containing link stations, SAPs, etc.
24	VIRTUAL_MMIO_ADDRESS	4	DD	Virtual address with GDT selector of MMIO * This is not set on the PC Network.
28	DATA_SEG_ADDRESS	4	DD	Virtual address with GDT selector of device drivers data segment *
* Indicates a returned value.				

### BRING\_UPS

**Explanation:** Indicates the results of the adapter bring-up testing. If the value is not X'0000', the command will terminate with a CCB\_RETCODE of X'07'. See "Bring-Up Errors for All CCBs" on page B-48 for a list of bring-up error codes and descriptions. This parameter is not recognized on the PC Network.

---

**SRAM\_ADDRESS**

**Explanation:** This defines the PC system memory segment where the adapter shared RAM is to be addressed.

This field is not recognized on the PC Network.

**For CCB1:**

- For Token-Ring Network adapters with PC I/O Bus

If the application program specifies a non-zero number for shared RAM when loading the adapter support software, that value will be used as the segment address for the shared RAM.

If zero is coded here, the defaults (X'D800' for adapter 0, X'D400' for adapter 1) are used.

The value must be on the same address boundary as the shared RAM size mapped into the PC system (for example, on an 8K boundary for 8K bytes of shared RAM, on a 16K boundary for 16K bytes of shared RAM, and so on).

- For Token-Ring Network adapters with Micro Channel

The input value is ignored (the shared RAM segment address is set using the Reference Diskette shipped with the PC system). A non-zero value coded here will not update the shared RAM address and no error code is returned to the application program.

The value returned will be the segment address used for the shared RAM on both Token-Ring Network adapters with PC I/O Bus and Token-Ring Network adapters with Micro Channel.

**For CCB2 and CCB3:**

- For Token-Ring Network adapters with PC I/O Bus

The shared RAM segment address is defined using the configuration parameters. See Appendix E, "Operating System/2 Extended Edition 1.1 Information" for additional information.

- For Token-Ring Network adapters with Micro Channel

The input value is ignored. The shared RAM segment address is set using the Reference Diskette shipped with the PC system

The value returned will be the segment address used for the shared RAM on both Token-Ring Network adapters with PC I/O Bus and Token-Ring Network adapters with Micro Channel. This value may not be used directly by the system administrator while the PC system is running in protect mode.

---

**ADPTR\_CHK\_EXIT**

**Explanation:** The address of a user-provided appendage routine that will be taken when an adapter error condition is detected. If the value is zero, no exit is defined. This exit may also be overridden by the DIR.OPEN.ADAPTER, DIR.SET.USER.APPENDAGE, and DIR.MODIFY.OPEN.PARMS commands. See "Exception Indications" on page B-40 for adapter check error codes.

---

**NETW\_STATUS\_EXIT**

**Explanation:** The address of a user-provided appendage routine that will be taken when the network status changes. This exit may also be overridden by the DIR.OPEN.ADAPTER, DIR.SET.USER.APPENDAGE, and DIR.MODIFY.OPEN.PARMS commands. See "Exception Indications" on page B-40 for network status codes.



### PC\_ERROR\_EXIT

**Explanation:** The address of a user-provided appendage routine that will be taken when the adapter support software detects an error condition in the PC system. This is not used on the PC Network. This exit may also be overridden by the DIR.OPEN.ADAPTER, DIR.SET.USER.APPENDAGE, and DIR.MODIFY.OPEN.PARMS commands. See "Exception Indications" on page B-40 for PC system errors.

---

### VIRTUAL\_SRAM\_ADDRESS

**Explanation:**

**For CCB2:** Virtual address of adapter shared RAM.

Defines the address (selector:offset) that can be used to access the Token-Ring Network adapter's shared RAM.

Defines the address (selector:offset) that can be used to access the PC Network allocated PC system RAM.

An entry for this memory segment is made into the LDT of the Operating System/2 Extended Edition 1.1 process that issues this command. This command is only valid for the process that issued it.

**For CCB3:** Virtual address with GDT Selector of adapter shared RAM.

Defines the address with GDT selector (selector:offset) that can be used to access the Token-Ring Network adapter's shared RAM.

Defines the address with GDT selector (selector:offset) that can be used to access the PC Network allocated PC system RAM.

---

### MMIO\_ADDRESS

**Explanation:**

**For CCB2:** Virtual address of adapter MMIO.

Defines the address (selector:offset) that can be used to access the adapter MMIO.

This parameter is NOT returned when using PC Network adapters.

An entry for this memory segment is made into the LDT of the Operating System/2 Extended Edition 1.1 process that issues this command. This command is only valid for the process that issued it.

**For CCB3:** Virtual address with GDT Selector of adapter MMIO.

Defines the address with GDT Selector (selector:offset) that can be used to access the adapter MMIO.

This parameter is NOT returned when using PC Network adapters.

---

### DATA\_SEG\_ADDRESS

**Explanation:**

**For CCB2:** Virtual address of the adapter support software's data segment.

Defines the address (selector:offset) that can be used to access the adapter support software's data segment.

This memory segment has been mapped in the GDT of the Operating System/2 Extended Edition 1.1 process that issues this command.

An entry for this memory segment is made into the LDT of the Operating System/2 Extended Edition 1.1 process that issues this command. This command is only valid for the process that issued it.

**For CCB3:** Virtual address with GDT Selector of adapter support software's data segment.

Defines the address with GDT Selector (selector:offset) that can be used to access the adapter support software's data segment.

## DIR.INTERRUPT

Hex 00

DIR.INTERRUPT

**Command Description:** This command is available to force an adapter interrupt. It performs no operation.

**For CCB1:** The adapter must have been initialized, but does not have to be opened for this command to be accepted.

**For CCB2 and CCB3:** The adapter must be opened before this command is issued.

**Command Specifics:** No parameter table is required.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.MODIFY.OPEN.PARMS

Hex 01

DIR.MODIFY.OPEN.PARMS

**Command Description:**

**This command is for CCB1 only:**

This command is used to modify certain values set by the DIR.OPEN.ADAPTER command.

**Command Specifics:** This command will be rejected if either a BUFFER.FREE command has been issued, or a RECEIVE command is active at the direct interface, or if a direct interface buffer pool has been defined.

After this command has been issued successfully, it may not be issued again until a DIR.RESTORE.OPEN.PARMS command has been issued and successfully completed.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.OPEN.ADAPTER

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
For CCB1 only:				
0	DIR_BUF_SIZE	2	DW	The size of the new direct interface SAP buffers
2	DIR_POOL_BLOCKS	2	DW	The length in 16-byte blocks, of buffers in the new direct interface buffer pool
4	DIR_POOL_ADDRESS	4	DD	The starting segment of the new direct interface buffer pool
8	ADPT_CHK_EXIT	4	DD	New I/O appendage exit, adapter check
12	NETW_STATUS_EXIT	4	DD	New I/O appendage exit, network status
16	PC_ERROR_EXIT	4	DD	New I/O appendage exit, error in PC system
20	OPEN_OPTIONS	2	DW	New options (wrap option is ignored)

See the same parameters under the DIR.OPEN.ADAPTER command description following.

## DIR.OPEN.ADAPTER

Hex 03

DIR.OPEN.ADAPTER

### Command Description:

**For CCB1:** This command makes the adapter ready for normal network communication. All buffers and tables will be re-initialized.

**For CCB2 and CCB3:** This command makes the adapter support software ready for normal network communications on all network adapters for a given application program. In addition, this command provides the application program with DLC and DIRECT interface information necessary for network communications.

The DIR.OPEN.ADAPTER command must be issued and complete successfully before any network communication can start. The only commands that can be issued before opening the adapter are those commands that use the System Key as defined by the configuration parameters or the DIR.STATUS command.

Since Operating System/2 Extended Edition 1.1 is a multi-tasking operating system, where multiple application programs can interface with the adapter support software, each DIR.OPEN.ADAPTER command will result in a logical open. This command is executed entirely in the PC system when a logical open is performed. The return code is available to the application program upon return from the adapter support software. A physical open will only occur if the adapter has been physically closed due to a system action or an unrecoverable error. When a physical open does occur all buffers and tables in the adapter will be re-initialized.

### Command Specifics:

**For CCB1:** This command may not be issued unless the adapter is in a "closed" state, such as that following a DIR.INITIALIZE or a DIR.CLOSE.ADAPTER command. The only commands that should be issued before the DIR.OPEN.ADAPTER command are DIR.INITIALIZE, DIR.INTERRUPT, and DIR.STATUS.

This command also makes the adapter ready for an adapter wrap test on Token-Ring Network adapters.

If the first DIR.OPEN.ADAPTER is in process when the second DIR.OPEN.ADAPTER is issued the value for NODE\_ADDRESS may be invalid.

If the OPEN\_OPTIONS bit 9 (Pass Parm Table bit) is on when the DIR.OPEN.ADAPTER command is issued then all the ADAPTER\_PARMS and DLC\_PARMS values in the parameter table that were set to use default values will be updated with the default value actually used. If the adapter is open when this command is issued and the open option bit 9 is on, the command will complete with the X'03' return code and also have its parameter tables updated with the currently active open parameter values. The actual value for the ADAPTER\_PARMS and DLC\_PARMS fields are returned. The actual value for the OPEN\_LOCK and PRODUCT\_ID\_ADDRESS fields are not returned, a value of zero is stored in these fields. If the DLC interface is not open then zeros will be returned in the DLC\_PARMS field.

The DIR.OPEN.ADAPTER parameter table has four pointers that point to function-oriented tables. These tables contain open parameters for the adapter itself, the DIRECT interface, the DLC interface, and NETBIOS. See Chapter 4, "NETBIOS" on page 4-1 for using NETBIOS.

#### For CCB2:

- This command may not be issued unless the adapter is in a logically closed state for the application program issuing the command. A logically closed state exists after one of the following occurs: adapter initialization, a fatal error detection, or the execution of a DIR.CLOSE.ADAPTER command.
  - After a DIR.OPEN.ADAPTER command is issued, no commands (other than commands using the System Key or the DIR.STATUS command) are allowed from the application program until the open completes.
  - Posting of the DIR.OPEN.ADAPTER command can only be done using a system semaphore or polling of the return code.
  - The adapter must be logically closed at the time the DIR.OPEN.ADAPTER command is issued.
  - The CCB\_SEMAPHORE is the posting semaphore.
  - The CCB\_APPL\_ID is the application program ID.
 

A unique ID is returned for each application program. The adapter support software uses this ID to reference application program resources. The application program ID returned must be used for all following commands that the application program issues.
  - The CCB\_APPL\_KEY is a key code.
 

The key code is used for the application program's resource security.

This is a key code to logically lock the resources of application programs. No resource of an application program can be used, discarded or modified without this key code. If an application program wishes not to request that its resources be controlled then X'0000' should be specified. If a key code is specified then all following commands must also specify a key that matches this CCB\_APPL\_KEY code.
- The DIR.OPEN.ADAPTER command's parameter table contains two pointers that point to function-oriented tables. The fields of these two tables will be returned by the adapter support software and will contain open parameters for the adapter itself and for the DLC interface.
- The Operating System/2 Extended Edition 1.1 process that actually issues the DIR.OPEN.ADAPTER command is considered the "application program." When this process is terminated all application program control blocks used to maintain the logical relationship between the application program and the adapter support software are cleaned up and made available for other application programs.

## DIR.OPEN.ADAPTER

- After the DIR.OPEN.ADAPTER command has been issued, the application program should issue the DIR.STATUS command to retrieve the current network status to ensure that the network is in working order.

### For CCB3:

- This command may not be issued unless the adapter is in a logically closed state for the application program issuing the command. A logically closed state exists after one of the following occurs: adapter initialization, a fatal error detection, or the execution of a DIR.CLOSE.ADAPTER command.
  - After a DIR.OPEN.ADAPTER command is issued, no commands (other than commands using the System Key) are allowed from the application program until the open completes.
  - Posting of the DIR.OPEN.ADAPTER command can only be done using an appendage or polling of the return code.
  - The adapter must be logically closed at the time the DIR.OPEN.ADAPTER command is issued.
  - The CCB\_APPL\_ID is the application program ID.

A unique ID is returned for each application program. The adapter support software uses this ID to reference application program resources. The application program ID returned must be used for all following commands that the application program issues.

- The CCB\_APPL\_KEY is a key code.

The key code is used for the application program's resource security.

This is a key code to logically lock the resources of application programs. No resource of an application program can be used, discarded or modified without this key code. If an application program wishes not to request that its resources be controlled then X'0000' should be specified. If a key code is specified then all following commands must also specify a key that matches this CCB\_APPL\_KEY code.

- The DIR.OPEN.ADAPTER command's parameter table contains two pointers that point to function-oriented tables. The fields of these two tables will be returned by the adapter support software and will contain open parameters for the adapter itself and for the DLC interface.
- After the DIR.OPEN.ADAPTER command has been issued, the application program should issue the DIR.STATUS command to retrieve the current network status to ensure that the network is in working order.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

## CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>For CCB1:</b>				
0	ADAPTER_PARMS	4	DD	Pointer to the adapter parameters
4	DIRECT_PARMS	4	DD	Pointer, direct interface parameters
8	DLC_PARMS	4	DD	Pointer to the DLC parameters
12	NCB_PARMS	4	DD	Pointer to the NETBIOS parameters
<b>For CCB2 and CCB3:</b>				
0	ADAPTER_PARMS_OFFSET	2	DW	Offset to the adapter parameters
2		2	DW	Reserved for the application program
4		4	DD	Reserved
8	DLC_PARMS_OFFSET	2	DW	Offset to the DLC parameters
10		2	DW	Reserved for the application program
12		4	DD	Reserved

**ADAPTER\_PARMS and ADAPTER\_PARMS\_OFFSET**

**Explanation:** See Table 3-1 on page 3-38.

**For CCB1:** This parameter is required and points to a parameter table. There must be a non-zero value provided for this parameter.

**For CCB2 and CCB3:** This parameter is an offset that points to a parameter table. This pointer should be provided so that global DLC parameters can be returned to the application program.  
Most parameters are returned values that have been defined by configuration parameters.

**DIRECT\_PARMS**

**Explanation:** This parameter is required and points to a parameter table. There must be a non-zero value provided for this parameter. See the table on 3-43.

**DLC\_PARMS and DLC\_PARMS\_OFFSET**

**Explanation:** See the table on 3-44.

**For CCB1:** This parameter is required and points to a parameter table. It must be defined (not zero) if any interface other than the direct interface is to be used. If this field value is zero, the DLC interface and NETBIOS are not operational.

**For CCB2 and CCB3:** This parameter is an offset that points to a parameter table. This pointer should be provided so that global adapter parameters can be returned to the application program.

Most parameters are returned values that have been defined by configuration parameters.

**NCB\_PARMS**

**Explanation:** This field is a pointer to a parameter table. If this field value is zero, when NETBIOS is used, all default values will be used. See page 4-7 for the NETBIOS Open Parameters table.

## DIR.OPEN.ADAPTER

Table 3-1. Adapter Parms Open Parameters				
OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	OPEN_ERROR_CODE	2	DW	Open adapter errors detected. Set by the adapter. **
2	OPEN_OPTIONS	2	DW	Various options *
4	NODE_ADDRESS	6	DB	This adapter's address **
10	GROUP_ADDRESS	4	DB	Set group address *
14	FUNCTIONAL_ADDR	4	DB	Set functional address *
18	NUMBER_RCV_BUFFERS	2	DW	Number of receive buffers *
20	RCV_BUFFER_LEN	2	DW	Length of the receive buffers *
22	DHB_BUFFER_LENGTH	2	DW	Length of the transmit buffers *
24	DATA_HOLD_BUFFERS	1	DB	Number of transmit buffers. *  This parameter is not recognized on the PC Network. Reserved for the Token-Ring Network.
<b>For CCB1:</b>				
25		1	DB	Reserved
26	OPEN_LOCK	2	DW	A protection code to control closing the adapter
28	PRODUCT_ID_ADDRESS	4	DD	Address of the 18-byte product ID
<b>For CCB2 and CCB3:</b>				
25		3	DB	Reserved
28	PRODUCT_ID_OFFSET	2	DW	Offset of 18-byte product ID
30		2	DW	Reserved for the application program
32	BRING_UPS	2	DW	Bring up errors *
34	INIT_WARNINGS	2	DW	Initialization warnings *
<b>For CCB2:</b>				
36	SEMAPHORE_COUNT	2	DW	Count of system semaphore handles
38	SYS_SEMAPHORE_TABLE	4	DD	Address to system semaphore table
<b>For CCB3:</b>				
36		2	DW	Reserved for the application program
38		4	DD	Reserved for the application program
42	DDNAME	8	DB	Device driver name
* Indicates a returned value for CCB2 and CCB3.				
** Indicates a returned value for all CCBs.				

### OPEN\_ERROR\_CODE

**Explanation:** Indicates the results of the open adapter testing. If the value is not X'0000', the command will terminate with a CCB\_RETCODE of X'07'. See "Token-Ring Network Adapter Open Errors for All CCBs" in Appendix B, "Return Codes" for open error codes.

### OPEN\_OPTIONS

**Explanation:** Various options, each defined by a bit. A bit on (1) indicates that the option is to be taken. Bit 15 is the high-order (leftmost) bit. These bits are not recognized on the PC Network.

**For CCB2 and CCB3:** These are returned values.

**Bit 15 - Wrap Interface** The adapter will not attach itself to the network. Instead it will cause all user transmit data to be wrapped as received data.

**Bit 14 - Disable Hard Error** Prevents network status changes involving "Hard Error" and "Transmit Beacon" bits from causing interrupts.

**Bit 13 - Disable Soft Errors** Prevents network status changes involving "Soft Error" bit from causing interrupts.

**Bit 12 - Pass Adapter MAC Frames** Pass, as direct interface data to the PC system, all adapter class MAC frames that are received but not supported by the adapter. If this bit is off, these frames will be ignored.

**Bit 11 - Pass Attention MAC Frames** Pass, as direct interface data to the PC system, all attention MAC frames that are not the same as the last received attention MAC frame. If this option is off, these frames will not be passed to the application program.

**Bit 10 - Reserved** Should be zero but is not checked by the adapter support software.

**Bit 9 For CCB1 - Pass Parm Table** If the adapter is already open, then the options used when opening the adapter will be returned to the users parameter table.

**Bit 9 For CCB2 and CCB3 - Reserved** Should be zero, but is not checked by the adapter support software.

**Bit 8 - Contender** When the contender bit is on, the adapter will participate in monitor contention (claim token) if the opportunity occurs. When the contender bit is off, and the need is detected by another adapter, this adapter will not participate.

If the need for determining a new active monitor is detected by this adapter, monitor contention (claim token) processing will be initiated by this adapter in either case.

**Bit 7 - Pass Beacon MAC frames** Pass, as direct interface data to the PC system, the first beacon MAC frame and all subsequent beacon MAC frames that have a change in the source address or the beacon type.

**Bit 6 Reserved** Should be zero, but is not checked by the adapter support software.

**Bit 5 - Remote Program Load** This bit is only implemented in 16/4 adapters. It prevents the adapter from becoming a monitor during the open process. If this bit is on, the adapter will fail the open process if there is no other adapter on the ring when it attempts to insert on the ring.

**Bit 4 - Token Release** This bit is only available for 16/4 adapters when operating at 16 Mbps. If not set, 16Mbps adapters will get early token release; that is the default. Setting this bit on, selects no early token release for an adapters operating at 16Mbps.

**For CCB1:** If Token Release is selected by using the command line parameter, this bit will be set on when the DIR.OPEN.ADAPTER command completes.

**Bits 0 through 3 - Reserved** Should be zero, but are not checked by the adapter support software.

See the *IBM Token-Ring Network Architecture Reference* for more about network operation.

---

## NODE\_ADDRESS

**Explanation:** The 6-byte specific node address of this station on the network. The value must not be all ones. The two high-order (leftmost) bits must be B'01'. For other restrictions and details about addresses, see the *IBM Token-Ring Network Architecture Reference*.

If the NODE\_ADDRESS parameter was provided when the adapter support software was loaded, that address will be used rather than the address provided in this parameter field. The address used will be returned in this field by the adapter support software for return to the application program.



## DIR.OPEN.ADAPTER

**For CCB1:** If the value is zero, the address encoded on the adapter will be the node address by default and that value will be placed in this field by the adapter support software for return to the application program.

If the OPEN\_LOCK field is coded as zero then the following will be true.

- If a locally administered address was specified when the adapter support software was loaded, the locally administered address will be used regardless of the contents of this field.
- If a zero was specified when the adapter support software was loaded, the encoded address will be used regardless of the contents of this field.

**For CCB2 and CCB3:** See Appendix E, "Operating System/2 Extended Edition 1.1 Information."

---

### GROUP\_ADDRESS

**Explanation:** Sets the group address for which the adapter will receive messages. If the value is zero, no group address is set.

---

### FUNCTIONAL\_ADDR

**Explanation:** Sets the functional address the adapter will receive messages for. The most significant bit and the least significant bit of this field are ignored by the adapter. If the value is zero, no functional address is set.

**For CCB1:** If NETBIOS is made operational, it will re-issue a DIR.SET.FUNCTIONAL.ADDRESS command using all bits set in the current functional address and adding X'00000080' to the bits being used.

---

### NUMBER\_RCV\_BUFFERS

**Explanation:** The number of receive buffers needed for the adapter to open. The adapter will configure all remaining RAM as receive buffers after other memory requirements have been met. If the number available is less than the number requested, the DIR.OPEN.ADAPTER command will fail. If the number available is greater than the number requested, no action will occur. If this value is less than 2, the default of 8 will be used.

---

### RCV\_BUFFER\_LENGTH

**Explanation:** The length of each of the receive buffers. The value must be a multiple of 8 with 96 as minimum, and a maximum of 2048.

If the value is zero, the default of 112 will be used. Each buffer holds 8 fewer bytes of data than the specified size. Therefore, a buffer defined as 112 bytes long can hold only 104 bytes of data. If a frame received from the network is longer than one buffer, receive buffers will be chained.

---

### DHB\_BUFFER\_LENGTH

**Explanation:** The length of each of the transmit buffers.

For the following Token-Ring Network adapters the maximum DHB size will be 2048.

Token-Ring Network PC Adapters and PC Adapter IIs  
Token-Ring Network Adapter/As

For all new Token-Ring Network adapters operating at 4Mbps the maximum DHB size will be 4464, and for all Token-Ring Network adapters operating at 16Mbps the maximum DHB size will be 17960.

**Note:** If a size greater than 2048 is used, it is important to make sure that all adapters receiving these frames will also be able to handle the larger frame size.

If the value is zero, the default of 600 will be used. Each buffer holds 6 fewer bytes of data than the specified size. Therefore, a buffer defined as 600 can hold only 594 bytes.

### DATA\_HOLD\_BUFFERS

**Explanation:** This defines the number of transmit buffers in the adapter that data from the PC system is stored in. This parameter is not recognized on the PC Network.

The adapter will accept any value between zero and 255, but the integrity of adapter operation cannot be guaranteed if the value is greater than 2. Requesting two buffers may improve adapter performance by allowing a frame to be moved into the second buffer while the adapter is transmitting from the first. However, this will reduce the storage available for receive buffers. Transmit buffers are not chained. If the value is zero, the default of 1 will be used.

### OPEN\_LOCK

**Explanation:** A code provided to the application program to lock the adapter open. Only a DIR.CLOSE.ADAPTER command which has a matching key code can close the adapter. When using this feature you must make sure that the adapter is closed when the application program is finished, or all application programs using the adapter must follow consistent rules about opening and closing the adapter. This field permits one application program to supervise adapter closing when more than one application program or operation has access.

It is recommended that the application program code this field as zero.

This field is not returned by coding the OPEN\_OPTION bit 9.

### PRODUCT\_ID\_ADDRESS

**Explanation:** The address in PC system memory where an 18-byte product ID is located.

The product ID provides indications about the PC system and programs used. The field must contain all zeros, point to a location containing all zeros, or point to a product ID field prepared as shown in Table 3-2.

This field is not returned by coding the OPEN\_OPTION bit 9.

### PRODUCT\_ID\_OFFSET

**Explanation:** The offset in PC system memory where an 18-byte product ID is located.

The product ID provides indications about the PC system and programs used. The field must contain all zeros, point to a location containing all zeros, or point to a product ID field prepared as follows:

Table 3-2. Product ID Field	
Byte 0	X'01' indicates PC system.
Byte 1	X'10'
Bytes 2-5	The machine type from the serial number tag at rear of PC system (the last 4 digits). Enter in EBCDIC. For example, for serial number = 61382, code the field as: (F0 F0 F0 F6 F1 F3 F8 F2)
Bytes 6-17	Zero-filled

### BRING\_UPS

**Explanation:** Bring up error codes. See "Bring-Up Errors for All CCBs" on page B-48 for a list of all valid codes. Indicates the results of Token-Ring Network adapter bring-up testing.

---

### INIT\_WARNINGS

**Explanation:** Processing during initialization may detect errors that are not fatal to the initialization of the adapter support software. These errors will result in the use of default values to configure the adapter support software. This field is used to notify the user that such errors have occurred and were not fixed by the system administrator during system initialization. Below are the values contained in this field.

**X'00'** No errors detected during configuration parameters processing.

**X'01'** No configuration parameters found for adapter.

**X'02'** Errors found during configuration parameters processing, default values used.

---

### SEMAPHORE\_COUNT

**Explanation:** Count of system semaphore handles.

Number of System semaphore handles referenced by the SYS\_SEMAPHORE\_TABLE parameter. If this value is greater than 12 the command will terminate with CCB\_RETCODE set to X'06'.

---

### SYS\_SEMAPHORE\_TABLE

**Explanation:** Address of System semaphore handle table. System semaphore handles that will be used for posting asynchronous events can be registered with the adapter support software prior to using the semaphore handles. Registering System semaphore handles prior to using them will allow the adapter support software to obtain new handles in advance that can be used by its device driver. In doing this, less overhead will exist when a command is issued with one of the registered System semaphore handles in the CCB\_SEMAPHORE field.

A table of System semaphore handles may be passed using this parameter. The table may contain up to twelve 4-byte System semaphore handles that are placed in a control block pointed to by this parameter. The first handle will be located in bytes 0-3 with other handles following (for example, bytes 4-7, bytes 8-11). Values used for System semaphore handles are the values returned from Operating System/2 when the System semaphore is created or opened.

Only the Operating System/2 process that issues the DIR.OPEN.ADAPTER command and provides System semaphore handles can use these handles. All other processes associated with the application program must provide a handle returned from Operating System/2 for the given process when the System semaphore is created or opened.

**Note:** The semaphores must not be created as exclusive.

---

### DDNAME

**Explanation:** This is the name of the application program's device driver that contains the user exits to be called when events complete. The adapter support software will issue an Operating System/2 Device Help Attach DD function call to obtain the application program device driver's intercommunication entry point. The application program's device driver must have enabled inter-device driver communications by setting the Device\_Attribute field of the device driver header (this allows support for the adapter support software to use the ATTACHDD function and obtain the application program's device driver entry point). ATTACHDD is an Operating System/2 device help function. When the application program's device driver issues the DIR.OPEN.ADAPTER command, this parameter specifies the application program's device driver name.

**For CCB1: Direct Params Open Parameters**

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	DIR_BUF_SIZE	2	DW	Size of buffers in "direct" buffer pool
2	DIR_POOL_BLOCKS	2	DW	The length in 16-byte blocks, of buffers in "direct" buffer pool
4	DIR_POOL_ADDRESS	4	DD	Starting segment of "direct" buffer pool
8	ADPT_CHK_EXIT	4	DD	I/O appendage exit, adapter check
12	NETW_STATUS_EXIT	4	DD	I/O appendage exit, network status change
16	PC_ERROR_EXIT	4	DD	I/O appendage exit, error in the PC system
20	WORK_ADDR	4	DD	Adapter work area, segment value. This parameter is not recognized on the PC Network.
24	WORK_LEN_REQ	2	DW	Adapter work area length, requested. This parameter is not recognized on the PC Network.
26	WORK_LEN_ACT	2	DW	Adapter work area length, actual. * This parameter is not recognized on the PC Network.
* Indicates a returned value.				

**DIR\_BUF\_SIZE**

**Explanation:** The size of the buffers in the direct station buffer pool including adapter overhead space. The minimum length is 80 bytes and the length must be a multiple of 16. If the value is zero, the default of 160 is used.

**DIR\_POOL\_BLOCKS**

**Explanation:** The number of 16-byte blocks assigned as the Direct Station buffer pool. If the value is zero, the default of 256 (4096 bytes) is used. If DIR\_POOL\_ADDRESS is zero, this parameter is ignored.

**DIR\_POOL\_ADDRESS**

**Explanation:** The segment value of the address in PC system memory where the adapter support software is to build the Direct Station buffer pool. See "Buffer Pools" on page 2-35. If the value is zero, the application program has the responsibility of building its own buffer pool. If this is the case, the DIR\_BUF\_SIZE parameter must still indicate the size of each buffer.

**ADPT\_CHK\_EXIT**

**Explanation:** The address of a user-provided appendage routine that will be taken when the adapter detects an error in the adapter. If the value is zero, the default is as defined by the DIR.INITIALIZE command.

**NETW\_STATUS\_EXIT**

**Explanation:** The address of a user-provided appendage routine that will be taken when the network status changes. If the value is zero, the default is as defined by the DIR.INITIALIZE command.

**PC\_ERROR\_EXIT**

**Explanation:** The address of a user-provided appendage routine that will be taken when the adapter support software detects an error in the PC system. If the value is zero, the default is as defined by the DIR.INITIALIZE command.

## DIR.OPEN.ADAPTER

---

### WORK\_ADDR

**Explanation:** A segment value in PC system memory where a work area has been allocated for the adapter support software. This value is ignored if the WORK\_LEN\_REQ field is zero. This parameter is not recognized on the PC Network.

---

### WORK\_LEN\_REQ

**Explanation:** The length of the work area in PC system memory, defined by the parameter WORK\_ADDR. The length must be enough to contain all adapter support software work areas defined by the DIR.OPEN.ADAPTER parameters DLC\_MAX\_SAP and DLC\_MAX\_STATIONS. To calculate the PC system memory space needed, use the formula:

PC system RAM (bytes) =  
48 (direct interface stations) plus  
36 times the number of SAPs (DLC\_MAX\_SAP) plus  
6 times the number of stations (DLC\_MAX\_STATIONS)

If this value is 0, the internal work area is used and the WORK\_ADDR field will be ignored. This parameter is not recognized on the PC Network.

---

### WORK\_LEN\_ACT

**Explanation:** The returned value for the length of the work area in PC system memory required by the adapter support software. If the actual length is greater than the work area (internal or specified), the command will terminate with the CCB\_RETCODE field containing: X'12' (available work area exceeded). This parameter is not recognized on the PC Network. It is retained for Token-Ring Network compatibility.

#### For all CCBs: DLC Parms Open Parameters

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	DLC_MAX_SAP	1	DB	Maximum number of SAPs *
1	DLC_MAX_STATIONS	1	DB	Maximum number of link stations *
2	DLC_MAX_GSAP	1	DB	Maximum number of group SAPs *
3	DLC_MAX_GMEM	1	DB	Maximum members per group SAP *
4	DLC_T1_TICK_ONE	1	DB	DLC timer T1 interval, group one *
5	DLC_T2_TICK_ONE	1	DB	DLC timer T2 interval, group one *
6	DLC_Ti_TICK_ONE	1	DB	DLC timer Ti interval, group one *
7	DLC_T1_TICK_TWO	1	DB	DLC timer T1 interval, group two *
8	DLC_T2_TICK_TWO	1	DB	DLC timer T2 interval, group two *
9	DLC_Ti_TICK_TWO	1	DB	DLC timer Ti interval, group two *

\* Indicates a returned value for Operating System/2 (CCB2 and CCB3).

---

### DLC\_MAX\_SAP

**Explanation:** The maximum number of SAPs that can be opened at one time. The maximum value that is allowable may be limited by the amount of available PC system RAM work area in the adapter support software, or by the amount of adapter shared RAM. If NETBIOS is opened, one of the SAPs is required for the adapter support software use and therefore is unavailable for application program use. The maximum value of this parameter is 127 (126 for NETBIOS). If the value is zero, the default is 2.

---

**DLC\_MAX\_STATIONS**

**Explanation:** The maximum number of link stations that can be opened at one time. The maximum value that is allowable may be limited by the amount of available PC system RAM work area in the adapter support software or by the amount of adapter shared RAM. If NETBIOS is opened, some of the link stations will be used by the SAP assigned for adapter support software use. If the value is zero, the default is 6. The maximum value of this parameter is 225 for the Token-Ring Network and the PC Network.

---

**DLC\_MAX\_GSAP**

**Explanation:** The maximum number of group SAPs that can be opened at one time. If the value is zero, no group SAPs are allowed. The maximum value of this parameter is 126 for the Token-Ring Network (125 for the PC Network).

---

**DLC\_MAX\_GMEM**

**Explanation:** The maximum number of SAPs that can be assigned to any given group. The maximum value of this parameter is 127 for the Token-Ring Network (126 for the PC Network).

---

**DLC\_T1\_TICK\_ONE**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the short DLC timer T1 (T1 timer values 1 - 5). If the value is zero, the default of 5 (200 - 400 milliseconds) is used. See "Timers" on page 2-31.

---

**DLC\_T2\_TICK\_ONE**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the short DLC timer T2 (T2 timer values 1 - 5). If the value is zero, the default of 1 (40 - 80 milliseconds) is used. See "Timers" on page 2-31.

---

**DLC\_Ti\_TICK\_ONE**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the short DLC timer Ti (Ti timer values 1 - 5). If the value is zero, the default of 25 (1 - 2 seconds) is used. See "Timers" on page 2-31.

---

**DLC\_T1\_TICK\_TWO**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the long DLC timer T1 (timer values 6 - 10). If the value is zero, the default of 25 (1 - 2 seconds) is used. See "Timers" on page 2-31.

---

**DLC\_T2\_TICK\_TWO**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the long DLC timer T2 (timer values 6 - 10). If the value is zero, the default of 10 (400 - 800 milliseconds) is used. See "Timers" on page 2-31.

---

**DLC\_Ti\_TICK\_TWO**

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for the long DLC timer Ti (timer values 6 - 10). If the value is zero, the default of 125 (5 - 10 seconds) is used. See "Timers" on page 2-31.

---

See page 4-7 for the NETBIOS Open Parameters table.

## DIR.OPEN.DIRECT

Hex 35

DIR.OPEN.DIRECT

**Command Description:****This command is for CCB2 and CCB3 only:**

This command provides a single application program with the capability of receiving frames using the direct station.

**Command Specifics:** An application program is given ownership of the direct station and may receive frames once receive commands have been issued for the direct station. If the direct station has already been assigned to an application program, this command will terminate with CCB\_RETCODE of X'63'. This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

**CCB Parameter Table**

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>For CCB2 and CCB3:</b>				
0	DIR_BUF_SIZE	2	DW	Size of buffers in the buffer pool
2	DIR_POOL_LENGTH	2	DW	Length of buffer pool
4	DIR_POOL_ADDRESS	4	DD	Starting address of buffer pool
8	OPEN_OPTIONS	2	DW	Open options
* Indicates a returned value.				

**DIR\_BUF\_SIZE**

**Explanation:** This is the size of the entire buffer including all overhead.

The minimum length is 80 bytes and the length must be a multiple of 16. If this value is zero the default of 160 is used. The maximum size is dependent on the DIR\_POOL\_LENGTH.

**DIR\_POOL\_LENGTH**

**Explanation:** This is the length of the direct buffer pool. This is the number of 16 byte blocks in the direct buffer pool. If the value is left as zero, the value is ignored.

**DIR\_POOL\_ADDRESS**

**Explanation:** This is the starting address of the buffer pool. If this value is zero, it is the responsibility of the application program to build its own buffer pool. If this is done, DIR\_BUF\_SIZE must reflect the size of these buffers.

---

**OPEN\_OPTIONS**

**Explanation:** See the open options described in Appendix E, "Operating System/2 Extended Edition 1.1 Information."

**Note:** The "wrap interface" and "token release" options are ignored if specified with this command. These options can only be specified with the configuration parameters.

**DIR.READ.LOG**

Hex 08

DIR.READ.LOG

**Command Description:** This command reads and resets the adapter logs. The log data is transferred to the buffer indicated in the parameter table.

**Command Specifics:** If the LOG\_ID is X'0001', this command executes entirely in the PC system. The CCB\_RETCODE is available upon return.

**For CCB2 and CCB3:** This command may be issued by a system administrator using the System Key as defined by the configuration parameters.

The system administrator or application program that issued the DIR.OPEN.DIRECT command and has been assigned ownership of the direct stations may issue this command without the System Key.

The System Key code is located in the CCB\_PARAMETER\_2 field. The System Key code is needed to read the adapter log and the direct interface log if the direct stations have not been assigned to the system administrator. For example, if an application program issued the DIR.OPEN.DIRECT command and has ownership of the direct stations, the System Key code has to match the System Key defined by the configuration parameters in order for a system administrator to read the logs as described above.

Application programs affected by a DIR.READ.LOG issued with the System Key can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.



## CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LOG_ID	2	DW	Identify the log to read
<b>For CCB1:</b>				
2	LOG_BUF_LENGTH	2	DW	Size of the buffer at LOG_BUF_ADDR
4	LOG_BUF_ADDR	4	DD	Address to place log data
8	LOG_ACT_LENGTH	2	DW	Actual length of log *
<b>For CCB2 and CCB3:</b>				
2	LOG_BUF_LENGTH	2	DW	Size of the buffer at LOG_BUF_OFFSET
4	LOG_BUF_OFFSET	2	DW	Offset to place log data
6		2	DW	Reserved for the application program
8	LOG_ACT_LENGTH	2	DW	Actual length of log *
* Indicates a returned value.				

**LOG\_ID**

**Explanation:** Identifies the log to read as follows:

- X'0000' Adapter error log
- X'0001' Direct interface error log
- X'0002' Both logs

**LOG\_BUF\_LENGTH**

**Explanation:**

**For CCB1:** The length of the buffer defined by LOG\_BUF\_ADDR.

**For CCB2 and CCB3:** The length of the buffer defined by LOG\_BUF\_OFFSET.

**LOG\_BUF\_ADDR**

**Explanation:** The address in PC system memory of the buffer (defined by the application program) where the log data is to be placed.

**LOG\_BUF\_OFFSET**

**Explanation:** The offset in PC system memory of the buffer (defined by the application program) where the log data is to be placed.

**LOG\_ACT\_LENGTH**

**Explanation:** The actual length of the log as returned by the adapter or adapter support software. If this value is greater than that defined by the LOG\_BUF\_LENGTH parameter, the full log was not transferred. The CCB\_RETCODE will contain X'15'.

**Note:** When data is lost (CCB\_RETCODE X'15'), the log is still reset.

## Log Formats

- Adapter log

When one or more log counters reaches 255, the application program network status appendage routine will be taken with network status indicating a counter overflow. These counters are reset after they are read.

The information read from this log is 14 bytes long and returned to the buffer in this order:

Byte	8086 Type	Token-Ring Network	PC Network
0	DB	Line Error	CRC Errors
1	DB	Internal error	Reserved
2	DB	Burst error	Alignment errors
3	DB	A/C error	Reserved
4	DB	Abort delimiter	Transmit errors
5	DB	Reserved	Reserved
6	DB	Lost frame	Collision Errors
7	DB	Receive congestion	Receive congestion errors
8	DB	Frame copied error	Reserved
9	DB	Frequency error	Reserved
10	DB	Token error	Reserved
11	DB	Reserved	Reserved
12	DB	Reserved	Reserved
13	DB	Reserved	Reserved

- Direct Interface Log

The direct interface log consists of 18 bytes of counters and they are returned to the buffer in this order:

Bytes	8086 Type	Meaning
0-3	DD	Number of frames transmitted
4-7	DD	Number of frames received
8-11	DD	Number of frames discarded (no receive command)
12-15	DD	Number of times data was lost
16-17	DW	Number of buffers available in the SAP buffer pool

- Both Adapter and Direct Interface logs

Both logs will be placed in the buffer. The adapter log is first, followed by the direct interface log.

## DIR.SET.EXCEPTION.FLAGS

### DIR.RESTORE.OPEN.PARMS

Hex 02

#### DIR.RESTORE.OPEN.PARMS

**Command Description:**

**This command is for CCB1 only:**

This command is used to restore adapter open parameter values modified by the DIR.MODIFY.OPEN.PARMS command.

**Command Specifics:** This command is rejected with a X'06' return code if a DIR.MODIFY.OPEN.PARMS has not previously been issued and completed successfully.

No parameter table is required. The parameter table pointer field is used as adapter work space.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

### DIR.SET.EXCEPTION.FLAGS

Hex 2D

#### DIR.SET.EXCEPTION.FLAGS

**Command Description:**

**This command is for CCB2 and CCB3 only:**

This command defines user notification appendages and flags if an application program wants to be notified of exception conditions.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software. If the appendages or flags are zero, the appendage flag is not defined and the application program will not be notified of the exception condition.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

**CCB Parameter Table**

**Note:** See "Exception Indications" on page B-40 for more information.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>For CCB2:</b>				
0	ADAPTER_CHECK_FLAG	4	DD	User notification flag for adapter check
4	NETWORK_STATUS_FLAG	4	DD	User notification flag for network status
8	PC_ERROR_FLAG	4	DD	User notification flag for PC system errors
12	SYSTEM_ACTION_FLAG	4	DD	User notification flag for system action
<b>For CCB3:</b>				
0	ADAPTER_CHECK_APPNDG_OFFSET	2	DW	Offset to the adapter check appendage
2		2	DW	Reserved for the application program
4	NETWORK_STATUS_APPNDG_OFFSET	2	DW	Offset to the network status appendage
6		2	DW	Reserved for the application program
8	PC_ERROR_APPNDG_OFFSET	2	DW	Offset to the PC system error appendage
10		2	DW	Reserved for the application program
12	SYSTEM_ACTION_APPNDG_OFFSET	2	DW	Offset to the system action appendage
14		2	DW	Reserved for the application program

**Parameters**

**For CCB2:** The flag parameters are used to request that information relating to exception events be passed to the application program. It is recommended that this command be issued right after issuing the DIR.OPEN.ADAPTER command if the application program wants to be notified of any exception conditions. A READ command should then be issued with EVENT\_SET bits 4 and 5 on. Whenever a critical exception or non-critical exception occurs, this READ command will complete and the posting event and event error code will be returned from the adapter support software.

**For CCB3:** These offsets to application program appendages are used to request that information relating to exception events be passed to the application program. These parameters will consist of a 2 byte offset for the exception appendage that will be passed in register DI to the application program when the adapter support software calls the application program. It is recommended that this command be issued right after issuing DIR.OPEN.ADAPTER if the application program wants to be notified of any exception conditions.

## DIR.SET.GROUP.ADDRESS

## DIR.SET.FUNCTIONAL.ADDRESS

Hex 07

### DIR.SET.FUNCTIONAL.ADDRESS

**Command Description:** This command sets the functional addresses for the adapter to receive messages. This command may be used to change (reset) the functional address values that were set earlier.

**Command Specifics:** If the most significant bit of the functional address field is off (0), the bits in the adapter functional address corresponding to the bits provided in the functional address field of the CCB are set on. If the most significant bit of the field is on (1), the bits in the adapter functional address corresponding to the bits provided in the functional address field are reset. Bits 31, 1, and 0 are not affected by this command but are set by the adapter.

**For CCB1:** The functional address bits to be changed are placed in the CCB\_PARM\_TAB field, but is formatted as DB. The least significant bit is ignored and both the least and most significant bit will always be set to zero in the adapter. For example, X'FFFFFFFF' will reset all functional address bits or X'00000060' will reset bits 5 and 6.

**For CCB2 and CCB3:** The functional address bits to be set or reset are placed in the CCB\_PARM\_OFFSET and CCB\_PARAMETER\_1 field but is formatted as DB. For example, X'00000060' will set bits 5 and 6, or X'80000040' to reset bit 6.

This command may be issued by a system administrator using the System Key as defined by the configuration parameters.

This function is provided to allow an application program to set bits of the functional address. Bits that have been previously set by application programs are the only bits that can be reset without the System Key. The bits set by the configuration parameters cannot be reset without the System Key. The System Key, if used, should be coded in the CCB\_PARAMETER\_2 field.

Application programs affected by a DIR.SET.FUNCTIONAL.ADDRESS issued with the System Key can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.SET.GROUP.ADDRESS

Hex 06

### DIR.SET.GROUP.ADDRESS

**Command Description:** This command sets the group addresses for the adapter to receive messages. This command may be used to change the group address values that were set earlier.

**Command Specifics:** If no group address is desired, set the value to X'00000000'.

**For CCB1:** The group address to be set is placed in the CCB\_PARM\_TAB field but is formatted as DB.

**For CCB2 and CCB3:** The group address to be set is placed in the CCB\_PARM\_OFFSET field but is formatted as DB. This command can only be issued by a system administrator using the System Key as defined by the configuration parameters. The System Key should be coded in the CCB\_PARAMETER\_2 field.

Application programs affected by a DIR.SET.GROUP.ADDRESS can be notified of the event. See "System Action Exceptions for OS/2" on page B-58 for more information.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.SET.USER.APPENDAGE

Hex 2D

### DIR.SET.USER.APPENDAGE

#### Command Description:

**This command is for CCB1 only:**

This command changes the appendage addresses set by the DIR.OPEN.ADAPTER, DIR.INITIALIZE, and DIR.MODIFY.OPEN.PARMS commands.

**Command Specifics:** This command will set the appendage addresses as defined in the CCB parameter table if the CCB\_PARM\_TAB field value is other than zero. If the CCB\_PARM\_TAB field is set to zero, the appendage addresses will be restored to their values before the **last** DIR.SET.USER.APPENDAGE command was completed successfully.

This command may be issued only when the adapter is open. If the adapter was opened with an OPEN\_LOCK, then this command will take no action and will complete with a return code of X'00'. This command is executed entirely in the adapter support software in the PC system. Therefore, the command completion appendage is not required, as the command is complete upon return. However, the command completion appendage will be taken if provided.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	ADPT_CHK_EXIT	4	DD	I/O appendage, adapter check
4	NETW_STATUS_EXIT	4	DD	I/O appendage, network status change
8	PC_ERROR_EXIT	4	DD	I/O appendage, error in PC system

Refer to the parameter descriptions of the DIR.OPEN.ADAPTER command beginning on page 3-34.

## DIR.STATUS

Hex 21

## DIR.STATUS

**Command Description:** This command reads the general status information.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**For CCB1:** The status information is pointed to by the address in the CCB\_PARM\_TAB field when the command is completed.

**For CCB2 and CCB3:** The status information is pointed to by the offset in the CCB\_PARM OFFSET field when the command is completed.

If a valid CCB\_APPL\_ID is not being used, only polling of the CCB\_RETCODE can be used for posting of this command completion. In this case, a successful completion will result in a return code of X'0C'. If a valid CCB\_APPL\_ID is used, this command can be posted like any other command. A successful completion will result in a return code of X'00'.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	ENCODED_ADDR	6	DB	Adapter's permanent encoded address *
6	NODE_ADDRESS	6	DB	This adapter's network address *
12	GROUP_ADDRESS	4	DB	This adapter's group address *
16	FUNCTIONAL_ADDR	4	DB	This adapter's functional address *
20	MAX_SAP	1	DB	Maximum allowable SAPs *
21	OPEN_SAP	1	DB	Number of SAPs open *
22	MAX_STATION	1	DB	Maximum allowable link stations *
23	OPEN_STATION	1	DB	Number of link stations open *
24	AVAIL_STATION	1	DB	Number of link stations available *
25	ADAPTER_CONFIG	1	DB	Adapter configuration bits *
26	MICROCODE_LEVEL	10	DB	The adapter's microcode level *
<b>For CCB1:</b>				
36	ADAPTER_PARMS_ADDR	4	DD	Shared RAM address of Adapter Parms *
40	ADAPTER_MAC_ADDR	4	DD	Shared RAM address of Adapter MAC buffer *
44	TICK_CNTR_ADDR	4	DD	Address of the timer tick counter *
48	LAST_NTWK_STATUS	2	DW	Most recent network status issued *
<b>For CCB2:</b>				
36	ADAPTER_PARMS_ADDR	4	DD	Address of adapter parms *
40	ADAPTER_MAC_ADDR	4	DD	Address of adapter MAC buffer *
44	TICK_CNTR_ADDR	4	DD	Address of the timer tick counter *
* All entries are returned by the adapter support software to the parameter table in PC system memory.				

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
48	LAST_NTWK_STATUS	2	DW	Most recent network status issued *
50	ADAPTER_TYPE	2	DW	Network adapter type *
<b>For CCB3:</b>				
36	ADAPTER_PARMS_ADDR	4	DD	Address with GDT selector of adapter parms *
40	ADAPTER_MAC_ADDR	4	DD	Address with GDT selector of adapter MAC buffer *
44	TICK_CNTR_ADDR	4	DD	Address with GDT selector of the timer tick counter *
48	LAST_NTWK_STATUS	2	DW	Most recent network status issued *
50	ADAPTER_TYPE	2	DW	Network adapter type *
* All entries are returned by the adapter support software to the parameter table in PC system memory.				

---

### ENCODED\_ADDR

**Explanation:** The permanent address encoded on the adapter.

---

### NODE\_ADDRESS

**Explanation:**

**For CCB1:** The adapter's network address as set by the DIR.OPEN.ADAPTER command.

**For CCB2 and CCB3:** The adapter's network address as set by the configuration parameters.

---

### GROUP\_ADDRESS

**Explanation:** The adapter's group address.

---

### FUNCTIONAL\_ADDR

**Explanation:** The adapter's functional address.

---

### MAX\_SAP

**Explanation:** The maximum number of SAPs allowed.

---

### OPEN\_SAP

**Explanation:** The number of SAPs that have been opened by DLC.OPEN.SAP commands.

---

### MAX\_STATION

**Explanation:**

**For CCB1:** The maximum number of link stations allowed as set by the DIR.OPEN.ADAPTER command.

**For CCB2 and CCB3:** The maximum number of link stations allowed as set by the configuration parameters.

---

### OPEN\_STATION

**Explanation:** The number of link stations that have been opened by DLC.OPEN.STATION commands or the DLC.CONNECT.STATION command as a result of receiving a SABME.



## DIR.STATUS

---

### AVAIL\_STATION

**Explanation:** The number of link stations that have not been previously reserved by DLC.OPEN.SAP or DLC.REALLOCATE commands.

---

### ADAPTER\_CONFIG

**Explanation:** Below are the descriptions of the adapter configuration bits:

- Bit 7** For CCB1: The Original PC Network Adapter is present.  
For CCB2 and CCB3: Reserved
- Bit 6** Reserved
- Bit 5** Reserved
- Bit 4** Token release: This bit is only available for 16/4 adapters when operating at 16 Mbps. If not set, 16Mbps adapters will get early token release; that is the default. Setting this bit on, selects no early token release for an adapters operating at 16Mbps.
- Bits 3,2** Shared RAM mapped into PC system memory (B'00' = 8K, B'01' = 16K, B'10' = 32K, and B'11' = 64K. For example, if bits 3 and 2 are 0 and 1 respectively then Shared RAM size is 16K.)
- Note:** This value does not necessarily indicate the total shared RAM available on the adapter.
- Bit 1** Reserved
- Bit 0** Adapter data rate (0 = 4Mbps and 1 = 16Mbps)
- 

### MICROCODE\_LEVEL

**Explanation:** The number representing the engineering level of the adapter microcode.

---

### ADAPTER\_PARMS\_ADDR

**Explanation:** The address of a PC system read-only region of adapter parameters.

Refer to "Adapter Status Parameter Table" on page B-38 for the parameter table layout.

**For CCB2:** An entry for this memory segment is made into the local descriptor table (LDT) of the process that issues this command. Only the process that issues this command will have access to this memory area.

**For CCB3:** This memory segment has been mapped in the global descriptor table (GDT).

---

### ADAPTER\_MAC\_ADDR

**Explanation:** The address of a PC system read-only region of adapter MAC buffer.

**For CCB2:** An entry for this memory segment is made into the local descriptor table (LDT) of the process that issues this command. Only the process that issues this command will have access to this memory area.

**For CCB3:** This memory segment has been mapped in the global descriptor table (GDT).

---

### TICK\_CNTR\_ADDR

**Explanation:**

**For CCB1:** The address of a 4-byte field containing the number of 100 millisecond timer interrupts received from the adapter's timer since the last DIR.INITIALIZE command. The tick counter may be read, but should not be written.

**For CCB2:** The address of a 4-byte field containing the number of 100 millisecond timer interrupts received

from the adapter's timer since the last physical open by a DIR.OPEN.ADAPTER command. The tick counter may be read, but should not be written.

An entry for this memory segment is made into the local descriptor table (LDT) of the process that issues this command. Only the process that issues this command will have access to this memory area.

**For CCB3:** The address of a 4-byte field containing the number of 100 millisecond timer interrupts received from the adapter's timer since the last physical open by a DIR.OPEN.ADAPTER command. The tick counter may be read, but should not be written.

This memory segment has been mapped in the global descriptor table (GDT).

## LAST\_NTWK\_STATUS

**Explanation:** The most recent network status change.

## ADAPTER\_TYPE

**Explanation:**

**For CCB2 and CCB3:** The network adapter type as indicated below:

X'0001'	Token-Ring Network PC Adapter
X'0002'	Token-Ring Network PC Adapter II
X'0004'	Token-Ring Network Adapter/A
X'0008'	Token-Ring Network PC Adapter II
X'0020'	Token-Ring Network 16/4 Adapter
X'0040'	Token-Ring Network 16/4 Adapter/A
X'0080'	Token-Ring Network Adapter/A
X'4000'	PC Network adapters
X'8000'	PC Network adapters

## DIR.TIMER.CANCEL

Hex 23

DIR.TIMER.CANCEL

**Command Description:** This command cancels a timer that was previously initiated by a DIR.TIMER.SET command.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

When the DIR.TIMER.CANCEL command is completed, the DIR.TIMER.SET command is terminated with a CCB\_RETCODE value of X'0A', and the DIR.TIMER.SET command is not posted except for the setting of the return code. The following CCB field points to the address of the DIR.TIMER.SET command that is to be cancelled.

**For CCB1:** The CCB\_PARM\_TAB field.

**For CCB2 and CCB3:** The CCB\_PARM\_OFFSET and CCB\_PARAMETER\_1 fields are combined to form a four byte (8086 declaration type DD) address of the timer to be cancelled.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.TIMER.CANCEL.GROUP

Hex 2C

### DIR.TIMER.CANCEL.GROUP

**Command Description:** This command cancels a group of timer commands that were previously initiated by DIR.TIMER.SET commands.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

**For CCB1:** The CCB\_PARM\_TAB field points to the address of a timer completion appendage. All timers that have that address as a timer completion appendage will be terminated with a return code of X'0A'. When the DIR.TIMER.SET command is terminated, the command completion appendage will not be taken. All canceled CCBs will be pointed to by the CCB\_POINTER field of the DIR.TIMER.CANCEL.GROUP command. This command may be issued while the adapter is closed.

**For CCB2:** The DIR.TIMER.CANCEL.GROUP command's CCB\_PARM\_OFFSET and CCB\_PARAMETER\_1 fields contain a TIMER\_CMPL\_FLAG. All timer set commands that have been issued with their CCB\_CMPL\_FLAG equal to the TIMER\_CMPL\_FLAG will be cancelled with a CCB\_RETCODE of X'0A'.

No command completion is done for the DIR.TIMER.SET commands other than setting the CCB\_RETCODE to X'0A'. The DIR.TIMER.SET command's CCBs are chained using the CCB\_POINTER field of the DIR.TIMER.CANCEL.GROUP command CCB.

**For CCB3:** The DIR.TIMER.CANCEL.GROUP command's CCB\_PARM\_OFFSET and CCB\_PARAMETER\_1 fields contain a TIMER\_CMPL\_OFFSET. This offset is used to specify the address of the command completion appendage (CCB\_APPNDG\_OFFSET) of all outstanding DIR.TIMER.SET commands that are to be cancelled. These commands will be cancelled with a CCB\_RETCODE of X'0A'.

No command completion is done for the DIR.TIMER.SET commands other than setting the CCB\_RETCODE to X'0A'. The DIR.TIMER.SET command's CCBs are chained using the CCB\_POINTER field of the DIR.TIMER.CANCEL.GROUP command CCB.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## DIR.TIMER.SET

Hex 22

DIR.TIMER.SET

**Command Description:** This command sets a special programmable timer to expire in some multiple of half-second periods. On the PC Network, the value is not actually half-seconds, but 491.5 millisecond periods.

- For the Token-Ring Network: From 0 to 13,107 (109 minutes)
- For the PC Network: From 0 to 13,107 (107.3 minutes)

The timer expires after the next tick if the value is set to zero.

### Command Specifics:

**For CCB1:** The timer value is coded in the first two bytes of the CCB\_PARM\_TAB field.

When the specified time expires, the command completes, with the CCB\_RETCODE set to X'00'.

This command may be issued any time after the adapter has been initialized while the adapter is either open or closed, but if a command follows that changes the adapter open/closed state, all outstanding timers will be canceled. The number of timers that may be set is limited only by the number of commands that may be outstanding.

**For CCB2:** The timer value is coded in the CCB\_PARM\_OFFSET field.

When the allotted time expires, the command completes as defined by the CCB\_CMPL\_FLAG and the CCB\_SEMAPHORE parameters with the CCB\_RETCODE set to X'00'.

**For CCB3:** The timer value is coded in the CCB\_PARM\_OFFSET field.

When the allotted time expires, the command completion appendage will be called, if defined. The CCB\_RETCODE is set to X'00'.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## PDT.TRACE.ON

Hex 24

PDT.TRACE.ON

### Command Description:

**This command is for CCB1 only:**

This command provides an interrupt trace for all adapter traffic.

**Command Specifics:** The command provides entries for the following activities:

- Each CCB when it is issued to the adapter if the initial return code is X'FF'.
- Each CCB completion
- Each NCB when issued by the application program (return code = X'FF')
- All adapter interrupts to the PC system. If the interrupt is a timer interrupt only, a trace entry is not made, but the timer interrupts are counted for reporting. Then when a non-timer interrupt occurs, a

## PDT.TRACE.ON

timer trace entry is made containing the accumulated timer interrupts followed by a trace entry for the non-timer interrupt.

Only one trace command can be active. The trace includes all activity for either primary or alternate adapter or both. The command is terminated by either a PDT.TRACE.OFF command or an exception, when issued.

This command is executed entirely in the adapter support software in the PC system. Therefore, the command completion appendage is not required, as the command is complete upon return. However, the command completion appendage will be taken if provided.

The CCB\_ADAPTER field of the CCB may be any value between X'00' and X'03'.

The location of the trace table is pointed to by the value placed in the CCB\_PARM\_TAB field by the application program. The trace will wrap when the buffer fills.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>For CCB1 only:</b>				
0	TABLE_LENGTH	2	DW	Length of the TRACE_TABLE
2	CURRENT_OFF	2	DB	Offset of the current trace entry *
4	START_TICK_0	4	DD	Adapter 0 timer tick count, trace start *
8	STOP_TICK_0	4	DD	Adapter 0 timer tick count, trace stop *
12	START_TICK_1	4	DD	Adapter 1 timer tick count, trace start *
16	STOP_TICK_1	4	DD	Adapter 1 timer tick count, trace stop *
20		12	--	Adapter work area
32		--	--	Trace table
* Indicates a returned value.				

---

### TABLE\_LENGTH

**Explanation:** The length of the requested trace table with a minimum value of 256. The entries are 16 bytes long. If the length specified is not a multiple of 16, the last 1-15 bytes will be unused.

---

### CURRENT\_OFF

**Explanation:** The offset from the TRACE\_TABLE value of the most recent table entry.

The table will wrap around when full. If the first entry in the table is the PDT.TRACE.ON command, the table has not wrapped.

---

### START\_TICK\_0

**Explanation:** The value of the Adapter 0 timer tick counter, as set by the adapter, when the trace started. Adapter ticks occur every 100 milliseconds.

---

### STOP\_TICK\_0

**Explanation:** The value of the Adapter 0 timer tick counter, as set by the adapter, when the trace stopped.

---

**START\_TICK\_1**

**Explanation:** The value of the Adapter 1 timer tick counter, as set by the adapter, when the trace started.

---

**STOP\_TICK\_1**

**Explanation:** The value of the Adapter 1 timer tick counter, as set by the adapter, when the trace stopped.

---

**TRACE\_TABLE**

**Explanation:** The trace table starts here. The length is defined by the TABLE\_LENGTH field. The trace table formats for the Token-Ring Network and the PC Network are explained below:

**Trace Table Formats for Token-Ring Network**

Four trace entry formats are used and each trace entry is 16 bytes long.

The SS and SP registers point to 26 bytes of stack space used by the adapter support software when the trace entry is made.

**1. CCB Trace Entry**

Byte	Meaning
0	Adapter number (0/1)
1	Flags
2	CCB Command
3	Return code
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

The command code (byte 2) will be zero if not applicable. If this entry is the result of a request for data by the adapter following a Transmit CCB, the specific transmit command code will be inserted.

Byte 1: Flags:

Bit	Meaning
7	Adapter initialized
6	Initialize in process
5	Adapter opened
4	Open in process
3	SRB busy
2	Block bit on
1	Always 0
0	No adapter

The ES and BX registers point to the CCB.

**2. Adapter Interrupt Trace Entry (except timer)**

Byte	Token-Ring Network
0	ISRP Even
1	ISRP Odd
2	Command code of the interrupt
3	Return code
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

**Byte 0: Interrupt Status Register, Processor (ISRP) Even**

Bit	Meaning
7	CHCK/IRQ steering control (always 1)
6	Interrupt enabled (always 1)
5	Reserved
4	Timer Interrupt (100-millisecond programmable timer)
3	Error Interrupt
2	Access Interrupt
1	Always on
0	Adapter number (0 or 1)

**Byte 1: Interrupt Status Register, Processor (ISRP) Odd**

Bit	Meaning
7	Reserved
6	Adapter check
5	SRB response
4	ASB free
3	ARB command
2	SSB response
1	Reserved
0	Reserved

The ES and BX registers point to the applicable CCB or buffer if the interrupt is a result of a CCB, otherwise these bytes will be zero.

### 3. Adapter Timer Interrupt Trace Entry

Byte	Meaning
0	ISRP Even, adapter 0=X'D2' and adapter 1=X'D3'
1	X'00'
2-3	2-byte counter: number of timer interrupts, adapter 0 and 1
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

When no other interrupts are occurring, timer interrupts are maintained for reporting. The counter is updated for each timer interrupt (either adapter). The accumulated interrupts will be placed in a timer interrupt trace entry when a non-timer interrupt occurs producing another trace entry. The ES and BX registers point to the DIR.TIMER.SET CCB if this timer interrupt causes a DIR.TIMER.SET command to be completed. Otherwise, this field will be zero.

### 4. NCB Trace Entry

Byte	Meaning
0	X'0F' X'1F' X'2F' (see below)
1	Adapter number (0/1)
2-3	NCB command and return code
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

The ES and BX registers point to the NCB.

When a post routine is used while trace is active, three entries are made in the trace table.

Byte 0 contains:

- X'0F' for the entry when the NCB is first issued
- X'1F' when going to the user-supplied post routine
- X'2F' when returning from the post routine.

#### Trace Table Formats for the PC Network

Three trace entry formats are used and each trace entry is 16 bytes long.

The SS and SP registers point to 26 bytes of stack space used by the adapter support software when the trace entry is made.



**1. CCB Trace Entry**

Byte	Meaning
0	Adapter number (0/1)
1	Flags
2	CCB Command
3	Return code
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

The command code (byte 2) will be zero if not applicable. If this entry is the result of a request for data by the adapter following a Transmit CCB, the specific transmit command code will be inserted.

Byte 1: Flags:

Bit	Meaning
7	Adapter initialized
6	Initialize in process
5	Adapter opened
4	Open in process
3	Command request block busy
2	Reserved
1	Reserved
0	No adapter

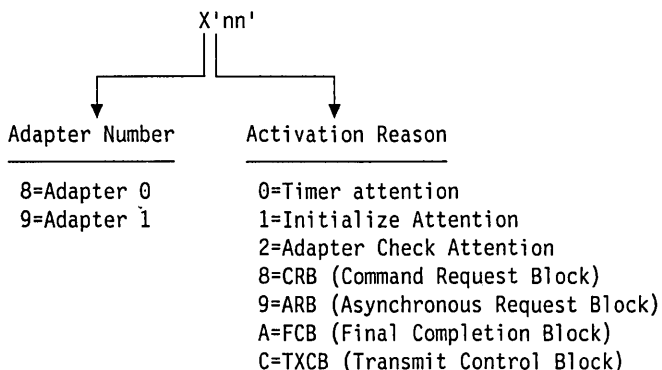
The ES and BX registers point to the CCB.

**2. Interrupt Trace Entry**

The interrupt trace entry structure will be described in general followed by each possible entry described separately.

Byte	PC Network
0	Adapter number (0/1) and activation reason code
1	Command code or X'00'
2	Command dependent information
3	Command dependent information
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

Byte 0: Adapter number (0/1) and activation reason code



The ES and BX registers point to the applicable CCB or buffer if the interrupt is a result of a CCB, otherwise these bytes will be zero.

**Timer Entry**

Below is the structure for the Timer Entry. This is when Byte 0 is X'80' or X'90'.

0	1	2-3	4-7	8-11	12-15
X'80' & X'90'	X'00'	Tick counter: Adapter 0 and 1	SS:SP	Address of interrupted code	ES:BX

**Note:** Timer interrupts with no other interrupts occurring are maintained in one trace entry. The counter is updated for each timer interrupt (either adapter) and the other values in the trace represent the latest trace entry.

**Initialize Entry**

Below is the structure for the Initialize Entry. This is when Byte 0 is X'81' or X'91'.

0	1	2-3	4-7	8-11	12-15
X'81' & X'91'	X'00'	X'0000'	SS:SP	Address of interrupted code	ES:BX

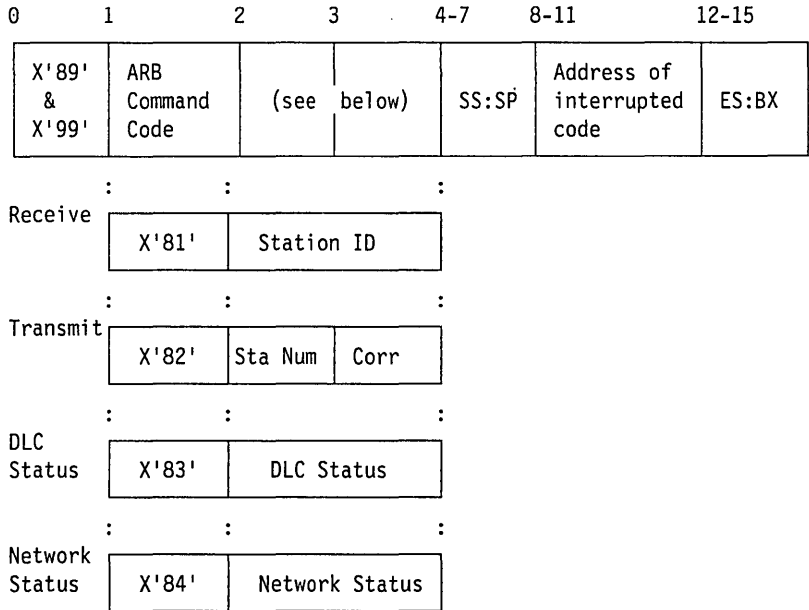
**CRB Entry**

Below is the structure for the Command Request Block Entry. This is when Byte 0 is X'88' or X'98'.

0	1	2	3	4-7	8-11	12-15
X'88' & X'98'	CRB Command Code	Return Code	Correlator (00 if RC= X'FF')	SS:SP	Address of interrupted code	ES:BX

**ARB Entry**

Below is the structure for the Asynchronous Request Block Entry. This is when Byte 0 is X'89' or X'99'.

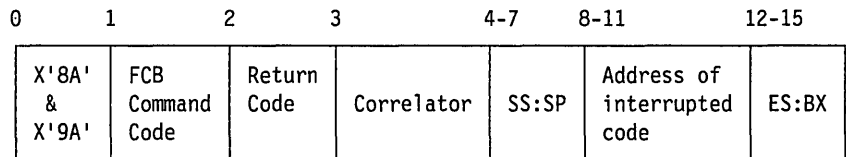


For the contents of the ES:BX registers, see below:

- For a Receive this will point to the Receive CCB if the Receive is completing (no Receive.Data appendage or a bad return code), else it will point to the first receive buffer if a Receive.Data appendage is specified.
- For a Transmit this will point to the Transmit CCB.
- For a DLC Status this will point to the DLC Status Buffer.
- For a Network Status this will point to a queue of outstanding CCBs if the adapter has been closed or it will be zero.

**FCB Entry**

Below is the structure for the Final Completion Block Entry. This is when Byte 0 is X'8A' or X'9A'.



**TXCB Entry**

Below is the structure for the Transmit Control Block Entry. This is when Byte 0 is X'8C' or X'9C'.

0	1	2	3	4-7	8-11	12-15
X'8C' & X'9C'	TXCB Command Code	Return Code	X'00'	SS:SP	Interrupted	ES:BX

**3. NCB Trace Entry**

Byte	Meaning
0	X'0F' X'1F' X'2F' X'3F' (see below)
1	Adapter number (0/1)
2-3	NCB command and return code
4-7	SS:SP registers
8-11	Address of the interrupted application program code
12-15	ES:BX registers

The ES and BX registers point to the NCB.

If there is no NCB post routine specified, there will be no X'2F' entry.

Byte 0 contains:

- X'0F' for the entry when the NCB is first issued
- X'1F' when posting the NCB
- X'2F' when returning from a post routine
- X'3F' when NETBIOS indicates finished with NCB

**PDT.TRACE.OFF**

Hex 25
PDT.TRACE.OFF

**Command Description:**

**This command is for CCB1 only:**

This command terminates the PDT.TRACE.ON command.

**Command Specifics:** The results of the PDT.TRACE.OFF command (when successful) are:

- The PDT.TRACE.ON command's CCB\_RETCODE field is set to X'0A'.
- The PDT.TRACE.OFF command's CCB\_RETCODE field is set to X'00'.
- The PDT.TRACE.OFF command's CCB\_POINTER field contains the address of the PDT.TRACE.ON CCB that was terminated.
- The CCB\_CMD\_CMPL completion exit of the PDT.TRACE.OFF command, if defined, is taken.
- The CCB\_CMD\_CMPL completion exit of the PDT.TRACE.ON command is ignored.

If a PDT.TRACE.ON command is not outstanding:

- The PDT.TRACE.OFF command's CCB\_RETCODE field is set to X'00'.

## PURGE.RESOURCES

- The PDT.TRACE.OFF command's CCB\_POINTER field contains 0.
- The CCB\_CMD\_CMPL completion exit of the PDT.TRACE.OFF command, if defined, is taken.

This command is executed entirely in the adapter support software in the PC system. Therefore, the command completion appendage is not required, as the command is complete upon return. However, if the command completion appendage is provided, it will be used as defined above.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

## PURGE.RESOURCES

Hex 36

### PURGE.RESOURCES

#### Command Description:

**This command is for CCB3 only:**

This command allows the application program to purge resources from the adapter support software that are owned by a terminating OS/2 process. This command can also be used to purge resources from the adapter support software that are allocated from memory being returned to the operating system.

The application program must use this command if all the following hold true:

- The application program is using the Device Driver Interface.
- The application program has passed the adapter support software control blocks allocated from memory owned by one or more OS/2 processes.
- The application program wishes to continue using the adapter support software services after one of its processes terminates. In this case the application program does not wish to issue the DIR.CLOSE.ADAPTER command.
- The application program wishes to free memory to the operating system but still has CCBs and buffers outstanding with the adapter support software but does not wish to issue the DIR.CLOSE.ADAPTER command.

#### Command Specifics:

1. Memory passed to an adapter support software device driver (LANDD) in the form of CCBs or buffers can be owned by different processes of an application program or by an application program's device driver. Since the LAN device drivers cannot guarantee that the active process is the owner of memory being passed to it (command invocations at interrupt time) all control blocks and buffers will be associated with a Resource ID. The Resource ID will be passed as a parameter with the CCB and will be associated with the CCB and all other control blocks (for example, logs and buffers) referenced by the CCB.

In order to have CCBs that have been purged returned to the application program, this command must be used specifying a Command Completion appendage.

An application program may wish to purge resources when it consists of more than one OS/2 process and each process has resources. For example, an application program has two OS/2 processes that have both been allocated memory (being used in the application program's SAP buffer pool). When one of the processes terminates the application program should use the PURGE.RESOURCES command to specify to the adapter support software what control blocks should be removed (cleaned up) from the adapter support software's internal queues.

2. To identify the resource ID (PURGE\_RESOURCE\_ID) of the control blocks to be purged, use the CCB\_PARAMETER\_2 field of the PURGE.RESOURCES CCB.

All control blocks that have been passed to the adapter support software with a resource ID that matches PURGE\_RESOURCE\_ID will be removed from the adapter support software internal queues and returned to the application program. All outstanding CCBs that are cancelled will be chained using the CCB\_POINTER fields with the CCB\_RETCODE field set to X'67'. When the adapter support software calls the application program's device driver (with the appropriate event appendage offset passed in register DI) and there are no resources associated with the PURGE.RESOURCE.ID, the 12-byte information table referenced by registers DX and DI will contain zeros.

3. If the application program does not purge a terminating process' resources immediately from the LAN device driver's internal queues other application programs can be adversely affected. If the memory owned by the terminating process is reallocated by OS/2 to another process the memory can be written into by the adapter support software if the terminating process's control blocks are not removed from the adapter support software's internal queues.
4. This command does not close stations, SAPs, or application programs, it only cleans up data areas.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

## READ

Hex 31

READ

### Command Description:

**This command is for CCB2 only:**

This command performs a read function for the DIRECT and DLC interfaces. The STATION\_ID is used to define the interface that the read is being issued for.

**Command Specifics:** When a READ command is issued, it is queued to wait for the specified read event for the STATION\_ID indicated in the command. Multiple READ commands may be outstanding at one time for each STATION\_ID. Each READ command specifies an event or set of events and a system semaphore handle (CCB\_SEMAPHORE). If an event has already occurred that matches an event specified with a READ command or at some later time an event occurs that matches an event specified with a READ command, the following will take place:

1. Data associated with the event will be copied into the READ command's CCB parameter table.
2. CCB\_RETCODE of the READ command CCB will be set to X'00'.
3. CCB\_SEMAPHORE, if provided, will be cleared signalling that the READ command has completed.

Once the READ command's CCB\_RETCODE has been set and CCB\_SEMAPHORE cleared, the READ command has completed and must be issued again to become active.

There are cases where an application program may wish to guarantee that a READ command is outstanding for a particular CCB command completion. For example, when a DIR.CLOSE.ADAPTER completes, all the application program's outstanding CCBs, SAP buffers, and receive frames are returned in the parameter table of an outstanding READ command. If no READ command is outstanding, the application program's data areas will not be returned. In order to guarantee that a READ command is outstanding, the application program may chain a READ command's CCB to a CCB by placing the address of the READ command's CCB in the CCB\_POINTER field and by setting the CCB\_READ\_FLAG field to a non-zero value. When the CCB completes, the READ command referenced by the CCB\_POINTER field will be used to post the completion of the command. In addition to updating the READ command's CCB parameter table, the READ command's CCB\_RETCODE will be set and CCB\_SEMAPHORE will be cleared (if it exists).

## READ

Valid Return Codes: see "CCB Return Codes Listed by Command" on page B-4.

### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>For CCB2 only:</b>				
0	STATION_ID	2	DW	SAP station ID
2	OPTION_INDICATOR	1	DB	READ option indicator
3	EVENT_SET	1	DB	Set of events for notification
4	EVENT	1	DB	Posting event *
5	CRITICAL_SUBSET	1	DB	Critical event subset identifier *
6	NOTIFICATION_FLAG	4	DD	Event user notification flag *
10	CCB_COUNT	2	DW	Count of CCBs chained to the EVENT_CCB_POINTER *
12	EVENT_CCB_POINTER	4	DD	Pointer to CCBs *
16	BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR *
18	FIRST_BUFFER_ADDR	4	DD	Address of first receive buffer *
22	RCV_FRAME_COUNT	2	DW	Count of received frames chained to RCV_FRAME_ADDR *
24	RCV_FRAME_ADDR	4	DD	Address of received frames *
28	EVENT_ERROR_CODE	2	DW	Exception code *, **
30	EVENT_ERROR_DATA	6	DW	Exception parameters *, **
<b>For DLC Status Changes:</b> For DLC status change events posting, the fields starting at offset 10 are defined differently. The description for those fields starts here.				
10	STATION_ID	2	DW	SAP or link station ID *
12	DLC_STATUS_CODE	2	DW	DLC status code *
14	FRMR_DATA	5	DB	Frame data *
19	ACCESS_PRIORITY	1	DB	The new access priority *
20	REMOTE_NODE	6	DB	Remote node address *
26	REMOTE_SAP	1	DB	Remote SAP address *
27		1	DB	Reserved
28	USER_STAT_VALUE	2	DW	User status value *
* Indicates a returned value.				
** Indicates that these parameters may also be defined as an 8086 declaration type of DB.				

---

#### STATION\_ID

**Explanation:** The STATION\_ID to be posted for events.

---

#### OPTION\_INDICATOR

**Explanation:** Option indicator for matching READ command requests.

Indicator used to match events with the READ command for posting of events.

#### Indicator Meaning

**X'00'** Match READ command using STATION\_ID, X'nsss', where nn is the SAP number and ss is the STATION number. All events on the completion list associated with the STATION\_ID number will be checked for a READ command match.

**X'01'** Match READ command using SAP number of the STATION\_ID, X'nn00', where nn is the SAP number. All events on the completion list associated with the SAP number will be checked for a READ command match.

**X'02'** Match READ command using all events.

All events on the application program's completion list will be checked for a READ command match.

If the OPTION\_INDICATOR is not X'00' through X'02' then the command will terminate with CCB\_RETCODE set to X'06'.

## EVENT\_SET

**Explanation:** Set of events for notification.

There are 7 different types of asynchronous events that can be posted. A single READ command can request to be posted for more than one event. Thus, multiple bits may be set (non-zero). Each event is bit-mapped using the EVENT\_SET byte as described below.

- Bit 7 is reserved.
- Bit 6 is for System Action. (non-critical)

A non-critical system action occurs while using the System Key defined by the configuration parameters and one of the following is true.

- The DIR.READ.LOG command has been issued to read the adapter or Direct Interface logs.
- The DIR.SET.FUNCTIONAL.ADDRESS command has been issued to modify the functional address.
- The DIR.SET GROUP ADDRESS command has been issued to modify the group address.
- The DLC.RESET command has been issued to reset SAP stations.

In these cases the adapter is not closed or reinitialized and service can still be provided to application programs once the following occurs.

- The functional or group addresses have been reset.

This is only applicable if an application program was using one of the addresses and the system action resulted in the address not being suitable for the application program.

- The SAP STATION(s) have been re-opened.

This is only applicable if a SAP STATION used by an application program was reset by the system action.

All other non-critical system actions do not require the application program to take further action as service has not been disrupted.

- Bit 5 is Network Status (Non-Critical)

Notification of a Network Status change.

- Bit 4 is Critical Exception

A critical exception occurs when the adapter is abruptly closed or reinitialized. The adapter can be closed either by an unrecoverable failure or by the action of a system administrator issuing the DIR.CLOSE.ADAPTER command with a System Key defined by the configuration parameters. The adapter can also be reinitialized by the action of a system administrator issuing the DIR.INITIALIZE command with the System Key defined by the configuration parameters.

Individual sub-events that result in critical exceptions are,

Network Status (Adapter closed cases only)



## READ

Adapter Check  
PC Error  
System Action (DIR.INITIALIZE and DIR.CLOSE.ADAPTER commands)

These critical exception events are provide in the CRITICAL\_SUBSET field when a critical exception occurs.

**Warning:** If a critical exception occurs and an application program does not have a READ command outstanding, the only action taken by the adapter support software will be the setting of all outstanding command's return code to X'07'. In this case, **NO** pointers to buffers from SAP buffer pools or receive frames and **NO** pointer to a list of outstanding CCBs will be provided to the user.

- Bit 3 is DLC status change  
Notification of a DLC status change.
- Bit 2 is Receive data  
MAC (Token-Ring Network only) and non-MAC frames are received.
- Bit 1 is Transmit completion  
A DLC or DIR transmit command completes. The Transmit completion bit is used to expedite data transfer operations.
- Bit 0 is Command completion  
A DLC or DIR command completes.

If EVENT\_SET is set to X'00' (no event bit set), then the command will terminate with CCB\_RETCODE of X'05'.

---

## EVENT

**Explanation:** The posting event.

This field contains the event being posted. Bit 0 through bit 6 are bit mapped as described above. For each posting event a single bit will be set to indicate the particular event that has occurred. Each type of event that is posted will have different data placed into the CCB of the READ command. The following describes the additional CCB fields used for each event:

- Command Completion
  - CCB\_COUNT
  - EVENT\_CCB\_POINTER
  - BUFFER\_COUNT  
(DLC.CLOSE.SAP and DLC.RESET commands)
  - FIRST\_BUFFER\_ADDR  
(DLC.CLOSE.SAP and DLC.RESET commands)
  - RCV\_FRAME\_COUNT  
(DLC.CLOSE.SAP, DLC.CLOSE.STATION, DLC.RESET, and RECEIVE commands)
  - RCV\_FRAME\_ADDR  
(DLC.CLOSE.SAP, DLC.CLOSE.STATION, DLC.RESET, and RECEIVE commands)

- Transmit Completion
  - CCB\_COUNT
  - EVENT\_CCB\_POINTER
- Receive Data
  - RCV\_FRAME\_COUNT
  - RCV\_FRAME\_ADDR
- DLC Status Change
  - NOTIFICATION\_FLAG

Starting at offset 10 in the READ command's CCB parameter table, the DLC Status Table is defined below.

- STATION\_ID
- DLC.STATUS\_CODE
- FRMR\_DATA
- ACCESS\_PRIORITY
- REMOTE\_NODE
- REMOTE\_SAP
- USER\_STAT\_VALUE
- Critical Exception
  - CRITICAL\_SUBSET
  - NOTIFICATION\_FLAG
  - CCB\_COUNT
  - EVENT\_CCB\_POINTER
  - BUFFER\_COUNT
  - FIRST\_BUFFER\_ADDR
  - RCV\_FRAME\_COUNT
  - RCV\_FRAME\_ADDR
  - EVENT\_ERROR\_CODE
  - EVENT\_ERROR\_DATA
- Network Status (Non-Critical)
  - NOTIFICATION\_FLAG
  - EVENT\_ERROR\_CODE
- System Action (Non-Critical)
  - NOTIFICATION\_FLAG
  - CCB\_COUNT
  - EVENT\_CCB\_POINTER
  - BUFFER\_COUNT
  - FIRST\_BUFFER\_ADDR
  - RCV\_FRAME\_COUNT
  - RCV\_FRAME\_ADDR
  - EVENT\_ERROR\_CODE
  - EVENT\_ERROR\_DATA

---

**CRITICAL\_SUBSET**

**Explanation:** Exception event causing a critical exception.

When a critical exception event posts, this field will contain the actual event that resulted in the critical exception. The individual events are mapped as follows:

<b>CRITICAL_SUBSET</b>	<b>Meaning</b>
<b>X'01'</b>	Network Status
<b>X'02'</b>	Adapter Check
<b>X'03'</b>	PC System Error
<b>X'04'</b>	System Action

---

**NOTIFICATION\_FLAG**

**Explanation:** User notification flag for events.

For notification of exception conditions, the user exception flag defined with the DIR.SET.EXCEPTION.FLAGS command is returned.

For notification of a DLC Status change, this value will contain the DLC\_STATUS\_FLAG, as defined with the DLC.OPEN.SAP command, for the link station that has experienced the change.

These flags are preserved across invocations and may be used by the application program for user specific data.

---

**CCB\_COUNT**

**Explanation:** Count of CCBs chained to EVENT\_CCB\_POINTER.

The number of CCBs chained to the EVENT\_CCB\_POINTER field as described below.

---

**EVENT\_CCB\_POINTER**

**Explanation:** Pointer to Command Control Blocks

Points to a CCB when notifying an application program of a single completed command.

When notifying an application program of the DIR.CLOSE.ADAPTER, DIR.TIMER.CANCEL.GROUP, DLC.CLOSE.SAP and DLC.RESET commands completing, this field will point to the completing command with a list of outstanding commands chained together using the CCB\_POINTER of the commands.

If TRANSMIT commands are issued specifying that their CCBs be chained together upon completion, this field will point to the first member of a list of completed TRANSMIT command CCBs chained using the CCB\_POINTER of the commands.

If an exception occurs that causes the adapter to close (encounters an unrecoverable error) or causes SAP stations to be reset, all outstanding commands will be chained to the EVENT\_CCB\_POINTER field using the CCB\_POINTER of the commands. The CCB\_RETCODE of all commands chained to the EVENT\_CCB\_POINTER will be set as follows:

- If an Adapter Check, Network Status, or PC System Detected Error occurs, the CCB\_RETCODE will be X'07'.
- If System Action occurs, the CCB\_RETCODE will be X'62'.

**Note:** The adapter can be closed either by an unrecoverable failure or by the action of a system administrator issuing commands with a System Key. In addition the System Key can be used to

reset SAP stations. All Commands that make use of a System Key are clearly stated in the description of the commands.

### **BUFFER\_COUNT**

**Explanation:** Count of buffers chained to FIRST\_BUFFER\_ADDR.

The number of buffers chained to the FIRST\_BUFFER\_ADDR field is contained in this field.

### **FIRST\_BUFFER\_ADDR**

**Explanation:** Address of the first buffer in the buffer pools being returned.

If the adapter encounters an unrecoverable error, a System Action exception occurs, or the DLC.CLOSE.SAP or DLC.RESET commands have been issued that cause SAP or direct stations to be closed or reset, the following will occur:

- All buffers contained in associated SAP or direct station buffer pools will be chained to this field. This field will contain the address of the first buffer of the buffer pool with all other buffers chained.

**Note:** The adapter can be closed either by a catastrophic failure or by the action of a system administrator issuing commands with a System Key. In addition the System Key can be used to reset SAP stations. All Commands that make use of a System Key are clearly stated in the description of the commands.

### **RCV\_FRAME\_COUNT**

**Explanation:** Count of frames chained to RCV\_FRAME\_ADDR.

The number of frames chained to the RCV\_FRAME\_ADDR field is contained in this field.

### **RCV\_FRAME\_ADDR**

**Explanation:** Address of received frames.

When data is received and posted with the READ command requesting receive data, the RCV\_FRAME\_ADDR field will contain the address of the first buffer of a frame with all other frames chained using the first buffer of each frame.

If the adapter encounters an unrecoverable error, a System Action exception occurs, or DLC.CLOSE.SAP, DLC.CLOSE.STATION, and DLC.RESET commands have been issued that cause the STATION ID to be closed or reset, the following will occur:

- All received frames associated with the STATION ID(s) on the completion list will be removed and chained to this field.

**Note:** The adapter can be closed either by a catastrophic failure or by the action of a system administrator issuing commands with a System Key. In addition the System Key can be used to reset SAP stations. All Commands that make use of a System Key are clearly stated in the description of the commands.

### **EVENT\_ERROR\_CODE**

**Explanation:** Reason, Status or Error Codes

Codes used for identifying error conditions and status changes for exceptions are placed into the EVENT\_CODE field. See "Exception Indications" on page B-40 for all exception conditions.

## READ

---

### EVENT\_ERROR\_DATA

**Explanation:** Exception Parameters

The Adapter Check exception and a PC System Detected Error exception have 3 parameters defined for passing maintenance and RAS data. These parameters are placed into this field. See "Exception Indications" on page B-40 for all exception conditions.

### DLC Status Change Events

For DLC status change events posting, the fields starting at offset 10 are defined differently. The description for those fields are as follows.

---

### STATION\_ID

**Explanation:** Station ID for DLC status change.

---

### DLC.STATUS.CODE

**Explanation:** The DLC Status Code.

**Note:** See Appendix B, "Return Codes" for the DLC Status Codes.

Multiple bits may be set when a DLC Status is posted.

---

### FRMR\_DATA

**Explanation:** Frame data

Five bytes of reason code that are applicable when an FRMR is either transmitted or received. See the *IBM Token-Ring Network Architecture Reference* for a complete description of this information.

---

### ACCESS\_PRIORITY

**Explanation:** The new access priority.

The new access priority that is applicable when status bit 5 is on. the format is B'nnn00000' where 'nnn' is the access priority. This byte is not used and is set to zero when using PC Network adapters.

---

### REMOTE\_NODE

**Explanation:** Remote six-byte node address.

The six-byte node address of the remote partner for a newly opened link station. Applicable when status bit 10 is on.

---

### REMOTE\_SAP

**Explanation:** Remote SAP address.

The one-byte remote SAP value for a newly opened link station. Applicable when status bit 10 is on.

---

### USER\_STAT\_VALUE

**Explanation:** User status value.

User value as defined in the DLC.OPEN.SAP command.

**READ.CANCEL**

Hex 32

READ.CANCEL

**Command Description:****This command is for CCB2 only:**

This command cancels an outstanding READ command for an individual application program.

**Command Specifics:** This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software. The cancelled READ CCB (if one is outstanding) will be terminated with the CCB\_RETCODE set to X'0A'. The READ command will not be posted using the post semaphore. The post semaphore for the READ.CANCEL command will be cleared to signal the completion of the READ.CANCEL and the READ command will not be posted other than to set the return code.

The fields CCB\_PARM\_OFFSET and CCB\_PARAMETER\_1 are combined to form a four byte (8086 declaration type DD) address of the READ CCB to be cancelled.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

**RECEIVE**

Hex 28

RECEIVE

**Command Description:** This command receives data for the station defined in the STATION\_ID field of the CCB.

**Command Specifics:** When a receive command is issued, it is queued in the PC system, awaiting received data for the specified station. Multiple receive commands may be active at one time, but only one for each specific STATION\_ID. Once data is received for an outstanding receive command and there are adequate receive buffers available in the pool, the following takes place:

**For CCB1:**

1. The adapter support software will fill receive buffers from the appropriate buffer pool and place the address of the first buffer in the receive command parameter table.
2. If the optional RECEIVED\_DATA user appendage is defined and the return code is X'00', the following happens:
  - a. The address of the CCB is placed in registers DS and SI.
  - b. The address of the first receive data buffer is placed in registers ES and BX.
  - c. The RECEIVE\_DATA user appendage exit is taken.

The receive command will remain active to receive any data that follows. Therefore, once a receive command is issued for a station, it will continue to be active until terminated by an exception condition or, until a RECEIVE.CANCEL is issued for the RECEIVE or until the SAP or link station is closed.

## RECEIVE

If the optional RECEIVED\_DATA user appendage is not defined or the return code is not X'00', the command will be completed in the same way as any other command. The CCB\_CMD\_CMPL user exit will be taken with the CCB address in registers ES and BX.

When a DLC.CLOSE.SAP or DLC.CLOSE.STATION is issued, the receive command associated with that SAP or station will be terminated with a CCB\_RETCODE of X'0A', and the address of the RECEIVE command CCB will be placed in the CCB\_POINTER field of the command causing the SAP or station to close. The RECEIVE command's completion appendage is not taken.

When the RECEIVE command has been issued with RECEIVE\_DATA set, and when data has been received successfully, this command never actually completes with a return code of X'00'. However, when data is received successfully, there is an implied return code of X'00'. The actual return code will remain X'FF'. A return code is set only when the command terminates, such as when a lost data condition occurs or if the RECEIVE\_DATA was not set when the command was issued.

### For CCB2:

1. The adapter support software will fill receive buffers from the appropriate buffer pool and place the address of the first buffer in the receive command parameter table.
2. If the optional RECEIVED\_FLAG is set and the return code is X'00', the following happens:
  - a. If a READ command defined for notification of receive data is outstanding, the addresses will be copied into the READ command's parameter table and the READ semaphore will be cleared.
  - b. If no READ command defined for notification of receive data is outstanding when the receive data event occurs, the address of the first receive buffer is placed onto a completion list. Upon reception of a READ command the completion list will be scanned. If the RCV\_READ\_OPTION field of the RECEIVE command is set for chaining, all data received for the STATION\_ID will be chained and posted at once. The first receive buffer is copied into the READ command's parameter table and the READ semaphore will be cleared.
  - c. If the optional RECEIVED\_FLAG is not set or the return code is not X'00', the command will be completed in the same way as any other command. The CCB\_CMD\_FLAG will be used to determine how the command completion notification should be handled.

The receive command will remain active to receive any data that follows. Therefore, once a receive command is issued for a station, it will continue to be active until terminated by an exception condition or, until a RECEIVE.CANCEL is issued for the RECEIVE or until the SAP or link station is closed.

When a DLC.CLOSE.SAP or DLC.CLOSE.STATION is issued, the receive command associated with that SAP or station will be terminated with a CCB\_RETCODE of X'0A' and the address of the RECEIVE command CCB will be placed in the CCB\_POINTER field of the command causing the SAP or station to close. There will be no completion of any READ commands that may be outstanding requesting receive data posting, or clearing of semaphores defined by the RECEIVE command.

When the RECEIVE command has been issued with RECEIVE\_FLAG set, and data has been received successfully, this command never actually completes with a return code of X'00'. However, when data is received successfully, there is an implied return code of X'00'. The actual return code will remain X'FF' (command in process). A return code is set only when the command terminates, such as when a lost data condition occurs or if the RECEIVE\_FLAG was not set when the command was issued.

**For CCB3:**

1. The adapter support software will fill receive buffers from the appropriate buffer pool and place the address of the first buffer in the receive command parameter table.
2. If the optional RCV\_DATA\_APPNDG user appendage is defined and the return code is X'00', the adapter support software will call the application program's device driver with the following parameters set:
  - Register DI contains the RCV\_DATA\_APPNDG as defined by the RECEIVE command.
  - An invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software the application program must remove the invocation code from the stack.
  - Register DS contains the application program device driver's protect mode data segment selector.
  - Register CX contains the adapter number.
  - Registers ES and BX contain a virtual address to the first SAP buffer of the receive information.
  - Registers AX and SI contain a virtual address of the RECEIVE command's CCB for which receive data has been processed.

The receive command will remain active to receive any data that follows. Therefore, once a receive command is issued for a station, it will continue to be active until terminated by an exception condition or, until a RECEIVE.CANCEL is issued for the RECEIVE or until the SAP or link station is closed.

When a DLC.CLOSE.SAP or DLC.CLOSE.STATION is issued, the receive command associated with that SAP or station will be terminated with a CCB\_RETCODE of X'0A' and the address of the RECEIVE command CCB will be placed in the CCB\_POINTER field of the command causing the SAP or station to close. The RECEIVE command's completion appendage is not taken.

When the RECEIVE command has been issued with RCV\_DATA\_APPNDG set, and data has been received successfully, this command never actually completes with a return code of X'00'. However, when data is received successfully, there is an implied return code of X'00'. The actual return code will remain X'FF' (command in process). A return code is set only when the command terminates, such as when a lost data condition occurs or if the RCV\_DATA\_APPNDG was not set when the command was issued.

**Valid Return Codes: see "CCB Return Codes Listed by Command" on page B-4.**



# RECEIVE

## CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	Defines the station receiving data
2	USER_LENGTH	2	DW	Length of user data in buffers
<b>For CCB1:</b>				
4	RECEIVED_DATA	4	DD	Optional user exit for received data
<b>For CCB2:</b>				
4	RECEIVE_FLAG	4	DD	Optional receive user flag
<b>For CCB3:</b>				
4	RCV_DATA_APPNDG	2	DW	Offset to receive data appendage
6		2	DW	Reserved for the application program
<b>For all CCBs:</b>				
8	FIRST_BUFFER	4	DD	First receive buffer address from the adapter *
12	OPTIONS	1	DB	RECEIVE options
<b>For CCB2:</b>				
13		3	DB	Reserved
16	RCV_READ_OPTION	1	DB	Read posting option
<b>For CCB3:</b>				
13		3	DB	Reserved
16		1	DB	Reserved for the application program
* Indicates a returned value.				

## STATION\_ID

**Explanation:** Defines the station and the kind of data the station will receive. The STATION\_ID is defined in "SAPs, Stations, and IDs" on page 2-24. It identifies the station to receive data as follows:

- X'0000'** Direct station, receive both MAC and non-MAC frames. Direct station receives only non-MAC on the PC Network.
- X'0001'** Direct station, receive MAC frames. Not used on the PC Network. Reserved for the Token-Ring Network.
- X'0002'** Direct station, receive non-MAC frames
- X'nn00'** SAP, receive data for SAP 'nn'
- X'nsss'** Link station, receive data for SAP 'nn', station 'ss'

Every station that is defined to the adapter may have a RECEIVE command outstanding, but there may be only one RECEIVE command for any specific station.

**Note:** If no receive command is active for link station X'nsss', the frame received by the adapter for the link station will be received by SAP X'nn00' if it has a receive command active.

## USER\_LENGTH

**Explanation:** This field specifies the length of a user space in the buffer for private data. The data placed in the receive buffer starts at an offset specified by the USER\_OFFSET field of the receive buffer. The information placed in the user space is not altered by the adapter support software nor is it overwritten by the received frame data. See "Buffer Pools" on page 2-35 for the receive buffer.

---

**RECEIVED\_DATA**

**Explanation:** The address of a user-provided appendage routine that will be taken when data is received. By coding this parameter, the application program may receive data and keep the same receive command active to receive subsequent data.

---

**RECEIVE\_FLAG**

**Explanation:** This is a user flag that specifies whether or not received data should be posted using the READ command specifying notification for receive data. By setting this flag to a non-zero value, the application program may receive data and keep the same receive command outstanding to receive subsequent data.

---

**RCV\_DATA\_APPNDG**

**Explanation:** This is the offset of the receive data appendage where received data is posted to the application program. This address offset is passed to the application program's device driver intercommunication entry point in register DI when the adapter support software calls the application program's device driver.

---

**FIRST\_BUFFER**

**Explanation:**

**For CCB1 and CCB3:** A returned value indicating the address of the first buffer. This is the same address value that is placed in registers ES and BX. If the address is X'00000000', there is no receive data.

**For CCB2:** A returned value. If no RECEIVE\_FLAG is specified, the address of the first receive buffer is placed into this field. This is the same value placed in the READ command's CCB parameter table when a READ command requesting notification of receive data completes. If the address is X'00000000', there is no receive data.

---

**OPTIONS**

**Explanation:** Options set by the application program to inform the adapter support software how to present received information to the application program.

- BIT 7 = CONTIGUOUS MAC

Not used on PC Network.

Applicable only if the received frame is a MAC frame.

If this bit is on, the entire frame is placed into the buffers as a continuous data string after the USER\_SPACE.

If this bit is off, the 32-byte LAN header is removed from the frame and is placed in a special location in buffer one.

## RECEIVE

- BIT 6 = CONTIGUOUS DATA

Applicable only if the received frame is a non-MAC frame.

If this bit is on, the entire frame is placed into the buffers as a continuous data string.

If this bit is off, the 32-byte LAN HEADER is placed in buffer one followed by the DLC HEADER, and all the received data.

- BIT 5 = BREAK

If this bit is on, the first received data is placed in the second receive buffer. The first buffer contains only the buffer header data.

- BITS 4-0 = Reserved; should be zero, but is not checked.

---

## RCV\_READ\_OPTION

**Explanation:** This field is only used when the RECEIVE\_FLAG is set to a non-zero value.

Frames that have been received are destined for SAP stations, link stations, or one of the direct stations. If a RECEIVE command has been issued for a SAP station but not for any of the link stations opened under the SAP, then data received for the link station will be received using the SAP's RECEIVE command. In this case it is possible for the application program to issue a READ command requesting receive data for a link station. If RECEIVE commands have been issued for all link stations opened under a SAP, it is also possible for a READ command to be issued requesting receive data for a SAP and its link stations. To prevent the application program from having to issue a READ command for each frame received, received frames can be chained together. To do this the application program must specify ahead of time how received frames are to be chained. The RCV\_READ\_OPTION field should be set as described below to allow these chained frames.

- If the RCV\_READ\_OPTION field contains a X'00', received frames will not be chained.

This option specifies that received frames for the station are to be placed separately onto the adapter support software's completion list. A READ command will have to be issued to retrieve each frame from the completion list.

- If the RCV\_READ\_OPTION field contains a X'01', all frames received for a link station will be chained.

This option specifies that received frames for the station are to be chained onto the adapter support software's completion list for the specified station ID. A single READ command can be issued to retrieve all chained frames from the completion list for this station ID. If the RECEIVE command is for a SAP station this option has the same affect as a RCV\_READ\_OPTION field containing X'02' for frames received for the SAP STATION.

- If the RCV\_READ\_OPTION field contains X'02', all frames received for a SAP will be chained.

This option specifies that received frames for the station ID are to be chained onto the adapter support software's completion list. All frames received for a SAP and its link stations are chained together. A single READ command can be issued to retrieve all chained frames from the completion list for this link station.

If RECEIVE commands have been issued for link stations, the options specified with the link station's RECEIVE commands will be used to determine how receive data is chained. If a link station has a RECEIVE command outstanding specifying that its received frames should be chained on a link station basis, its frames would not be chained together with the SAP's frames. All link stations that do not have a RECEIVE command outstanding will have their received frames chained as specified by the SAP's RECEIVE command. For example, if a link station did not have a RECEIVE command outstanding, but the SAP did have one specifying that no received frames be chained, all frames received for the link station and the SAP would be placed onto the adapter support software's completion list individually. If neither the SAP nor the link station had a RECEIVE command outstanding, the direct station would be used. See "SAPs, Stations, and IDs" on page 2-24.

**Notes:**

1. The RCV\_READ\_OPTION for the SAP and the link station are independent of each other. The option specified for a link station's RECEIVE command will be considered first when determining if received frames are to be chained. A RECEIVE command does not have to be outstanding for a SAP in order to have received frames for link stations chained on a SAP basis. This of course requires that link stations have RECEIVE commands outstanding with RCV\_READ\_OPTION set to X'02'.
2. If the value of the RCV\_READ\_OPTION field is not within the range of X'00' to X'02', the command will terminate with CCB\_RETCODE of X'06'.
3. The RCV\_READ\_OPTION of a SAP's RECEIVE command will be used to determine how the SAP's receive data is chained.
4. The RCV\_READ\_OPTION of a link station's RECEIVE command will be used to determine how the link station's receive data is chained.
5. The RCV\_READ\_OPTION of a SAP's RECEIVE command will be used to determine how link station receive data for the SAP is chained when its link stations do not have RECEIVE commands outstanding.
6. A single RECEIVE command for a SAP (RCV\_READ\_OPTION = X'00') can be used to receive all data for a SAP and its link stations with each received frame being placed separately on to the completion list.
7. A single RECEIVE command for a SAP (RCV\_READ\_OPTION = X'01') can be used to chain all receive data for a SAP and its link stations on individual station ID queues using a separate completion list entry for each station ID.
8. A single RECEIVE command for a SAP (RCV\_READ\_OPTION = X'02') can be used to chain all receive data for a SAP and its link station on a single SAP queue using one completion list entry.
9. An overall summary, READ commands need to match RECEIVE commands' RCV\_READ\_OPTION settings.

For information on receive buffers, see "Buffer Pools" on page 2-35.

**RECEIVE.CANCEL**

Hex 29

RECEIVE.CANCEL

**Command Description:** This command cancels a receive command on any specific SAP or link station including the direct station.

**Command Specifics:** The STATION\_ID specifies the SAP or station of a pending receive that is to be canceled. This command is executed entirely in the PC system. The return code is available to the application program upon return from the adapter support software.

The canceled RECEIVE CCB, if there is one, is terminated with a CCB\_RETCODE value of X'0A', command canceled by user request.

**For CCB1 and CCB3:** The RECEIVE command's completion appendage will not be taken. However, the RECEIVE.CANCEL command's completion appendage will be taken if provided.

**For CCB2:** If a READ command is outstanding requesting notification of receive data, the cancelation of a receive will not affect the READ. The READ command will not complete and the READ command is not posted except for the setting of the return code. The RECEIVE.CANCEL commands will post as defined by the CCB\_CMPL\_FLAG and CCB\_SEMAPHORE parameters.

## RECEIVE.MODIFY

After completion of the RECEIVE.CANCEL, the CCB\_POINTER will contain the address of the RECEIVE command canceled.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

**For CCB1:** The station ID is located at the CCB\_PARM\_TAB field of the control block.

**For CCB1, CCB2, and CCB3:** The station ID is located at the CCB\_PARM\_OFFSET field of the control block.

## RECEIVE.MODIFY

Hex 2A

RECEIVE.MODIFY

### Command Description:

**This command is for CCB1 and CCB3 only:**

This command receives data and puts some of the data into a buffer not taken from the SAP buffer pool.

**Command Specifics:** This command operates the same as the RECEIVE command, with the following exceptions:

- There are no receive options in the parameter table.
- Only data (non-MAC) frames may be received.
- Data is received into one SAP buffer and, additionally, one user buffer.
- The format of received data (that is, the data following the DLC header) is assumed to be:

1111hh...hhdd...dd

where:

- 1111 is a 2-byte field whose value is its own length (2 bytes) plus the length in bytes of the hh...hh field. The length field (1111) will have a format defined as DW.
- hh...hh is a message header.
- dd...dd is message data.
- When data is received:
  - A SAP buffer is obtained by the adapter support software.

#### For CCB1:

- The first 58 bytes of the SAP buffer are prepared exactly as when executing a RECEIVE command with the option not continuous data.
- At byte 58 (plus user length, if applicable), the received 1111hh...hh is placed into the SAP buffer. (If the data [1111hh...hhdd...dd] exceeds the length of the buffer, the frame is discarded by the adapter support software and no indication is given to the application program.)
- The adapter support software calls a subroutine defined in the SUBROUTINE@ parameter to obtain the length and location of an application program buffer.

An option is to call the RECEIVE\_DATA appendage once data has been placed into the application program's buffer.

#### For CCB3:

- The first 62 bytes of the SAP buffer are prepared exactly as when executing a RECEIVE command with the option not continuous data.
- At byte 62 (plus user length, if applicable), the received 1111hh...hh is placed into the SAP buffer. (If the data [1111hh...hhdd...dd] exceeds the length of the buffer, the

frame is discarded by the adapter support software and no indication is given to the application program.)

- The adapter support software calls the application program's device driver intercommunication entry point, passing the offset of a subroutine address (SUBROUTINE@) in the DI register. This call is made by the adapter support software to obtain the length and location of an application program buffer.

An option is to call the application program's device driver intercommunication entry point with the RCV\_DATA\_APPNDG parameter passed in the DI register, once data has been placed into the application program's buffer.

**Valid Return Codes:** see "CCB Return Codes Listed by Command" on page B-4.

#### CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	Defines the station receiving data
2	USER_LENGTH	2	DW	Length of user data in buffers
<b>For: CCB1:</b>				
4	RECEIVED_DATA	4	DD	Optional user exit for received data
<b>For: CCB3:</b>				
4	RCV_DATA_APPNDG	2	DW	Received data appendage offset
6		2	DW	Reserved for the application program
<b>For CCB1 and CCB3:</b>				
8	FIRST_BUFFER	4	DD	First receive buffer address *
<b>For CCB1:</b>				
12	SUBROUTINE@	4	DD	The address of a subroutine
<b>For CCB3:</b>				
12	SUBROUTINE@	2	DW	Offset to the subroutine
14		2	DW	Reserved for the application program
* Indicates a returned value.				
@ Indicates an address throughout this document.				

#### STATION\_ID

**Explanation:** Defines the station and the kind of data the station will receive. The STATION\_ID is defined in "SAPs, Stations, and IDs" on page 2-24. It identifies the station to receive data as follows:

- X'0000'** Direct station, receive non-MAC frames only.
- X'0002'** Direct station, receive non-MAC frames.
- X'nn00'** SAP, receive data for SAP 'nn'.
- X'nnss'** Link station, receive data for SAP 'nn', station 'ss'.

Every station that is defined for this adapter may have a RECEIVE or RECEIVE.MODIFY command outstanding, but there may be only one RECEIVE or RECEIVE.MODIFY command for any specific station.

## RECEIVE.MODIFY

---

### USER\_LENGTH

**Explanation:** This field specifies the length of a user space in the buffer for private data. The data placed in the receive buffer starts at an offset specified by the USER\_OFFSET field of the receive buffer. The information placed in the user space is not altered by the adapter support software or the received frame data. See "Buffer Pools" on page 2-35 for the receive buffer.

---

### RECEIVE\_DATA

**Explanation:**

This is the address of an appendage routine provided by the application program that will be taken to receive data. By coding this parameter the application program may receive data and keep the same RECEIVE command active to receive subsequent data.

When the adapter support software has updated the application program buffer obtained from the SUBROUTINE@ call, the receive appendage will be called to post the reception of data. See "RECEIVE" on page 3-77 for register usage when the receive appendage is called to post the reception of data.

When the application program's receive appendage is called to post the reception of data, Bytes 6 and 7 (the LENGTH\_IN\_BUFFER field) of the SAP buffer referenced by ES and BX will be set to the length of the data moved into the application program's receive buffer starting at offset 30 (SOURCE\_ADDRESS). If the received data is more than what will fit in the application program's receive buffer then bytes 6 and 7 will be set to X'FFFF' and any excess data is lost.

Upon return, the SAP buffer is returned to the available pool.

---

### RCV\_DATA\_APPNDG

**Explanation:** This is an address offset of the receive data appendage where receive modify data information is posted to the application program.

When the adapter support software has updated the user buffer obtained from the SUBROUTINE@ call with the received data, the application program's device driver will be called with this receive appendage offset passed in register DI. The adapter support software will enter the application program device driver's intercommunication entry point with a Call Far instruction and the application program's device driver must return with a Return Far instruction. This call will be made using the application program's device driver entry point obtained when the DIR.OPEN.ADAPTER command is issued. See "RECEIVE" on page 3-77 for register usage when the application program is called to post the reception of data.

When the application program device driver's receive data appendage is called to post the reception of data, bytes 6-7 (LENGTH\_IN\_BUFFER field) of the SAP buffer referenced by ES and BX will be set to the length of the data moved into the user's receive buffer starting at offset 34 (SOURCE ADDRESS). If the received data is more than what will fit in the user's receive buffer then the bytes 6-7 will be set to X'FFFF' and any excess data is lost.

Upon return, the SAP buffer is returned to the available pool.

---

### FIRST\_BUFFER

**Explanation:** A returned value indicating the address of the first buffer. This is the same address value that is placed in registers ES and BX. If the address is X'00000000', there is no receive data.

---

**SUBROUTINE@****Explanation:**

**For CCB1:** The address of a subroutine or appendage that the adapter support software calls to obtain the address and length of an additional user buffer. The adapter support software will enter the subroutine with a Call Far instruction and the subroutine must return with a Return Far instruction. This field must be provided. When the SUBROUTINE@ is entered the following parameters are set:

- Registers ES and BX point to the SAP buffer.
- Registers ES and DI point to offset 30 of the SAP buffer (the source address of the frame).
- Registers AX and SI point to this adapter's node address in shared RAM.
- Register CX contains the adapter number.
- Register DX contains the number of bytes left in the frame.

When the appendage subroutine is completed, it must set the AL register and issue a far return instruction.

- If the AL register is set to zero then:
  - Registers ES and DI point to a receive buffer
  - Register CX indicates the length of the receive buffer.
- If the AL register is not set to zero then:
  - The adapter support software returns the SAP buffer to the pool
  - The received data is discarded
  - The received data appendage is not taken.

If the frame is an I frame, it is treated by the adapter's DLC logic as if it had been successfully received.

**For CCB3:** This is the address offset of the application program's subroutine that the adapter support software will call to obtain a user's buffer length and address. The adapter support software will enter the application program's device driver with a Call Far instruction and the application program's device driver must return with a Return Far instruction.

When data is received, the application program's device driver will be called to obtain a buffer. This call will be made using the application program's device driver entry point obtained when the DIR.OPEN.ADAPTER command is issued. The following will be set for this call.

- Register DI contains the offset of a subroutine within the application program's device driver code segment.
- Register DS contains the called device driver's protect mode data segment selector.
- Registers ES and BX contain a virtual address to a SAP buffer.
- Registers AX and SI contain a virtual address to this adapter's node address.
- Register CX contains the adapter number.
- Register DX contains the number of bytes of the frame that have not been removed from the adapter's receive buffer(s).

Before returning from the SUBROUTINE@ call, the application program's device driver must set the AL register to indicate to the adapter support software what action is to be taken:

- If the AL register is set to zero then:
  - Registers DX and BX contain a 32-bit physical address of a user receive buffer, where DX contains the most significant word of the address.
  - Register CX contains the length of the user receive buffer.
- If AL register is set to non-zero then:
  - The adapter support software returns the SAP buffer to the available pool.
  - The received frame is discarded.



## TRANSMIT

If the discarded frame is an I-frame, the frame is treated by the adapter's DLC logic as if it had been successfully received.

See "Buffer Pools" on page 2-35.

## TRANSMIT.DIR.FRAME

Hex 0A

### TRANSMIT.DIR.FRAME

**Command Description:** This command transmits data for the direct station.

**Command Specifics:** This command may be used only for the direct stations. It is invalid for all STATION\_IDs except direct stations.

The entire transmission frame must be prepared by the application program, including the LAN header and any required data headers.

The LAN header in the user's buffer must reserve the space for the adapter to insert the source address. The user's buffer is not altered. The adapter sets the source address into the transmit buffer in shared RAM. However, the adapter support software passes the high-order bit of the Source Address, used to indicate the presence of routing information, as supplied by the application program to the adapter. The high-order bit is loaded, as it was supplied, into shared RAM. The adapter verifies that the access priority and source class (MAC frames only) are valid. MAC frames do not exist on the PC Network. See Figures 2-4, 2-5, and 2-6 starting on page 2-42 for the configuration of the header.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

## TRANSMIT.I.FRAME

Hex 0B

### TRANSMIT.I.FRAME

**Command Description:** This command transmits "information" data for a link station.

**Command Specifics:** This command may be used only for a link station. The only data supplied by the application program is the actual data portion of the message. The LAN and DLC headers are not built by the application program. The DLC code handles all transmission retries. The maximum size of the user data provided with this command is limited to the DHB size minus six bytes.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

### Transmit Completion

Under normal conditions, this command will terminate when verification of its receipt has been received from the receiving link station. Since link stations are controlled by the DLC functions, DLC information is exchanged to verify delivery at the link level in addition to the verification at the physical level.

### For CCB1:

1. If the return code is X'00' and the link station's MAXOUT parameter is not a value of 1, multiple TRANSMIT.I.FRAME commands for a given link station may be completed at the same time. All frames acknowledged by a specific received acknowledgment are completed at the same time for a given link station. The completed TRANSMIT.I.FRAME commands for a given link station will be queued in the order in which they were issued,

each one pointing to the next by the CCB\_POINTER field until the last CCB which has zeros in the field. The adapter will take the appendage exit of the first CCB in the queue.

2. In a case where transmit commands are queued and an abortive condition occurs, all outstanding TRANSMIT.I.FRAME commands for a given link station will be queued and ended with appropriate return codes, as described in the previous step.

If the return code is X'28' then the link station will enter the disconnected mode. Once in the disconnected mode the application program will have to re-establish the connection before continuing transmission. See "Link Station States" on page 2-29.

**For CCB2:** If the application program has specified that each TRANSMIT.I.FRAME's CCB be placed onto the adapter support software's internal completions list (setting CCB\_CMPL\_FLAG) then the following will occur.

1. If a READ command is outstanding, or when a READ command is issued requesting notification of completed TRANSMIT commands or command completions, then the READ command will post. The READ command's CCB parameter table will contain a pointer to a single TRANSMIT.I.FRAME CCB or a chain of TRANSMIT.I.FRAME CCBs (if chaining is specified in the transmit commands).
2. In a case where transmit commands are chained and an abortive condition occurs (for example, link lost), all outstanding TRANSMIT.I.FRAME commands will be chained and completed, as previously explained.

If the return code is X'28', the link station will enter the disconnected mode. Once in the disconnected mode, the application program will have to re-establish the connection before continuing transmission. See "Link Station States" on page 2-29.

## TRANSMIT.TEST.CMD

Hex 11

TRANSMIT.TEST.CMD

**Command Description:** This command requests the adapter to transmit a Test command frame with the poll bit set.

**Command Specifics:** This command may be used only for a SAP. The adapter provides the DLC header. The application program must provide the LAN header and the optional test information. The first buffer must contain only the LAN header.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

## TRANSMIT

### TRANSMIT.UI.FRAME

Hex 0D

#### TRANSMIT.UI.FRAME

**Command Description:** This command transmits unnumbered information data for a SAP.

**Command Specifics:** This command may be used only by a SAP. The adapter provides the DLC header information. The application program must provide the LAN header and data portions of the message. The first buffer must contain only the LAN header.

Verification that the destination adapter or an intervening bridge adapter received the message may be checked from the FS field when the frame is removed by the transmitting adapter. However, no further receive verification takes place. For more information on the FS field see the following table and its field descriptions or "Frame Status" on page B-39.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

### TRANSMIT.XID.CMD

Hex 0E

#### TRANSMIT.XID.CMD

**Command Description:** This command transmits an XID command with the poll bit set on.

**Command Specifics:** This command may be used only by a SAP. The user provides the LAN header and data field. The user must also leave room for the DLC header that is provided by the adapter support software. If the SAP option indicates that the adapter handles XID commands, the adapter will provide the data. The first buffer must contain only the LAN header.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

### TRANSMIT.XID.RESP.FINAL

Hex 0F

#### TRANSMIT.XID.RESP.FINAL

**Command Description:** This command transmits an XID response with the final bit on.

**Command Specifics:** This command may be used only by a SAP opened with the SAP option specifying that the application program will handle XID commands. The user provides the LAN header and data field. The user must also leave room for the DLC header that is provided by the adapter support software. The first buffer must contain only the LAN header.

Refer to "Transmit Command Specifics" on page 3-91 for information common to all transmit commands.

**TRANSMIT.XID.RESP.NOT.FINAL**

Hex 10

**TRANSMIT.XID.RESP.NOT.FINAL**

**Command Description:** This command transmits an XID response with the final bit off.

**Command Specifics:** This command may be used only by a SAP opened with the SAP option specifying the application program will handle XID commands. The user provides the LAN header and data field. The user must also leave room for the DLC header that is provided by the adapter support software. The first buffer must contain only the LAN header.

Refer to “Transmit Command Specifics” on page 3-91 for information common to all transmit commands.

**Transmit Command Specifics**

The seven transmit commands are variations of the same basic transmit command. The command completion, parameter table and field explanations, and return codes are explained here. All differences are noted with the specific command description.

**Command Completion:** The transmit command will terminate when the frame has been transmitted and read back in by the adapter.

If the FS field is X'CC' (both “address recognized” and both “frame copied” bits on) and there are no other error conditions, the CCB\_RETCODE will be X'00' for all TRANSMIT commands. If the FS field is anything other than X'CC', the command will terminate with a CCB\_RETCODE of X'22'. Except for the TRANSMIT.I.FRAME command, if the FS field is not X'CC' the adapter support software and adapter will retransmit the frame based on its MAX\_RETRY\_CNT. For more information on the FS field see the following table and its field descriptions or “Frame Status” on page B-39.

**For CCB2:** To reduce the number of READ commands that must be issued in order to receive notification of completed transmit commands, the application program can specify that completed TRANSMIT CCBs be chained together. Once TRANSMIT CCBs are chained, a single READ command may be issued to retrieve all completed TRANSMIT CCBs at a given time.

**Valid Return Codes:** see “CCB Return Codes Listed by Command” on page B-4.

# TRANSMIT

## CCB Parameter Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	Defines the station sending data
2	TRANSMIT_FS	1	DB	Stripped FS field *
3	RSAP	1	DB	Remote SAP value
4	XMIT_QUEUE_ONE	4	DD	Address of the first transmit queue
8	XMIT_QUEUE_TWO	4	DD	Address of the second transmit queue
12	BUFFER_LEN_ONE	2	DW	Length of transmit buffer BUFFER_ONE
14	BUFFER_LEN_TWO	2	DW	Length of transmit buffer BUFFER_TWO
16	BUFFER_ONE	4	DD	The address of the first transmit buffer
20	BUFFER_TWO	4	DD	The address of the second transmit buffer
<b>For CCB2:</b>				
24	XMIT_READ_OPTION	1	DB	Read posting option
<b>For CCB3:</b>				
24		1	DB	Reserved for the application program
* Indicates a returned value.				

---

### STATION\_ID

**Explanation:** Defines what station is sending the data. The STATION\_ID is explained in "SAPs, Stations, and IDs" on page 2-24. It identifies the station that is to transmit data as follows:

- X'00nn'** Direct station, transmit both MAC and non-MAC frames (TRANSMIT.DIR.FRAME only). MAC frames do not exist on the PC Network.
- X'nn00'** SAP, transmit data for SAP 'nn' (Any transmit except TRANSMIT.DIR.FRAME and TRANSMIT.I.FRAME)
- X'nns'** Link station, transmit data for SAP 'nn', station 'ss' (Transmit.I.FRAME only).

**Note:** A station may have more than one TRANSMIT command outstanding at one time. The adapter support software will queue TRANSMIT commands.

---

### TRANSMIT\_FS

**Explanation:** This is the FS field as returned by the adapter or the adapter support software. It is a copy of the FS field after the frame has gone around the ring and has been read back by the ring interface of the adapter. See "Frame Status" on page B-39.

On the PC Network, this field is set to X'00' if the adapter detects an error on transmission, and the command will terminate with a return code of X'22'. If the adapter does not detect an error, the field is set to X'CC' and the command will terminate with a return code of X'00'.

The TRANSMIT\_FS field can be interrogated for the reason for transmission failure if the command terminates with CCB\_RETCODE of X'22'.

This field is not returned when the transmit command is for a link station, because all transmission retry is handled by the link facilities. The FS field is explained in the *IBM Token-Ring Network Architecture Reference*.

---

**RSAP**

**Explanation:** The SAP\_VALUE of the remote SAP that the sending (local) SAP is communicating with. This value is ignored if the sending station is a link station or a direct station.

---

**XMIT\_QUEUE\_ONE**

**Explanation:** The address of the first (or only) buffer in a queue of buffers to be transmitted. The data in all the buffers will be transmitted as one frame.

The buffers in this queue will not be returned to the SAP buffer pool upon command completion.

This transmit queue of buffers will not be used if the value is X'00000000'.

If the NEXT BUF POINTER field of the first buffer is not zero, there are additional buffers in this XMIT\_QUEUE\_ONE queue.

See "Transmit Buffers" on page 2-40 for details of transmit queues and buffers.

---

**XMIT\_QUEUE\_TWO**

**Explanation:** The address of the second queue of buffers to be transmitted.

If there are buffers in XMIT\_QUEUE\_ONE, the data in XMIT\_QUEUE\_TWO buffers will be transmitted following the data in XMIT\_QUEUE\_ONE buffers as one frame.

Prior to taking an appendage exit or posting completion, the buffers in this queue will be returned to the SAP buffer pool and this field set to zero upon command completion if the return code equals zero (X'00').

This transmit queue of buffers will not be used if the value is X'00000000'.

If the NEXT BUF POINTER field of the buffer is not zero, there are additional buffers in this XMIT\_QUEUE\_TWO queue.

See "Transmit Buffers" on page 2-40 for details of transmit queues and buffers.

---

**BUFFER\_LEN\_ONE**

**Explanation:** The length of the transmit buffer, containing the data to be transmitted, located by the contents of the BUFFER\_ONE field.

If this field is 0, all the following fields are ignored.

---

**BUFFER\_LEN\_TWO**

**Explanation:** The length of the transmit buffer, containing the data to be transmitted, located by the contents of the BUFFER\_TWO field.

If this field is 0, buffer two is not used.

---

**BUFFER\_ONE**

**Explanation:** The address of the buffer containing data to be transmitted. The data is transmitted following the data in XMIT\_QUEUE\_ONE and XMIT\_QUEUE\_TWO as one frame.

The length of the buffer is defined by BUFFER\_LEN\_ONE.

The buffer is not used if BUFFER\_LEN\_ONE is 0.

See "Transmit Buffers" on page 2-40 for details of transmit queues and buffers.

**For CCB3:** If this field is used to reference transmit data, it must be a 32-bit physical address, not a virtual address.

---

### **BUFFER\_TWO**

**Explanation:** The address of the buffer containing data to be transmitted.

The length of the buffer is defined by BUFFER\_LEN\_TWO. The data is transmitted following the data in XMIT\_QUEUE\_ONE, XMIT\_QUEUE\_TWO, and BUFFER\_ONE as one frame.

The buffer is not used if BUFFER\_LEN\_TWO is 0.

See "Transmit Buffers" on page 2-40 for details of transmit queues and buffers.

**For CCB3:** If this field is used to reference transmit data, it must be a 32-bit physical address, not a virtual address.

---

### **XMIT\_READ\_OPTION**

**Explanation:** To eliminate the need of issuing an individual READ command to receive notification for each completion of a transmit command, the application program can specify that completed transmit commands be chained together using the CCB\_POINTER of each CCB. The XMIT\_READ\_OPTION must be set for each individual transmit command. The following is a list of the various XMIT\_READ\_OPTIONS:

- X'00'** Chain this transmit command on a LINK STATION basis when this command completes. This option is only valid for TRANSMIT.I.FRAME commands.
- X'01'** Do not chain this transmit command when it completes.
- X'02'** Chain this transmit command on a SAP STATION basis when this command completes

---

## Chapter 4. NETBIOS

NETBIOS Overview	4-2
Version Information	4-2
The Network Control Block	4-3
NCB Field Explanations	4-3
NETBIOS Operational Parameters	4-7
Local Area Network Support Program Old Parameters	4-8
NETBIOS Notes	4-11
Local Area Network Support Program New Parameters	4-11
NETBIOS 3.0 (OS/2)	4-11
DLC Timers	4-12
NETBIOS (NCB) Calling Conventions	4-12
Local Area Network Support Program Calling Conventions	4-12
Dynamic Link Routine Interface Calling Conventions	4-12
Device Driver Interface Calling Conventions	4-13
NCB Completion	4-13
Wait Type Commands	4-13
No Wait Type Commands	4-13
NETBIOS Command Descriptions	4-14
NCB.ADD.GROUP.NAME	4-14
NCB.ADD.NAME	4-14
NCB.CALL	4-15
NCB.CANCEL	4-16
NCB.CHAIN.SEND	4-17
NCB.CHAIN.SEND.NO.ACK	4-18
NCB.DELETE.NAME	4-19
NCB.FIND.NAME	4-20
NCB.HANG.UP	4-21
NCB.LAN.STATUS.ALERT	4-22
NCB.LISTEN	4-22
NCB.RECEIVE	4-23
NCB.RECEIVE.ANY	4-24
NCB.RECEIVE.BROADCAST.DATAGRAM	4-25
NCB.RECEIVE.DATAGRAM	4-26
NCB.RESET	4-26
NCB.SEND	4-30
NCB.SEND.BROADCAST.DATAGRAM	4-31
NCB.SEND.DATAGRAM	4-32
NCB.SEND.NO.ACK	4-32
NCB.SESSION.STATUS	4-33
NCB.STATUS	4-35
NCB.TRACE	4-37
NCB.UNLINK	4-41



---

## NETBIOS Overview

NETBIOS can be used with any of the IBM Token-Ring Network or PC Network adapters to extend the adapter support software by permitting NETBIOS (Network Basic Input/Output System) application programs to operate on the network.

NETBIOS provides a communication interface between the application program and the attached medium. All communication functions from the physical layer through the session layer are handled by NETBIOS, the adapter support software and the adapter card.

A NETBIOS session is a logical connection between any two names on the network. A session is established by having an NCB.LISTEN issued from one name and an NCB.CALL issued from the other name. Once a session is established, two way guaranteed-delivery communication is possible between the two names.

Two basic types of data transfer are supported. Reliable data transfer is provided by the *session* layer. If data is lost or errors occur, NETBIOS will return an error code to the application program through the NCB. Data transfer using datagram support goes directly to the *link* layer. This type of data transfer is "best effort" and receipt of data is not guaranteed.

The following are needed to use NETBIOS application programs on an IBM network:

- An IBM Network adapter
- Adapter support software (Provided by the Local Area Network Support Program or Operating System/2 Extended Edition 1.1)
- NETBIOS (Provided by the Local Area Network Support Program or Operating System/2 Extended Edition 1.1)

NETBIOS maintains a table of names that the node is known by on the network. These names are provided to NETBIOS by the application program. A name can be a unique name or a group name. NETBIOS checks the network to verify that a unique name is not already in use at another adapter. A group name can be used by several adapters. Names are used as the basis for communication between application programs. If the name is in the NETBIOS name table, a session can be established. NETBIOS can have from 1 to 254 selectable names and one NAME\_NUMBER\_1, the default is 16 plus 1. All names are 16 characters long. The NAME\_NUMBER\_1 is always present and consists of 10 bytes of binary zeros followed by the adapter's universally administered address. This NETBIOS name is referred to as NETBIOS\_NAME\_NUMBER\_1.

## Version Information

NETBIOS has been made available in several different products. It is provided as a part of the Local Area Network Support Program. On the diskette for the Local Area Network Support Program, the NETBIOS file is DXMT0MOD.SYS. All currently supported NETBIOS version numbers are listed below.

**Note:** The information contained in this chapter does not pertain to the NETBIOS interface that was supplied on the *original* PC Network Adapter provided by the IBM PC Network Protocol Driver.

- The Local Area Network Support Program Version 1.00 provides NETBIOS 2.0.
- The Local Area Network Support Program Version 1.01 provides NETBIOS 2.1.
- The Local Area Network Support Program Version 1.02 provides NETBIOS 2.2.
- The Local Area Network Support Program Version 1.03 provides NETBIOS 2.3.
- The Local Area Network Support Program Version 1.10 provides NETBIOS 2.3.
- The Communications Manager provided with Operating System/2 Extended Edition 1.1 provides NETBIOS 3.0.

**Note:** NETBIOS 2.X includes all the changes from earlier releases of NETBIOS.

---

## The Network Control Block

NETBIOS is operated using a control block called the Network Control Block (NCB). (The Token-Ring Network NCB is the same as the NCB in PC Network.)

### NCB Field Explanations

The following chart shows the contents of the network control block.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
<b>Note:</b> Field names ending with @ indicate an address throughout this document.				
0	NCB_COMMAND	1	DB	Command field
1	NCB_RETCODE	1	DB	Return code
2	NCB_LSN	1	DB	Local session number
3	NCB_NUM	1	DB	Number of application program name
4	NCB_BUFFER@	4	DD	Pointer to message buffer address (segment:offset)
8	NCB_LENGTH	2	DW	Buffer length in bytes
10	NCB_CALLNAME	16	DB	Name on local or remote NETBIOS. This field has a different use for the CHAIN.SEND commands and RESET command.
26	NCB_NAME	16	DB	Name on the local NETBIOS session. This field has a different use for the NCB.RESET command.
42	NCB_RTO	1	DB	Receive timeout
43	NCB_STO	1	DB	Send timeout
<b>For all NETBIOS except the Operating System/2 Device Driver Interface:</b>				
44	NCB_POST@	4	DD	Pointer to post routine (segment:offset) or X'00000000'
<b>For NETBIOS using the Operating System/2 Device Driver Interface:</b>				
44	NCB_POST@	2	DB	A value returned to the post routine in register DI (0 = do not post)
46	NCB_DD_ID	2	DB	Device driver identification
<b>For all NETBIOS:</b>				
48	NCB_ADPTR_NUM	1	DB	Use X'00' for the primary adapter. Use X'01' for the alternate adapter.
49	NCB_CMD_CMPL	1	DB	Command status
50	NCB_RESERVE	14	DB	Reserved area for all commands except NCB.RESET

---

#### NCB\_COMMAND

**Explanation:** The command to be performed by the adapter. The high-order bit defines the wait/no-wait option.

When an application program issues a "wait" command to NETBIOS, control is not returned to the application program until the command is complete. When the command completes, check either the AL register or the NCB\_RETCODE field for the completion code.

**For NETBIOS using the Operating System/2 Device Driver Interface:** With the exception of RESET and CANCEL, the NETBIOS Device Driver Interface does not have wait commands.

When an application program issues a "no-wait" command to NETBIOS, control is returned to the application program at the earliest possible time, typically before the command completes. With the

no-wait option, two return codes are returned. An immediate return code is returned when the command is accepted by NETBIOS and the final return code is returned with the command completion. If the immediate return code in the AL register is not X'00', NETBIOS will not proceed and therefore will not provide a final return code.

When a no-wait command completes, if the NCB\_POST@ is zero, the completion code is set by NETBIOS. It is the responsibility of the application program to determine when NETBIOS is finished by periodically checking the completion code for a change from X'FF'. If NCB\_POST@ is not zero, NETBIOS takes the address to be a post routine and after command completion, control is given to the post routine. See the description for the NCB\_POST@ field.

---

## NCB\_RETCODE

**Explanation:** The completion code as provided by NETBIOS.

While the value is X'FF', the application program must not change either the control block or any data associated with the command.

- Return code value of X'00' indicates successful completion of the command.
- Return code values of X'01' through X'2F' indicate terminations that are described in "NCB Return Codes" on page B-27.
- Return code values of X'30' through X'40' indicate user errors that are described in "NCB Return Codes" on page B-27. These return codes are only returned when using Operating System/2 Extended Edition 1.1.
- Return code values of X'41' through X'4F' indicate user errors that are described in "NCB Return Codes" on page B-27.
- Return code values of X'50' through X'FE' indicate a PC system error or an adapter error and are described in "NCB Return Codes" on page B-27.

---

## NCB\_LSN

**Explanation:** A 1-byte field indicating the local session number. This is the number of the session the application program has with another name on the network. This is valid only after an NCB.CALL or NCB.LISTEN command has been completed successfully. For SEND and RECEIVE commands under session support, this field must be provided. NETBIOS uses a modulo 254 technique to provide numbers from X'01' to X'FE'.

The RESET command uses this field. This field is not used for datagram support.

---

## NCB\_NUM

**Explanation:** A 1-byte number provided by NETBIOS after an NCB.ADD.NAME or NCB.ADD.GROUP.NAME command is executed. This number, not the name, must be used with all datagram support commands and for NCB.RECEIVE.ANY commands.

The number for NETBIOS\_NAME\_NUMBER\_1 is always X'01'. NETBIOS uses a modulo 255 technique to provide numbers from X'02' to X'FE' for the remaining names.

The RESET command uses this field.

---

### NCB\_BUFFER@

**Explanation:** A 4-byte field containing the address of the buffer area assigned by the application program. This field is in define double-word (DD) format (segment:offset) and must be a valid address in PC system memory.

**For Operating System/2:** When using the Device Driver Interface of Operating System/2 Extended Edition 1.1, the address is a physical address.

---

### NCB\_LENGTH

**Explanation:** This field of 2 bytes indicates the length in bytes of the data buffer. For receive type commands, the field is updated by NETBIOS to indicate the number of bytes actually received.

---

### NCB\_CALLNAME

**Explanation:** This is a 16-byte name that the application program wants to communicate with. The name can either be on your adapter or any other adapter.

For an NCB.CHAIN.SEND or NCB.CHAIN.SEND.NO.ACK command, the first 6 bytes are used to specify the second buffer. The first 2 bytes are the length of the buffer and the remaining 4 bytes are the address of the buffer in memory.

**For Operating System/2:** When used as a buffer pointer for the Device Driver Interface of Operating System/2 Extended Edition 1.1, the NCB\_CALLNAME field is a physical address. This field is also used by the NCB.RESET command.

---

### NCB\_NAME

**Explanation:** A name that the node is known by on the network. The name is 16 bytes long. The NETBIOS\_NAME\_NUMBER\_1 may be used as a name. The NETBIOS\_NAME\_NUMBER\_1 is 10 bytes of zeros followed by the 6 bytes of the NODE\_ADDRESS.

**For Operating System/2 Extended Edition 1.1:** Also used by NCB.RESET for parameters.

---

### NCB\_RTO

**Explanation:** A 1-byte field used by the NCB.CALL and NCB.LISTEN commands to specify a timeout period for all receives associated with that session. The timeout value is specified in increments of 500 milliseconds. If X'00' is specified, the default is no timeout. The timeout period may be different for each session, but is fixed when the session is established. The timeout period at the other end of the session may also be different.

---

### NCB\_STO

**Explanation:** A 1-byte field used by the NCB.CALL and NCB.LISTEN commands to specify a timeout period for all sends associated with that session. The timeout value is specified in increments of 500 milliseconds. If X'00' is specified, the default is no timeout. The timeout period may be different for each session, but is fixed once the session is established. The timeout period at the other end of the session may also be different. Send timeouts should be used with caution because they will always end the session if they expire.

---

## NCB\_POST@

**Explanation:** If this field is not zero, it is an indication to NETBIOS that the application program has a post routine that gets control when the NCB completes.

- The post routine should be as short in duration as possible.
- The register contents upon entry are listed here.
  - AX = the NCB completion code (AH is always zero).
  - ES and BX point to the NCB.

If the post address is all zeros, the post routine will not be called by NETBIOS and the application program must check the return code field for a change from X'FF'.

**For NETBIOS Used with DOS:** NCB\_POST@ is a four-byte address (segment:offset) that gets control. The application program returns control by issuing an "interrupt return" instruction.

**For Operating System/2 NETBIOS Using the Dynamic Link Routine Interface:** NCB\_POST@ is a four-byte address (selector:offset) that gets control. The application program returns control by issuing a "return" instruction (return from a "call").

**For Operating System/2 NETBIOS Using the Device Driver Interface:** NCB\_POST@ is a two-byte number. NETBIOS gives control to the post-routine by issuing a "Call Far" with the value X'0002' pushed on the stack and with the value of NCB\_POST@ in register DI. Register DS contains the application program's device driver data segment. Note that the "Call Far" is made to the application program's device driver entry point. Control is returned to NETBIOS by issuing a "Return Far 2" instruction (to account for the X'0002' that was pushed on the stack).

---

## NCB\_DD\_ID

**Explanation:** The identification number of the device driver.

This field is supplied to the device driver application program by NETBIOS when the first NCB.RESET completes. The application program must then provide this value in all subsequent commands.

---

## NCB\_ADPTR\_NUM

**Explanation:** Defines which adapter is to be used. Must be either X'00' to use the primary adapter or X'01' for the alternate adapter. The adapter must have the corresponding (primary/alternate) switch set correctly. Other values are reserved.

---

## NCB\_CMD\_CMPL

**Explanation:** This value is the same as the NCB\_RETCODE.

---

## NCB\_RESERVE

**Explanation:** A 14-byte reserved field. Used as a work area by NETBIOS. Also contains certain system information when an NCB completes with either a X'4x' or X'Fx' return code:

- For the Local Area Network Support Program

Offset from NCB_RESERVE	Length in Bytes	Meaning
0	2	Value of last adapter bring-up code
2	2	Return code of last CCB open issued
4	2	Last network status
6	2	Last adapter check status
8	2	Last PC error code
10	1	Last CCB code generated by NETBIOS during NCB.RESET or initialization
11	1	Return code of CCB in offset 10

- For Operating System/2 Extended Edition 1.1

Offset from NCB_RESERVE	Length in Bytes	Meaning
0	2	Value of last adapter bring-up code
2	2	Return code of last CCB open issued
4	2	Last network status
6	2	Last adapter check status
8	2	Internal NETBIOS information
10	2	Last PC error code
12	1	Last CCB code generated by NETBIOS during NCB.RESET or initialization
13	1	Return code of CCB in offset 12

---

## NETBIOS Operational Parameters

There are three different methods of setting the NETBIOS operational parameters:

- Local Area Network Support Program Old Parameters
- Local Area Network Support Program Parameters (see Appendix D, "Local Area Network Support Program Information")

This allows the user to modify the NETBIOS operational parameters at load time by adding the parameters to the device driver command line.

- Operating System/2 NETBIOS 3.0 Parameters (see Appendix E, "Operating System/2 Extended Edition 1.1 Information")

This allows the user to modify the NETBIOS operational parameters by changing the Communications Manager configuration file.

## Local Area Network Support Program Old Parameters

This allows the user to modify the NETBIOS operational parameters by using the DIR.OPEN.ADAPTER command.

The parameters must be provided at open time and for each NCB.RESET command. If any of these parameters causes an error, the command will terminate with a CCB\_RETCODE or NCB\_RETCODE as follows:

- If a DIR.OPEN.ADAPTER command, the CCB\_RETCODE is X'10'. This can occur if the NCB\_MAX\_NAMES or NCB\_MAX\_SESSIONS values are not less than 255, or if there is insufficient work space available to satisfy the values of NCB\_STATIONS, NCB\_MAX\_NAMES, NCB\_MAX, and NCB\_MAX\_SESSIONS.
- If a NETBIOS command, the NCB\_RETCODE is X'FC'. This can occur if any of the remaining parameters are found to be invalid when NETBIOS issues its DLC.OPEN.SAP command.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0		4	--	Adapter work area
4	NCB_TIMER_T1	1	DB	T1 value (response timer)
5	NCB_TIMER_T2	1	DB	T2 value (acknowledgment timer)
6	NCB_TIMER_TI	1	DB	Ti value (inactivity timer)
7	NCB_MAXOUT	1	DB	Maximum transmits without a receive acknowledgment
8	NCB_MAXIN	1	DB	Maximum receives without a transmit acknowledgment
9	NCB_MAXOUT_INCR	1	DB	Dynamic window increment value
10	NCB_MAX_RETRY	1	DB	N2 value
11		1	DB	Adapter work area
12		3		Reserved
15	NCB_ACCESS_PRI	1	DB	Ring access priority
16	NCB_STATIONS	1	DB	Maximum NETBIOS link stations
17		19	--	Adapter work area
36	NCB_MAX_NAMES	1	DB	Maximum names in names table
37	NCB_MAX	1	DB	Maximum outstanding NCBs
38	NCB_MAX_SESSIONS	1	DB	Maximum sessions
39	Reserved	1	DB	
40	Reserved	1	DB	
41	NCB_OPTIONS	1	DB	Various options
42	NCB_POOL_LENGTH	2	DW	Length of area at NCB_POOL_ADDRESS
44	NCB_POOL_ADDRESS	4	DD	Starting segment of Message work area
48	NCB_TRANSMIT_TIMEOUT	1	DB	Time to wait for one query
49	NCB_TRANSMIT_COUNT	1	DB	Maximum number of times to transmit queries

### NCB\_TIMER\_T1

**Explanation:** Specifies the time period between 1 and 10 used to determine an inoperative condition on a link. The time intervals are defined by the DIR.OPEN.ADAPTER command. If the value is zero, the default of 5 is used. See "DLC Timers" on page 4-12.

---

## NCB\_TIMER\_T2

**Explanation:** Specifies the time period between 1 and 10 used to delay transmission of an acknowledgment for a received I-LPDU for link stations using this SAP. The time intervals are defined by the DIR.OPEN.ADAPTER command. If the value is zero, the default of 2 is used. If the value is greater than 10, the acknowledgment timer is not implemented and acknowledgments will be sent at the earliest opportunity. See "DLC Timers" on page 4-12.

---

## NCB\_TIMER\_T1

**Explanation:** Specifies the time period between 1 and 10 used to determine an inactive condition on a link. The time intervals are defined by the DIR.OPEN.ADAPTER command. If the value is zero, the default of 3 is used. See "DLC Timers" on page 4-12.

---

## NCB\_MAXOUT

**Explanation:** Specifies the maximum number of sequentially numbered transmitted I-LPDUs that a link station on NETBIOS SAP may have outstanding at any one time. The maximum valid value is 127. If the value is zero, the default of 2 is used.

---

## NCB\_MAXIN

**Explanation:** Specifies the maximum number of sequentially numbered received I-LPDUs that a link station on NETBIOS SAP may receive prior to sending an acknowledgment. The maximum valid value is 127. If the value is zero, the default of 1 is used.

---

## NCB\_MAXOUT\_INCR

**Explanation:** This dynamic window increment value, is used to reduce bridge congestion. If the two end points of a session are on different rings, and the adapter detects an error condition requiring retransmission, the MAXOUT counter will be set to 1. It will then be incremented by one each time MAXOUT\_INCR frames are acknowledged by the remote station, until it reaches the value of this field. If this field is set to a value of zero, the default of one is used. For more details, see the *IBM Token-Ring Network Architecture Reference*.

---

## NCB\_MAX\_RETRY

**Explanation:** Specifies the number of retries for an unacknowledged command LPDU, or in the case of an I-LPDU timeout, the number of times that the non-responding remote link station will be polled with an RR/RNR command LPDU. The maximum valid value is 255. If the value is zero, the default of 8 is used.

---

## NCB\_ACCESS\_PRI

**Explanation:** The transmit access priority value to be placed in the AC byte of all transmissions from the link station and SAP. The format is B'nnn00000', where 'nnn' is the access priority value. No checking is done and the low-order 5 bits are ignored. If the access priority is higher than allowed for the adapter, the error will be detected on the first transmission.

---

## NCB\_STATIONS

**Explanation:** The number of link stations that may be active at one time. This value must not exceed the value of DLC\_MAX\_STATIONS. During execution of an NCB.RESET command, this value must not exceed the current number of available link stations. If the value is zero, the default of 6 is used.

---



---

## NCB\_MAX\_NAMES

**Explanation:** The maximum number of names that may be in the name table. The adapter itself is entered as a name, using one of the positions. The maximum valid value is 254. If the value is zero, the default of 17 is used.

---

## NCB\_MAX

**Explanation:** The number of NCBs that may be outstanding at one time.

**Note:** This value indicates the number of “no-wait” and “wait” commands that may be issued.

The maximum valid value is 255. If the value is zero, the default of 12 is used.

---

## NCB\_MAX\_SESSIONS

**Explanation:** The maximum number of sessions that may be active at one time. The maximum valid value is 254. If the value is less than the value of NCB\_STATIONS, the NCB\_STATIONS value is used.

---

## NCB\_OPTIONS

**Explanation:** Various options, each represented by a bit. If the bit has a value of B'1', the option is active. The high-order bit is the leftmost bit, 7.

- Bit 7 is “auto open.”

When received by the adapter system interface, bit 7 causes an NCB.RESET command to close and then open the adapter. If bit 7 is zero, the NCB.RESET command will not perform the close and will ignore the fields containing the “number of sessions” and the “number of commands.” (They remain as defined in the DIR.OPEN.ADAPTER command.)

- Bit 6 is reserved.
- Bit 5 is “this ring only.”

When this bit is set on, NETBIOS will assume that all nodes are on the same ring.

- Bits 4 – 0 are reserved.
- 

## NCB\_POOL\_LENGTH

**Explanation:** The number of bytes of system memory assigned by the application for the NETBIOS work-area pool. If the value is zero, NETBIOS internal work-area in system memory as defined at load time is used.

---

## NCB\_POOL\_ADDRESS

**Explanation:** The starting address of the NETBIOS work-area pool for the adapter support software to build tables, buffers, and control blocks. If the NCB\_POOL\_LENGTH value is zero, this parameter is ignored.

---

## NCB\_TRANSMIT\_TIMEOUT

**Explanation:** A value to define the amount of time that NETBIOS will wait for a response to a private query on the network (such as an “add name query”). The value is in half-second increments. A value of 10 represents a time of 5 seconds. If the value is zero, the default of 1 (1/2 second) is used. If the value is greater than 20, 20 (10 seconds) is used.

---

## NCB\_TRANSMIT\_COUNT

**Explanation:** A value to define the number of times that private network queries, such as an “add name query,” will be transmitted for a given command. If a query is transmitted more than one time, the next one is transmitted after the NCB\_TRANSMIT\_TIMEOUT expires. If the value is zero, the default of 6 is used. If the value is greater than 10, 10 is used.

## NETBIOS Notes

- The number of stations determines how many physical nodes the adapter may establish connection with at one time.
- The number of sessions determines how many different NETBIOS sessions may be active at one time.
- There may be multiple sessions on one station.
- NETBIOS provides internal work area in system memory as defined at load time for buffers and work area. The application program may define its own space by properly coding the NCB\_POOL\_ADDRESS and NCB\_POOL\_LENGTH parameters.
- To calculate whether additional work space in bytes is needed, use the following formula:
  - 1900 plus
  - 20 times the maximum names (NCB\_MAX\_NAMES) plus
  - 40 times the maximum stations (NCB\_STATIONS) plus
  - 44 times the maximum sessions (NCB\_SESSIONS) plus
  - 100 times the maximum commands (NCB\_MAX) plus
  - 100 times the number of transmit buffers
- The remainder of the defined work area is configured as transmit buffers and there must be room for at least 10.
- All tables and buffers, except the receive buffers, must be in the same segment. Therefore, if these tables use more than the defined work area, the area defined by the application program must be large enough to hold all this information.
- The receive buffers are allocated by the adapter from the defined work area and any application program assigned memory after all other memory requirements are complete.

## Local Area Network Support Program New Parameters

This allows the user to modify the NETBIOS operational parameters at the NETBIOS device driver load time by adding the parameters to the device driver command line. See Appendix D, “Local Area Network Support Program Information” for a complete description of these new parameters.

## NETBIOS 3.0 (OS/2)

This allows the user to modify the NETBIOS operational parameters with the Communications Manager Configuration. See Appendix E, “Operating System/2 Extended Edition 1.1 Information” for a complete description of these parameters.

## DLC Timers

The following table indicates the timer values, in seconds, when the various parameter values are used.

These values are using the default DLC 'tick' values.

DLC.T1		DLC.T2		DLC.TI	
Parameter	Seconds	Parameter	Seconds	Parameter	Seconds
T1=1	0.2-0.4	T2=1	.04-.08	TI=1	1-2
T1=2	0.4-0.6	T2=2*	.08-.12	TI=2	2-3
T1=3	0.6-0.8	T2=3	.12-.16	TI=3*	3-4
T1=4	0.8-1.0	T2=4	.16-.20	TI=4	4-5
T1=5*	1.0-1.2	T2=5	.20-.24	TI=5	5-6
T1=6	1-2	T2=6	0.4-0.8	TI=6	5-10
T1=7	2-3	T2=7	0.8-1.2	TI=7	10-15
T1=8	3-4	T2=8	1.2-1.6	TI=8	15-20
T1=9	4-5	T2=9	1.6-2.0	TI=9	20-25
T1=10	5-6	T2=10	2.0-2.4	TI=10	25-30
		T2=11	disabled		
* This value is the default.					

---

## NETBIOS (NCB) Calling Conventions

### Local Area Network Support Program Calling Conventions

An application program using the Local Area Network Support Program should use the following conventions to initiate an NCB:

1. Register ES and BX point to the NCB.
2. Execute instruction Interrupt X'5C'.

### Dynamic Link Routine Interface Calling Conventions

An application program using the Operating System/2 Dynamic Link Routine Interface should use the following conventions to initiate an NCB:

1. Push the NCB's selector value onto the stack.
2. Push the NCB's offset value onto the stack.
3. 'Call Far' NETBIOS.

## Device Driver Interface Calling Conventions

An application program using the Operating System/2 Device Driver Interface should use the following conventions to initiate an NCB:

1. Registers ES and BX point to the NCB.
2. Push the value X'0000' onto the stack.
3. 'Call Far' the address obtained when the application program executed an Operating System/2 'ATTACHDD' to name NETBIOS\$.

**Note:** When using the Operating System/2 Device Driver Interface NETBIOS, the NCB and any buffers must be locked by the application program.

---

## NCB Completion

### Wait Type Commands

If the command issued is a wait-type command, control is not returned to the next instruction until the adapter has completed the command. When the command does complete, the return code will be in both the AL register and the NCB\_RETCODE field.

### No Wait Type Commands

If the command is a no-wait type, NETBIOS presents two return codes. An immediate return code is posted to the AL register. If the immediate return code in the AL register is other than X'00', command processing is complete. If the immediate return code is X'00', processing continues and provides a final return code when the command completes.

If the NCB\_POST@ field of the NCB is zero, the final return code is placed in the NCB\_CMD\_CMPL field, which must be checked for change by the application program. If the application is checking the NCB\_CMD\_CMPL field, a change of value from X'FF' indicates command completion. This value is the final return code.

If the NCB\_POST@ field is not zero, NETBIOS will give control to the post routine after setting the final return code in both the AL register and the NCB\_RETCODE field. The application program continues with the instruction located by the contents of the NCB\_POST@ field (the command completion appendage).

---

## NETBIOS Command Descriptions

Each command description begins with a box containing the command name. The hexadecimal number at the top of the box is the command code value. Both the wait and no-wait values are supplied when applicable.

### NCB.ADD.GROUP.NAME

Hex 36 Wait    B6 No-Wait
NCB.ADD.GROUP.NAME

**Command Description:** This command adds a 16-character name to the table of names. The name cannot be used by any other station across the network as a unique name, but can be added by any station as a group name. This is a name that this station will be known by.

**Command Specifics:** When NETBIOS processes this command, it sends name query requests on the network. If no reply is received, it is assumed that no other node on the network has defined the name as a unique name, and the name is added to that node. The adapter returns the number of the name in the NCB\_NUM field. This number is used in datagram support and for NCB.RECEIVE.ANY commands.

**Supplied Fields:**

- NCB\_ADPTR\_NUM (Adapter number 0 or 1)
- NCB\_NAME
- NCB\_POST@ (If the no-wait option is used)
- NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

**Returned Fields:**

- NCB\_RETCODE
- NCB\_NUM
- NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

### NCB.ADD.NAME

Hex 30 Wait    B0 No-Wait
NCB.ADD.NAME

**Command Description:** This command adds a 16-character name to the table of names. The name must be unique across the network. This is a name that this station will be known by.

**Command Specifics:** When NETBIOS processes this command, it sends name query requests on the network. If no reply is received, the name is assumed to be unique and is added to the table of names. The adapter returns the number of the name in the NCB\_NUM field. This number is used in datagram support and for NCB.RECEIVE.ANY commands.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_NAME  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

**Returned Fields:**

NCB\_RETCODE  
 NCB\_NUM  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## NCB.CALL

Hex 10 Wait      90 No-Wait

NCB.CALL

**Command Description:** This command opens a session with another name specified by the NCB\_CALLNAME field using the local name specified by the supplied NCB\_NAME field.

**Command Specifics:** The destination name station must have an NCB.LISTEN command outstanding in order for the session to be established. Sessions may be established with either a local or remote name. Multiple sessions may be established with the same pair of names. The timeout intervals are specified in 500-millisecond units. (A value of zero means that no timeout will occur.) The system timeout intervals and retry count are constants in NETBIOS. The NCB.CALL command itself aborts if unsuccessful after the system timeout interval (retry count exhausted). When the call is completed, a local session number (LSN) is assigned and used thereafter to refer to the established session.

Local session numbers (NCB\_LSN) are assigned in a round-robin technique, starting from the next available value within the range of 1 to 254.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_NAME  
 NCB\_CALLNAME  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_RTO (500 milliseconds increments) (If the field is set at X'00', no receive timeout will occur.)  
 NCB\_STO (500 milliseconds increments) (If the field is set at X'00', no send timeout will occur.)

## NCB.CANCEL

### Returned Fields:

NCB\_RETCODE  
NCB\_LSN  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## NCB.CANCEL

Hex 35 Wait

NCB.CANCEL

**Command Description:** This command requests that the command, whose NCB is at the address given by NCB\_BUFFER, be cancelled.

**Command Specifics:** Use caution when cancelling session commands. Cancellation will always end the session.

These commands may be cancelled:

- NCB.CALL
- NCB.CHAIN.SEND
- NCB.CHAIN.SEND.NO.ACK
- NCB.HANG.UP
- NCB.LAN.STATUS.ALERT
- NCB.LISTEN
- NCB.RECEIVE
- NCB.RECEIVE.ANY
- NCB.RECEIVE.BROADCAST.DATAGRAM
- NCB.RECEIVE.DATAGRAM
- NCB.SEND
- NCB.SEND.NO.ACK
- NCB.STATUS

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_BUFFER@ (Address of the NCB to be cancelled)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If an error occurs X'4X' or X'5X')

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## NCB.CHAIN.SEND

Hex 17 Wait    97 No-Wait

NCB.CHAIN.SEND

**Command Description:** This command sends data to the session partner as defined by the session number in the NCB\_LSN field. The data to send is in the buffer pointed to by the NCB\_BUFFER@ field and the NCB\_CALLNAME field.

**Command Specifics:** The data in the second buffer is concatenated to the data in the first buffer and sent as a single message. The NCB\_CALLNAME field is used to specify the length and address of the second buffer. The length is specified in the first 2 bytes and the second buffer address in the next 4 bytes.

When a session is closed by the remote side, all NCB.CHAIN.SEND commands pending for the closed session will be returned with a "session closed" status. If a local NCB.HANG.UP command is issued with any pending NCB.CHAIN.SEND commands, the NCB.CHAIN.SEND commands are completed.

If a session is aborted, a "session ended abnormally" status is returned. If the NCB.CHAIN.SEND timeout expires, the session is aborted and a "command timed out" status is returned. Timeout values for the NCB.CHAIN.SEND are associated with the session when an NCB.CALL or NCB.LISTEN is issued and cannot be specified with this command.

Message size must be between 0 and 131,070 bytes long.

If more than one NCB.SEND or NCB.CHAIN.SEND is pending, the data is transmitted in a first-in first-out order within a session.

If the NCB.CHAIN.SEND command cannot be completed for any reason, the session ends abnormally and the session is dropped. This is done to guarantee data integrity.

### Supplied Fields:

- NCB\_ADPTR\_NUM (Adapter number 0 or 1)
- NCB\_LENGTH
- NCB\_BUFFER@
- NCB\_POST@ (If the no-wait option is used)
- NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)
- NCB\_LSN
- NCB\_CALLNAME (Specifies the second buffer)
  - Bytes 0 - 1 = NCB\_LENGTH2
  - Bytes 2 - 5 = NCB\_BUFFER2@

### Returned Fields:

- NCB\_RETCODE
- NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .



## NCB.CHAIN.SEND.NO.ACK

Hex 72 Wait      F2 No-Wait

### NCB.CHAIN.SEND.NO.ACK

**Command Description:** This command has the benefits as the NCB.SEND.NO.ACK in that it does not require an acknowledgement.

This command sends data to the session partner as defined by the session number in the NCB\_LSN field. The data to send is in the buffer pointed to by the NCB\_BUFFER@ field. Two buffers can be chained together with this command.

The data in the second buffer is concatenated to the data in the first buffer and sent as a single message. The NCB.CALLNAME field is used to specify the length and address of the second buffer. The length is specified in the first 2 bytes and the second buffer address in the next 4 bytes.

When a session is closed by the remote side, all NCB.CHAIN.SEND.NO.ACK commands pending for the closed session will be returned with a "session closed" status. If a local NCB.HANG.UP command is issued with any pending NCB.CHAIN.SEND.NO.ACK commands, the NCB.CHAIN.SEND.NO.ACK commands are completed.

If a session is aborted, a "session ended abnormally" status is returned.

Message size must be between 0 and 65,535 bytes long.

If more than one send type command is pending, the data is transmitted in a first-in first-out order within a session.

**Command Specifics:** Lost data, as indicated by a X'07' return code, as well as the possible need for retransmission is the responsibility of the application program.

If the remote node cannot process the NCB.CHAIN.SEND.NO.ACK, it will be handled as an NCB.CHAIN.SEND; a DATA\_ACK must be received before the NCB.CHAIN.SEND.NO.ACK will complete.

It is the recommended that this command be used for transactions and not data transfers.

#### Notes:

- Note the following on a X'07' return code.
  - The return code is generated for the node that issued NCB.CHAIN.SEND.NO.ACK.
  - The command terminating with the X'07' must be re-issued.
  - The return code is issued if the receiver either had no NCB.RECEIVE or NCB.RECEIVE.ANY up, or the transmitted data exceeded the length of the receive buffer.
  - Unless the user is aware of the number of NCB.CHAIN.SEND.NO.ACKs issued, he has no indication of how many were not completed successfully.
  - All other NCB.CHAIN.SEND return codes apply.
- For data transmitted with NCB.CHAIN.SEND.NO.ACK, only one receive command will be satisfied. The NCB.CHAIN.SEND.NO.ACK data buffer must be no longer than the remote receive buffer for successful completion.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_LSN  
 NCB\_CALLNAME (Specifies the second buffer)

- Bytes 0 - 1 = NCB\_LENGTH2 DW format
- Bytes 2 - 5 = NCB\_BUFFER2@ DD format

**Returned Fields:**

NCB\_RETCODE  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

**NCB.DELETE.NAME**

Hex 31 Wait    B1 No-Wait

**NCB.DELETE.NAME**

**Command Description:** This command deletes a 16-character name from the table of names.

**Command Specifics:** If the name has active sessions when the NCB.DELETE.NAME command is issued, the name is flagged as “deregistered” and the “command completed, name has active sessions” status is returned to the user. The delete is delayed until the sessions associated with the name are closed. A deregistered name is not usable by subsequent NCBs.

If the name has only pending non-active session commands when the NCB.DELETE.NAME command is issued, the name is removed and the “command completed” status is returned to the user. The pending non-active session commands are terminated immediately with the “name was deleted” status. Non-active session commands are :

- NCB.LISTEN
- NCB.RECEIVE.ANY
- NCB.RECEIVE.DATAGRAM
- NCB.RECEIVE.BROADCAST.DATAGRAM

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_NAME  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

## NCB.FIND.NAME

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.FIND.NAME

Hex 78 Wait      F8 No-Wait
NCB.FIND.NAME

**Command Description:** This command finds the location on the network of a 16-character name. The name is specified in the NCB\_CALLNAME field.

**Command Specifics:** NETBIOS sends a name query request on the network. If any remote nodes have the requested name registered, they respond with an indication of how they have the name registered (unique/group).

If no response is received within the system timeout period, NETBIOS returns a CCB\_RETCODE of X'05'. If responses are received, NETBIOS returns the number of nodes that responded, followed by every unique LAN header from each responding node. The LAN header contains the adapter address of the remote node where the name is located. The returned data is located at the buffer address specified by the NCB\_BUFFER@ field, and the NCB\_LENGTH indicates the number of bytes of data stored. There is no guarantee that all nodes will respond within the given timeout period.

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LENGTH  
NCB\_BUFFER@  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_CALLNAME

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)  
NCB\_LENGTH

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## Data Areas Returned

OFF-SET	BYTE LEN	8086 TYPE	DESTINATION
0	2	DW	Number of nodes responding (will be greater than one only if used as a group name)
2	1	DB	Reserved
3	1	DB	Status: X'00' unique name X'01' group name
4	*	DB	The LAN header length and the LAN header from the remote node(s) where the name is located, (contains the address of the remote node). Each entry includes the header length and the LAN header for a total of 33 bytes for each entry. See the example of a LAN header below.

Length	PCF0	PCF1	Dest. Addr.	Source Addr.	Routing Info	
0	1	2	3	8-9	14-15	32

Refer to Figures 1-5 and 1-6 starting on page 1-11 for a description of the header data.

## NCB.HANG.UP

Hex 12 Wait	92 No-Wait
<b>NCB.HANG.UP</b>	

**Command Description:** This command closes the session specified by the local session number.

**Command Specifics:** When an NCB.HANG.UP command is issued to the adapter, all pending (local) NCB.RECEIVE commands are terminated and returned to the issuer with "session closed" in the NCB\_RETCODE field. The termination is valid whether or not any data had been transferred by the pending command. If a local NCB.SEND command is pending, the NCB.HANG.UP command will wait for up to 20 seconds for a pending send to complete. This delay is true whether or not the command has begun to transfer data, or is waiting for the remote side to issue an NCB.RECEIVE command. The NCB.HANG.UP is performed when any of the following conditions occur:

- The NCB.SEND completes.
- The NCB.SEND has aborted.
- The NCB.SEND fails because the session was terminated by the other side with an NCB.HANG.UP.
- The NCB.SEND fails because of the timeout specified when the session was opened.

If one of the above conditions does not occur within the system timeout period after the NCB.HANG.UP command is issued, the NCB.HANG.UP command is returned with a "command timed out" status and the session is aborted.

When a session closes, all NCB.SEND and NCB.RECEIVE commands pending on the closed session are returned to the user with a "session closed" status. If an NCB.RECEIVE.ANY command is pending on the local name used by the session, it is returned to the issuer with a "session closed" status.

Only a single NCB.RECEIVE.ANY command will be returned even though many NCB.RECEIVE.ANY commands are pending. Even though a single NCB.RECEIVE.ANY command is returned, many NCB.SENDs or NCB.RECEIVEs can be returned when pending.

When a session is abnormally terminated, all outstanding commands on that session will be returned to the issuer with a "session ended abnormally" status. When one side of a NETBIOS session issues NCB.HANG.UP, the remote partner will not free the session resources until at least one command completes, indicating the session is closed.

## NCB.LISTEN

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LSN  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## NCB.LAN.STATUS.ALERT

Hex F3 No-Wait

**NCB.LAN.STATUS.ALERT**

**Command Description:** This command is used by application programs that want to be notified of temporary ring error conditions that last for over one minute.

**Command Specifics:** There are two conditions where this command is completed:

- There is no temporary ring error or the current ring error has existed for less than one minute. In this case, any NCB.LAN.STATUS.ALERT commands completed will be queued by NETBIOS.
- The current ring error condition has existed for more than one minute. In this case all queued NCB.LAN.STATUS.ALERT commands will complete with a good return code. Any commands issued while this condition exists will be completed with a good return code.

**Note:** The queued NCB.LAN.STATUS.ALERT commands can be cancelled with the CANCEL command.

### Supplied Fields:

NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_COMMAND

### Returned Fields:

NCB\_RETCODE

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## NCB.LISTEN

Hex 11 Wait    91 No-Wait

**NCB.LISTEN**

**Command Description:** This command enables a session to be opened with the name specified in the NCB\_CALLNAME field, using the name specified by the NCB\_NAME field.

**Command Specifics:** If the NCB\_CALLNAME field has an asterisk (\*), a session will be established with any network node that issues an NCB.CALL to the local name.

NCB.LISTEN for a specific name has priority over an NCB.LISTEN for any name. Sessions may be established with either a local or remote name. Multiple sessions may be established with the same pair of names.

The timeout intervals are specified in 500-millisecond units. (A value of zero means that no timeout will occur.) An NCB.LISTEN command will not time out, but an NCB.LISTEN occupies a session entry and is considered a pending session in information returned in an NCB.STATUS command. Local session numbers (LSN) are assigned in a round-robin technique starting with the next available value within the range from 1 to 254. Also, if an asterisk (\*) is used for the called name, the name that made the call will be returned in the NCB\_CALLNAME field.

The error "name conflict detected" is returned if, during the completion for an NCB.LISTEN command, a name exists in more than one table. All nodes with the name registered, except the one where the NCB.LISTEN command has returned successfully, will report the "name conflict detected" error.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)

NCB\_NAME

NCB\_CALLNAME (This may be specified in the first byte as an asterisk (\*). The asterisk is used to listen for a call from any name to the local name. If a name is specified in this field, it takes priority over a name of an asterisk.)

NCB\_POST@ (If the no-wait option is used)

NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)

NCB\_RTO (500-millisecond increments) (If the field is set at X'00', no receive timeout will occur.)

NCB\_STO (500-millisecond increments) (If the field is set at X'00', no send timeout will occur.)

**Returned Fields:**

NCB\_RETCODE

NCB\_LSN

NCB\_CALLNAME (If NCB.LISTEN for any name is used, specified with an asterisk)

NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.RECEIVE

Hex 15 Wait	95 No-Wait
<b>NCB.RECEIVE</b>	

**Command Description:** This command receives data from its session partner that has issued one of the following commands:

- NCB.CHAIN.SEND
- NCB.CHAIN.SEND.NO.ACK
- NCB.SEND
- NCB.SEND.NO.ACK

## NCB.RECEIVE.ANY

**Command Specifics:** If more than one RECEIVE type command is outstanding, the NCB.RECEIVE has priority over the NCB.RECEIVE.ANY command. NCB.RECEIVE commands are processed in a first-in, first-out order. Timeout values are specified during an NCB.CALL or NCB.LISTEN and cannot be specified with this command.

When a session is closed, either by a local session close command or by the remote side closing the session, all pending NCBs for that session are returned with a "session closed" status.

A return code of X'06' is posted in the NCB\_RETCODE field if the receive buffer is not large enough for the message being received. Another receive can be issued to obtain the rest of the information before an NCB.SEND timeout occurs.

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LSN  
NCB\_BUFFER@  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_LENGTH

### Returned Fields:

NCB\_RETCODE  
NCB\_LENGTH  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.RECEIVE.ANY

Hex 16 Wait    96 No-Wait

### NCB.RECEIVE.ANY

**Command Description:** This command receives data for any session using the name defined in NCB\_NUM. The data is sent from one of the following commands:

- NCB.CHAIN.SEND
- NCB.CHAIN.SEND.NO.ACK
- NCB.SEND
- NCB.SEND.NO.ACK

You must use the name number instead of the name when issuing this command.

**Command Specifics:** If more than one RECEIVE type command is outstanding, the NCB.RECEIVE command has priority over the NCB.RECEIVE.ANY command.

If the NCB\_NUM field is set to X'FF' by the application program, then the receive is for any remote name that you have a session with, for any of your names. NETBIOS will set the NCB\_NUM field to the number of the name for which the data was received when the return code is presented.

When a session is closed, either by a local session close command, by the remote side closing the session, or a session abort, one NCB.RECEIVE.ANY or NCB.RECEIVE name will be posted with "session closed" or "session aborted" regardless of the number of session receives that may be pending. If an NCB.RECEIVE.ANY or an NCB.RECEIVE name is pending, it will post a "session closed" with the NCB\_LSN field containing the session number that closed. An NCB.RECEIVE.ANY with no name specified will post only if no NCB.RECEIVE.ANY name is pending for the session with that name.

A return code of X'06' is posted in the NCB\_RETCODE field if the receive buffer is not large enough for the message being received. Another receive can be issued to obtain the rest of the information before a timeout occurs.

When using DOS application programs, use "NCB.RECEIVE.ANY to any name" with caution as this command can receive messages for other programs running in the Personal Computer.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_NUM

**Returned Fields:**

NCB\_RETCODE  
 NCB\_LSN  
 NCB\_LENGTH  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)  
 NCB\_NUM (If X'FF' was specified in this field when issued)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

**NCB.RECEIVE.BROADCAST.DATAGRAM**

Hex 23 Wait    A3 No-Wait

**NCB.RECEIVE.BROADCAST.DATAGRAM**

**Command Description:** This command receives a datagram message from any name on the network that issues an NCB.SEND.BROADCAST.DATAGRAM.

**Command Specifics:** A "message incomplete" status is returned if the receive buffer is not large enough for the data being received. The remaining data is lost at this point.

When a broadcast datagram is received, all broadcast datagrams are completed.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_NUM



## NCB.RESET

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)  
NCB\_LENGTH  
NCB\_CALLNAME

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.RECEIVE.DATAGRAM

Hex 21 Wait    A1 No-Wait

### NCB.RECEIVE.DATAGRAM

**Command Description:** This command receives a datagram message from any name on the network that issues an NCB.SEND.DATAGRAM.

**Command Specifics:** A "message incomplete" status is returned if the receive buffer is not large enough for the data being received. The remaining data is lost at this point.

This command will not receive a broadcast datagram.

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LENGTH  
NCB\_BUFFER@  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_NUM (If X'FF', then receive a datagram from any name on the network for any name in the local table.)

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)  
NCB\_LENGTH  
NCB\_CALLNAME

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.RESET

There are four basic environments in NETBIOS:

1. Local Area Network Support Program using 'old' load NETBIOS parameters
2. Local Area Network Support Program using 'new' load NETBIOS parameters
3. Operating System/2 Extended Edition 1.1 applications using the Dynamic Link Routine Interface
4. Operating System/2 Extended Edition 1.1 applications using the Device Driver Interface.

In each of these environments, the RESET command operates in a different fashion. The following sections describe RESET operation in each of these environments.

Hex 32 Wait

**NCB.RESET with the Local Area Network Support Program Using Old NETBIOS Parameters**

**Command Description:** When RESET is issued, the following occurs:

1. All current NETBIOS names are deleted, all current sessions are aborted, and all outstanding NCBs are purged.
2. If indicated in the last DIR.OPEN.ADAPTER command, the adapter will be closed.
3. If the adapter is closed, it will be re-opened.

**Command Specifics:** If NETBIOS opens the adapter, the adapter will be closed on each RESET. If the application program opens the adapter, it may choose to keep from closing the adapter. See "DIR.OPEN.ADAPTER" on page 3-34.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LSN (The number of sessions. If 0, the default is 6)  
 NCB\_NUM (The number of NCBs. If 0, the default is 12)

**Returned Fields:**

MCB\_RETCODE  
 MCB\_RESERVE

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

Hex 32 Wait

**NCB.RESET with the Local Area Network Support Program Using New NETBIOS Parameters**

**Command Description:** When RESET is issued, the following occurs:

1. All current NETBIOS names are deleted, all current sessions are aborted, and all outstanding NCBs are purged.
2. If the load parameter CLOSE.ON.RESET is coded as YES, the adapter will be closed and then re-opened. (The default is NO.)

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LSN (The number of sessions)
 

- The maximum value is as defined by load parameters
- If coded as 0, the default is one of the following:
  - 6, if the load parameter RESET.VALUES=NO (default)
  - As defined by the load parameter SESSIONS, if the load parameter RESET.VALUES=YES.

 NCB\_NUM (The number of NCBs)
 

- The maximum value is as defined by the load parameters.
- If coded as 0, the default is one of the following:
  - 12, if the load parameter RESET.VALUES=NO (default)
  - As defined by the load parameter COMMANDS, if the load parameter RESET.VALUES=YES.

## NCB.RESET

### Returned Fields:

MCB\_RETCODE  
MCB\_RESERVE

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29.

Hex 32 wait

### NCB.RESET with Operating System/2 Using the Dynamic Link Routine Interface

**Command Description:** An application program executing in this environment must always issue a RESET under the following conditions:

1. As the first NCB issued by the application program
2. To recover from either a X'4x' or X'Fx' return code.

The OS/2 environment differs from the DOS environment in that RESET applies only to the application program and not for the entire PC system. An application program issues RESET to allocate resources for itself from a pool of NETBIOS resources.

When RESET is issued, only resources applicable to that application program are given back to NETBIOS. For that application program, all current NETBIOS names are deleted, all current sessions are aborted, and all outstanding NCBs are purged. After resources are freed, new resources are acquired for the application program, per the RESET request.

### Supplied Fields:

NCB\_COMMAND

NCB\_LSN (Determines whether the RESET is requesting or freeing resources.)

- If NCB\_LSN is X'00', it indicates a request for resources, per the parameters in NCB\_CALLNAME.
- If NCB\_LSN is not X'00', it indicates that all resources associated with the environment are to be freed.
  - NCB\_CALLNAME is ignored.
  - NCB\_NAME has the normal return values set (see below).

NCB\_CALLNAME is the resource request field. Any NCB\_CALLNAME areas not defined here are reserved and should be zero.

If this is a request to free resources (NCB\_LSN is not X'00'), NCB\_CALLNAME is ignored.

- REQ\_SESSIONS at NCB\_CALLNAME + 0 (one-byte field):
  - The number of sessions requested by the application program.
  - If zero, the default value is 16.
- REQ\_COMMANDS at NCB\_CALLNAME + 1 (one-byte field):
  - The number of commands requested by the application program.
  - If zero, the default value is 16.
- REQ\_NAMES at NCB\_CALLNAME + 2 (one-byte field):
  - The number of names requested by the application program. This does not include a reservation for NAME\_NUMBER\_1.
  - If zero, the default value is 8.
- REQ\_NAME\_ONE at NCB\_CALLNAME + 3 (one-byte field):
  - A request to reserve NAME\_NUMBER\_1 for this application program.
  - If 0, NAME\_NUMBER\_1 is not requested.

- If not 0, NAME\_NUMBER\_1 is desired to be reserved for this application program.

#### Returned Fields:

NCB\_NAME is used to return resource information. Areas of NCB\_NAME not defined here are reserved and set to zero.

##### 1. Returned values are resources actually obtained by the application program:

- ACT\_SESSIONS at NCB\_NAME+0 (one-byte field):  
The number of sessions obtained by the application program.
- ACT\_COMMANDS at NCB\_NAME+1 (one-byte field):  
The number of commands obtained by the application program.
- ACT\_NAMES at NCB\_NAME+2 (one-byte field):  
The number of names obtained by the application program. (Does not include NAME\_NUMBER\_1)
- ACT\_NAME\_ONE at NCB\_NAME+3 (one-byte field):
  - If zero, the application program does not have use of NAME\_NUMBER\_1.
  - If one, the application program has use of NAME\_NUMBER\_1.

##### 2. Returned values are resources defined at NETBIOS load time.

These values represent absolute maximums. The sum of all application program requests cannot exceed these values.

- LOAD\_SESSIONS at NCB\_NAME+8 (one-byte field):  
The number of sessions defined at NETBIOS load time.
- LOAD\_COMMANDS at NCB\_NAME+9 (one-byte field):  
The number of commands defined at NETBIOS load time.
- LOAD\_NAMES at NCB\_NAME+10 (one-byte field):  
The number of names defined at NETBIOS load time. This number includes NAME\_NUMBER\_1.
- LOAD\_STATIONS at NCB\_NAME+11 (one-byte field):  
The actual number of link stations obtained by NETBIOS.  
  
This may not be the number requested at load time. NETBIOS will obtain as many link stations as possible of the number requested.
- LOAD\_REMOTE\_NAMES at NCB\_NAME+14 (one-byte field):  
The number of remote names (RND) defined at NETBIOS load time.
- NCB\_NAME+15 (one-byte field) is reserved.

NCB\_RESERVE is used as in the Local Area Network Support Program.

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29.

## NCB.SEND

Hex 32

### NCB.RESET with Operating System/2 Using the Device Driver Interface

**Command Description:** An application program executing in this environment must always issue a RESET under the following conditions:

1. As the first NCB issued by the application program
2. To recover from either an X'4x' or X'Fx' return code.

The OS/2 environment differs from the DOS environment in that RESET applies only to the application program and not the entire PC system. An application program issues RESET to allocate resources for itself from a pool of NETBIOS resources.

When RESET is issued, only resources applicable to that application program are given back to NETBIOS. For that application program, all current NETBIOS names are deleted, all current sessions are aborted, and all outstanding NCBs are purged. After resources are freed, new resources are acquired for the application program, per the RESET request.

#### Supplied Fields:

NCB\_COMMAND: Same as Operating System/2 using the Dynamic Link Routine Interface

NCB\_LSN: Same as Operating System/2 using the Dynamic Link Routine Interface

NCB\_CALLNAME: Same as Operating System/2 using the Dynamic Link Routine Interface

NCB\_POST@: Reserved. Should be zero, but is not checked.

NCB\_DD\_ID:

- If this is the first RESET issued by the application program, the value must be X'0000'.
- Subsequent NCB.RESET commands issued by an application program must supply this field, as returned on the first NCB.RESET.

#### Returned Fields:

NCB\_NAME: Same as Operating System/2 using the Dynamic Link Routine Interface

NCB\_DD\_ID:

- This value must be placed in all subsequent NCBs issued by this application program

NCB\_RESERVE: Same as Operating System/2 using the Dynamic Link Routine Interface

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.SEND

Hex 14 Wait      94 No-Wait

### NCB.SEND

**Command Description:** This command sends data to the session partner as defined by the session number in the NCB\_LSN field. The data to send is in the buffer pointed to by the NCB\_BUFFER@ field.

**Command Specifics:** If the NCB.SEND timeout expires, the session is aborted and a “command timed out” status is returned. Timeout values for the NCB.SEND are associated with the session when an NCB.CALL or NCB.LISTEN was issued and cannot be specified with this command.

Message size must be between 0 and 65,535 bytes long.

If more than one session SEND type command is pending, the data is transmitted in a first-in, first-out order within a session.

If the NCB.SEND cannot be completed for any reason, the session ends abnormally and the session is dropped. This guarantees data integrity.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_LSN

**Returned Fields:**

NCB\_RETCODE  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see “NCB Return Codes Listed by Command” on page B-29 .

## **NCB.SEND.BROADCAST.DATAGRAM**

Hex 22 Wait      A2 No-Wait
<b>NCB.SEND.BROADCAST.DATAGRAM</b>

**Command Description:** This command sends a datagram message to every station that has an NCB.RECEIVE.BROADCAST.DATAGRAM command outstanding.

**Command Specifics:** If the remote station does not have an NCB.RECEIVE.BROADCAST.DATAGRAM command outstanding, it will not get the message. If a station issues an NCB.SEND.BROADCAST.DATAGRAM and has an NCB.RECEIVE.BROADCAST.DATAGRAM outstanding, the station will receive its own message. If a station has several broadcast messages pending, the next send command issued will satisfy all NCB.RECEIVE.BROADCAST commands.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_NUM

## NCB.SEND.NO.ACK

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.SEND.DATAGRAM

Hex 20 Wait    A0 No-Wait

NCB.SEND.DATAGRAM

**Command Description:** This command sends a datagram message to any unique name or group name on the network.

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LENGTH  
NCB\_BUFFER@  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_NUM  
NCB\_CALLNAME

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)

Valid Return Codes: see "NCB Return Codes Listed by Command" on page B-29 .

## NCB.SEND.NO.ACK

Hex 71 Wait    F1 No-Wait

NCB.SEND.NO.ACK

**Command Description:** The purpose of this command is to provide a send facility that does not require a data acknowledgement at the NETBIOS level. This reduces the completion time required of a SEND command.

This command sends data by the session number indicated in the local session number (LSN). The data is taken from the buffer indicated by the NCB\_BUFFER@ for the indicated number of bytes.

When a session is closed by the remote side, all NCB.SEND commands pending on the closed session will be returned with a 'Session closed' status. If a local NCB.HANG.UP command is issued with any pending NCB.SEND commands, the NCB.HANG.UP is delayed until the NCB.SEND commands are completed.

If a session aborts, a 'Session ended abnormally' status is returned. If the NCB.SEND timeout expires, the session is aborted and a 'Command timed out' status is returned. Timeout values for the NCB.SEND are associated with the session when an NCB.CALL or NCB.LISTEN was issued and cannot be specified here.

Messages are limited to a size starting with 0 and up to 65,535 bytes in length.

If more than one NCB.SEND or NCB.CHAIN.SEND is pending, the data is transmitted in a first-in, first-out (FIFO) order within a session.

**Command Specifics:** Lost data, as indicated by a X'07' return code, as well as the possible need for retransmission is the responsibility of the application program.

If the remote node cannot process the NCB.SEND.NO.ACK, it will be handled as an NCB.SEND; a DATA\_ACK must be received before the NCB.SEND.NO.ACK will complete.

It is the recommended that this command be used for transactions and not data transfers.

**Notes:**

1. Note the following on a X'07' return code.
  - The return code is generated for the node that issued NCB.SEND.NO.ACK.
  - The command terminating with the X'07' must be re-issued.
  - The return code is issued if the receiver either had no NCB.RECEIVE or NCB.RECEIVE.ANY up, or the transmitted data exceeded the length of the receive buffer.
  - Unless the user is aware of the number of NCB.SEND.NO.ACKs issued, he has no indication of how many were not completed successfully.
  - All other NCB.SEND return codes apply.
2. For data transmitted with NCB.SEND.NO.ACK, only one receive command will be satisfied. The NCB.SEND.NO.ACK data buffer must be no longer than the remote receive buffer for successful completion.

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
 NCB\_LENGTH  
 NCB\_BUFFER@  
 NCB\_POST@ (If the no-wait option is used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_LSN

**Returned Fields:**

NCB\_RETCODE  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29.

**NCB.SESSION.STATUS**

Hex 34 Wait	B4 No-Wait	
<b>NCB.SESSION.STATUS</b>		

**Command Description:** This command obtains the status of either all sessions for a local name or all sessions for all local names.



## NCB.SESSION.STATUS

**Command Specifics:** If the NCB\_NAME field contains a name, the session status will be returned for that name. If the NCB\_NAME field contains an asterisk in the first byte, the session status for all names will be returned.

The minimum valid buffer length is 4 bytes. An "illegal buffer length" status is returned if the NCB\_LENGTH field is less than 4.

A "message incomplete" status is returned if the NCB\_LENGTH field is less than the status data being generated. To obtain all status data, the buffer length must be at least 36 times the number of sessions being reported plus 4.

### Supplied Fields:

NCB\_ADPTR\_NUM (Adapter number 0 or 1)  
NCB\_LENGTH  
NCB\_BUFFER@  
NCB\_POST@ (If the no-wait option is used)  
NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
NCB\_NAME (Specify an asterisk (\*) for all names)

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (If error X'4X' or X'FX' occurs)  
NCB\_LENGTH

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

### Data Areas Returned

OFF-SET	BYTE LEN	8086 TYPE	DESCRIPTION
0	1	DB	Name number of sessions being reported; if 0, an add name is in process.
1	1	DB	Number of sessions with this name
2	1	DB	Number of NCB.RECEIVE.DATAGRAM and NCB.RECEIVE.BROADCAST.DATAGRAM commands outstanding
3	1	DB	Number of NCB.RECEIVE.ANY commands outstanding
4	1	DB	Local session number
5	1	DB	State of the session. This byte is represented as follows:  X'01' Listen outstanding X'02' CALL pending X'03' Session established X'04' HANG UP pending X'05' HANG UP complete X'06' Session aborted
6	16	DB	Local name
22	16	DB	Remote name
38	1	DB	Number of NCB.RECEIVEs outstanding
39	1	DB	Number of NCB.SEND and NCB.CHAIN.SENDs outstanding
The contents of bytes 4 - 39 (36 bytes) are repeated for every session, if adequate buffer space is available.			

## NCB.STATUS

Hex 33 Wait    B3 No-Wait

### NCB.STATUS

**Command Description:** This command requests the status of either a local or remote NETBIOS.

**Command Specifics:** The NCB\_CALLNAME field specifies which interface to get the status from. If the first byte of the field contains an asterisk (\*), the local NETBIOS status is returned. If the NCB\_CALLNAME field does not contain an asterisk, the status is returned from the node with that name. The status information is returned to the buffer defined by the NCB\_BUFFER@ field. The minimum number of bytes is 60. The maximum buffer size needed to hold the status information is 18 times the maximum number of names plus 60.

#### Supplied Fields:

NCB\_BUFFER@  
 NCB\_LENGTH  
 NCB\_CALLNAME (Local or remote or an asterisk for local)  
 NCB\_POST@ (If no-wait option used)  
 NCB\_DD\_ID (If using the Operating System/2 Device Driver Interface)  
 NCB\_ADPTR\_NUM (Adapter number 0 or 1)

#### Returned Fields:

NCB\_RETCODE  
 NCB\_LENGTH  
 NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

#### Data Areas Returned

OFF-SET	BYTE LEN	8086 TYPE	DESCRIPTION
0	6	DB	Adapter's encoded address
6	1	DB	Software release level number: X'00' for NETBIOS 1.X, X'02' for NETBIOS 2.X, and X'03' for NETBIOS 3.X
7	1	DB	Always X'00'
8	2	DW	If offset 6 indicates NETBIOS 1.X, the first byte is X'FF' and the last byte indicates the software level.  If offset 6 indicates NETBIOS 2.X, the first byte indicates the adapter type (X'FF' represents a Token-Ring Network adapter and X'FE' represents a PC Network adapter.) and the first character of the second byte indicates the type of parameters used (X'1' represents old parameters and X'2' represents new parameters.). The second character of the second byte is the second number of the software level. For example, X'11' would be NETBIOS 2.1 or X'21' would be NETBIOS 2.2.
10	2	DW	Duration of reporting period (in minutes)
12	2	DW	Number of FRMR frames received
14	2	DW	Number of FRMR frames transmitted
16	2	DW	Number of I-frames received in error
18	2	DW	Number of aborted transmissions
20	4	DW	Number of successfully transmitted packets
All counters roll over from X'F...F' to X'0...0' unless stated otherwise.			

# NCB.STATUS

OFF-SET	BYTE LEN	8086 TYPE	DESCRIPTION
24	4	DW	Number of successfully received packets
28	2	DW	Number of I-frames transmitted in error
30	2	DW	If offset 6 indicates NETBIOS 1.X, this field indicates lost data (out of SAP buffers). The counter does not increment beyond X'FFFF'.  If offset 6 indicates NETBIOS 2.X, this field indicates the number of times that a buffer was not available to service a request from a remote computer.
32	2	DW	Number of times DLC T1 timer expired
34	2	DW	Number of times DLC Ti timer expired
36	4	DD	Address of extended status information (local status only; undefined for remote status)  For OS/2 this field is reserved.
<p><b>Note:</b></p> <p>1. This is the PC system system memory location of adapter-specified status. The data is in a fixed location and an NCB.STATUS need not be reissued. The data may be interrogated, but should not be disturbed.</p> <p>This data is available only for local status. The pointer is undefined for remote status.</p> <p>The format of the data is (type DW, 2-byte fields):</p> <p><b>Bytes 0-1</b> DIR.INITIALIZE bring-up error code  <b>Bytes 2-3</b> DIR.OPEN.ADAPTER error code  <b>Bytes 4-5</b> Latest network status  <b>Bytes 6-7</b> Latest adapter check reason code  <b>Bytes 8-9</b> Latest PC-detected error (contents of AX register)  <b>Byte 10</b> Latest operational command code (4X or 5X). See note 2 below.  <b>Byte 11</b> CCB return code of the latest implicit command. See note 2 below.</p> <p>Adapter counters</p> <p><b>Bytes 12-13</b> Line errors  <b>Bytes 14-15</b> Internal errors  <b>Bytes 16-17</b> Burst errors  <b>Bytes 18-19</b> ARI/FCI delimiter  <b>Bytes 20-21</b> Abort delimiter  <b>Bytes 22-23</b> Reserved  <b>Bytes 24-25</b> Lost frame  <b>Bytes 26-27</b> Receive congestion  <b>Bytes 28-29</b> Frame copied errors  <b>Bytes 30-31</b> Frequency errors  <b>Bytes 32-33</b> Token errors  <b>Bytes 34-35</b> Reserved  <b>Bytes 36-37</b> DMA bus errors  <b>Bytes 38-39</b> DMA parity errors</p> <p>The adapter counters are valid only if no network status appendage is defined. If an appendage is defined, it is the responsibility of the user to maintain these counts. When no appendage is defined, these counters are updated when network status bit 7 (counter overflow) is reported. (The counters wrap from X'FFFF' to X'0000'.)</p> <p>These counters are the same as obtained with a DIR.READ.LOG command. Any application program that issues a DIR.READ.LOG command will cause these counters to be incorrect as to the correct total since they are reset by the DIR.READ.LOG command.</p> <p>2. When NETBIOS is initiated, a DIR.INITIALIZE and DIR.OPEN.ADAPTER are sometimes executed and a DLC.OPEN.SAP is executed. Bytes 10 and 11 are useful for determining the reason for certain '4X' return codes. Byte 10 provides the last of these CCB commands that was executed, and byte 11 contains the related CCB return code.</p>			
40	2	DW	Current number of free NCBs
42	2	DW	Configured maximum NCBs
44	2	DW	If offset 6 indicates NETBIOS 1.X, this field indicates the maximum number of NCBs (always 255).  If offset 6 indicates NETBIOS 2.X, this field indicates the maximum number of NCBs. With old parameters, the maximum number is 255. With new parameters, the maximum number is defined at load time.
All counters roll over from X'F...F' to X'0...0' unless stated otherwise.			

OFF-SET	BYTE LEN	8086 TYPE	DESCRIPTION
46	2	DW	If offset 6 indicates NETBIOS 1.X, this field indicates the number of times a local station went busy. If offset 6 indicates NETBIOS 2.X, this field indicates the number of times that a buffer was not available to transmit information.
48	2	DW	Maximum datagram packet size
50	2	DW	Number of pending sessions
<b>Note:</b> The pending session is either an NCB_CALL pending, NCB_LISTEN pending, session established, session aborted, NCB_HANG_UP pending, or NCB_HANG_UP complete.			
52	2	DW	Configured maximum pending sessions
54	2	DW	If offset 6 indicates NETBIOS 1.X, this field indicates the maximum number of pending sessions (always 254). If offset 6 indicates NETBIOS 2.X, this field indicates the maximum number of pending sessions. With old parameters, the maximum number is always 254. With new parameters, the maximum number is defined at load time.
56	2	DW	Maximum size of session data packet
58	2	DW	Number of names in the local names table
60	XXXX	DB	Local names table. Each name requires 18 bytes
<b>Note:</b> The first 16 bytes of each entry represent the name and the last 2 bytes represent the name status. The first status byte is equal to the name number (NCB_NUM). The second status byte is the status when it is masked with a X'87'. The mask is used to get the last 3 bits of the byte. The other bits are reserved and may have non-zero values.  Nrrrr000 - Trying to register a name Nrrrr100 - A registered name Nrrrr101 - A deregistered name Nrrrr110 - A detected duplicate name Nrrrr111 - A detected duplicate name with deregister pending Where: r = Reserved bit Where: N = 0 if the name is a unique name Where: N = 1 if the name is a group name  DE-REGISTERED NAME: If the name to be deleted still has an active session when a "NCB_DELETE_NAME" command is issued, the name is marked as "deregistered" and the name is removed from the local names table as soon as the session is closed.			
All counters roll over from X'F...F' to X'0...0' unless stated otherwise.			

## NCB.TRACE

Hex 79 Wait      F9 No-Wait

NCB.TRACE

**Command Description:** This command activates and de-activates a trace of all the NCBs issued to NETBIOS and some of the CCBs issued by NETBIOS, including transmits and receives.

**For Operating System/2:** The TRACE command is not available with Operating System/2. It is replaced with the Operating System/2 system trace. For a description of the Operating System/2 system trace, see Appendix E, "Operating System/2 Extended Edition 1.1 Information" on page E-1.

Receipt of NETBIOS protocol messages "remote trace off" are not supported.

### Trace Entries

The NETBIOS major trace code is X'A4'. There are two minor trace codes: X'20' indicates an Operating System/2 Device Driver Interface level trace entry and X'21' indicates an Operating System/2 Dynamic Link Routine Interface level trace entry.

## NCB.TRACE

**Command Specifics:** Both the immediate and the wait command are completed immediately and no post address is used.

To initiate the trace, the NCB\_NUM field must be set to X'FF', the NCB\_LENGTH field must be set to the length of the trace table (1024 or greater), and the NCB\_BUFFER@ must be set to the address of the trace table location in PC system memory. The NCB\_ADPTR\_NUM field must be set to a valid adapter number, but the trace will be active for both adapters. The automatic adapter open function of NETBIOS will not occur when this command is issued. This command may be issued either when an adapter is open or when it is closed.

To terminate the trace, this command is issued with the NCB\_NUM field set to '00'. In this case the NCB\_LENGTH and NCB\_BUFFER@ fields are returned values.

To terminate the trace locally and at remote adapters that have NETBIOS running and a trace active, set the NCB\_NUM field to X'01'. This will terminate the NCB.TRACE and also issue a PDT.TRACE.OFF command to terminate the adapter support software trace. It will also cause a request to be sent to NETBIOS at all remote adapters to issue NCB.TRACE (off) and PDT.TRACE.OFF commands. If the local trace is not active, the command may still be issued to terminate remote traces even though the command will be completed with a return code of X'0D'.

### Supplied Fields:

NCB\_BUFFER@ (Trace table address)  
NCB\_LENGTH (Trace table length (at least 1024))  
NCB\_NUM  
    (X'FF' = trace on,  
    X'00' = local trace off,  
    X'01' = local and all remote trace off)  
NCB\_ADPTR\_NUM (Adapter number 0 or 1)

### Returned Fields:

NCB\_RETCODE  
NCB\_RESERVE (if error X'4X' or X'FX')  
NCB\_BUFFER@ (Address of the trace table, returned when trace off option requested in NCB\_NUM field)  
NCB\_LENGTH (Length of the trace table, returned when trace off option requested in NCB\_NUM field)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .

### Trace Entry Format

Each trace entry is 32 bytes long. The initial entry is the table header. The next entry is the first trace entry. When all the trace table entry locations have been used, the table is wrapped back to the first entry overwriting previous entries. The initial, or header, entry is not overwritten.

Trace Table Header
1st Trace Entry
2nd Trace Entry
•
•
•
Nth Trace Entry

**The Trace Table Header format is:**

Byte	Usage
00-03	The address of the trace buffer (its own address)
04-05	The length of the trace buffer
06-07	The address of the first trace entry
08-15	Code change level
16-19	Trace entry counter (low word, high word)
20-21	Number of times internal response buffers were exhausted
22-23	Number of times internal transmit buffers were exhausted
24-27	Reserved
28-29	Offset of the end of the trace buffer
30-31	Offset of the next trace entry to be used

**The Trace Table Entry format**

Bytes 0 through 5 of all the trace table entries are the same format. The format of the remainder of the entry is dependent upon the content of bytes 2 (type) and 3 (modifier).

Byte	Usage
0	Adapter X'AA' = Adapter 0 X'BB' = Adapter 1 X'FF' = See note below
1	X'00' (non-3270 XMA code) or bank ID if 3270 XMA code executing
2	Type
3	Modifier of Type
4-5	Number of 100-millisecond ticks since last entry
<b>Note:</b> If this command is the first use of NETBIOS since it was loaded, the X'FF' entry will appear for the next NCB command issued. That should be the only occurrence of a X'FF' entry.	

Byte 2 Type	Byte 3 Modify	Meaning	Bytes 6-31
FF	00	NCB Issued	<b>Byte 6</b> The byte preceding the NCB
			<b>Byte 7</b> The byte following the NCB
			<b>Bytes 8-11</b> NCB Post address
			<b>Bytes 12-15</b> Address of data in PC system memory
			<b>Bytes 16-31</b> First 16 bytes of the NCB
01	01	NCB Immediate return	<b>Byte 6</b> The byte preceding the NCB
			<b>Byte 7</b> The byte following the NCB
			<b>Bytes 8-11</b> NCB Post address
			<b>Bytes 12-15</b> Address of data in PC system memory
			<b>Bytes 16-31</b> First 16 bytes of the NCB
02	02	NCB final return code	<b>Byte 6</b> The byte preceding the NCB
			<b>Byte 7</b> The byte following the NCB
			<b>Bytes 8-11</b> NCB Post address
			<b>Bytes 12-15</b> Address of data in PC system memory
			<b>Bytes 16-31</b> First 16 bytes of the NCB

# NCB.TRACE

Byte 2 Type	Byte 3 Modify	Meaning	Bytes 6-31
EE	00	Network Status	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-17</b> Network Status
	01	adapter support software internal error	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-17</b> Error Code
	02	Adapter Status	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-19</b> Pointer to the first in a queue of commands returned <b>Bytes 20-21</b> Adapter check reason code
	03	DLC Status	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-17</b> Station ID <b>Bytes 18-19</b> DLC Status <b>Bytes 20-24</b> FRMR Data <b>Byte 25</b> Access Priority <b>Bytes 26-31</b> Remote Node Address
0B	The CCB return code	Transmit I frame	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-31</b> NETBIOS header transmitted *
0D	The CCB return code	Transmit UI frame	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-31</b> NETBIOS header transmitted
28	00	Data received and processed	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-31</b> NETBIOS header received *
	FF	Data received and not processed	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-31</b> NETBIOS header received *
Any Other CCB Cmd Code	The CCB return code	Any other CCB command	<b>Bytes 6-7</b> DS <b>Bytes 8-9</b> SS <b>Bytes 10-11</b> SP <b>Bytes 12-15</b> Address of data in PC system memory <b>Bytes 16-31</b> Image of the CCB
<b>Note:</b> *If byte 20 of the trace is either X'15' or X'16', then bytes 30-31 contain the length of the user data transmitted or received.			

## NCB.UNLINK

Hex 70 Wait

NCB.UNLINK

**Command Description:** This command is provided for NETBIOS compatibility. NETBIOS treats this as a "no-operation."

**Supplied Fields:**

NCB\_ADPTR\_NUM (Adapter number 0 or 1)

**Returned Fields:**

NCB\_RETCODE

NCB\_RESERVE (If error X'4X' or X'FX' occurs)

**Valid Return Codes:** see "NCB Return Codes Listed by Command" on page B-29 .





---

## Chapter 5. The NETBIOS Frames Protocol

About This Chapter	5-3
Version Information	5-3
Assumptions	5-3
Related Documents	5-3
Terms and Definitions	5-4
The NETBIOS Service Access Point (SAP)	5-4
Addressing	5-4
Broadcast	5-4
Data Structures	5-4
Translation	5-4
NETBIOS Frame Formats and Protocols	5-5
NETBIOS Commands	5-5
General Information on Remote Name Directory (RND)	5-5
New NCB Commands	5-6
Header Format Overview	5-6
NETBIOS Header	5-6
General Format	5-6
NETBIOS Non-Session Frame Header (DLC UI-Frame)	5-6
NETBIOS Session Frame Header (DLC I-Frame)	5-7
NETBIOS Frame Header Fields	5-7
NETBIOS Frame Summary	5-8
Frames Listed Alphabetically	5-8
Frames Listed Numerically	5-8
Frames Listed by Function	5-9
Frames Listed by Transmission Type	5-10
I—Frames to Specific Address, No Broadcast	5-11
NETBIOS Frame Formats and Descriptions	5-11
NETBIOS Protocol Examples Without RND	5-37
Name Management Examples	5-38
Add a Name to the Network	5-38
Add a Name: Name Already On Network	5-39
Add a Name: Receive Multiple Responses	5-40
Remote Adapter Status Examples	5-41
Remote Adapter Status for a Name that Is Not on the Network	5-41
Remote Adapter Status for a Name that Is on the Network	5-42
Remote Adapter Status Data : Segmentation	5-43
Session Establishment Examples	5-44
Call a Name: Name Not on Network	5-44
Call a Name: Name on Network but No Listen	5-45
Call a Name: Name Found — Start Session	5-46
Session Data Transfer Examples	5-47
Send Session Data: One Send and One Receive	5-47
Send Session Data: One Send and Multiple Receives	5-48
Send Session Data: Segmentation and One Receive	5-49
Send Session Data: Segmentation and Multiple Receives	5-50
NETBIOS Protocol Examples with RND	5-51
Remote Adapter Status for a Name Not on Network or RND	5-52
Remote Adapter Status for a Name Not in RND but on the Network	5-53
Remote Adapter Status for a Remote Name that Is in RND	5-54
Session Establishment RND Examples	5-55
Call a Name: Name Not on Network or RND	5-55
Call a Name: Name Not in RND but is on Network - Start Session	5-56

NETBIOS Protocol SEND.NO.ACK Examples .....	5-57
Session Establishment Examples .....	5-58
Call a Name: Name Found - Start Session .....	5-58
Session Data Transfer Examples .....	5-59
Send Session Data: One SEND.NO.ACK and No Receive .....	5-59
Send Session Data: One SEND.NO.ACK and One Receive .....	5-60
Send Session Data: One SEND.NO.ACK and Multiple Receives .....	5-60

## Overview

This chapter describes the frame formats and protocols used by NETBIOS to communicate across one of the IBM local area networks. Each NETBIOS command results in a local action in the station and may result in the transmission of NETBIOS frames on the network.

The NETBIOS (Network Basic Input Output System) interface is a communication interface consisting of commands and their related control block structures. Communication on the network is done by using names to identify each communicating device. Two types of communication services are provided with this interface. One is a session-level service in which NETBIOS makes use of the 802.2 LLC connection-oriented services (Type 2) and the other is a datagram service in which NETBIOS makes use of the 802.2 LLC connectionless services (Type 1).

## Version Information

NETBIOS has been made available in several different products. It is provided as a part of the Local Area Network Support Program. On the diskette for the Local Area Network Support Program, the NETBIOS file is DXMT0MOD.SYS. All currently supported NETBIOS version numbers are listed below.

**Note:** The information contained in this chapter does not pertain to the NETBIOS interface that was supplied on the *original* PC Network Adapter provided by the IBM PC Network Protocol Driver.

- The Local Area Network Support Program Version 1.00 provides NETBIOS 2.0.
- The Local Area Network Support Program Version 1.01 provides NETBIOS 2.1.
- The Local Area Network Support Program Version 1.02 provides NETBIOS 2.2.
- The Local Area Network Support Program Version 1.10 provides NETBIOS 2.3.
- The Communications Manager provided with Operating System/2 Extended Edition 1.1 provides NETBIOS 3.0.

**Note:** NETBIOS 2.X includes all the changes from earlier releases of NETBIOS.

## Assumptions

It is assumed that the reader is familiar with the information contained in Chapter 4, "NETBIOS." The reader should also be familiar with general network architecture.

In all cases, except where explicitly stated otherwise, a function that applies to some specific version also applies to later (higher) versions.

## Related Documents

The following Local Area Network standards may be helpful.

- ISO 8802-2 Local Area Networks Logical Link Control
- ISO 8802-5 Local Area Networks Token Ring Access Method

---

## Terms and Definitions

Refer to the glossary in the back of this manual for a complete terminology list.

### The NETBIOS Service Access Point (SAP)

NETBIOS opens and uses SAP X'F0'. This SAP value is reserved by IBM for use by the NETBIOS function.

### Addressing

NETBIOS sets the functional address X'00000080' on the adapter. This functional address is used by NETBIOS when broadcasting frames. This functional address bit is reserved by IBM for use by the NETBIOS function.

### Broadcast

NETBIOS makes use of the broadcast capability of the network. In all cases but one, frames are sent as single-route broadcast: the broadcast bit and the single-route broadcast bit in the routing control field are both set to 1 (one). In one case a frame is sent as a general broadcast: the broadcast bit in the routing control field is set to 1 (one) and the single-route broadcast bit is set to 0 (zero).

**Single-Route Broadcast:** Single-route broadcast means that only one frame is to be delivered to each ring. This can be accomplished by designating selected bridges to forward single-route broadcast frames. Frames sent using single-route broadcast will have the broadcast bit and the single-route broadcast bit in the routing control field both set to 1 (one).

**General Broadcast:** General broadcast means that all bridges will forward such frames. A frame sent using general broadcast will have the broadcast bit in the routing control field set to 1 (one) and the single-route broadcast bit set to 0 (zero).

### Data Structures

The data definition of each NETBIOS frame is shown using the INTEL data types for the various fields. The DB (define byte) type is used to define single byte fields and byte string fields. The DW (define word) type is used primarily to define 2-byte numeric fields.

**Note:** It is important to realize that fields defined as DW will reside in memory as byte reversed. When a frame is transmitted, these fields will be transmitted as they appear in memory, byte reversed.

### Translation

NETBIOS assumes no translation for any character strings used in the NETBIOS frames (for example, names). These strings are viewed simply as binary data.

---

## NETBIOS Frame Formats and Protocols

### NETBIOS Commands

Below is a list of the NETBIOS commands. Some of these command names are referred to in the description of some of the NETBIOS frames.

ADD.GROUP.NAME  
ADD.NAME  
CALL  
CANCEL  
CHAIN.SEND  
CHAIN.SEND.NO.ACK  
DELETE.NAME  
FIND.NAME  
HANGUP  
LAN.STATUS.ALERT  
LISTEN  
RECEIVE  
RECEIVE.ANY  
RECEIVE.BROADCAST.DATAGRAM  
RECEIVE.DATAGRAM  
RESET  
SEND  
SEND.BROADCAST.DATAGRAM  
SEND.DATAGRAM  
SEND.NO.ACK  
SESSION.STATUS  
STATUS  
TRACE  
UNLINK

---

### General Information on Remote Name Directory (RND)

Remote Name Directory functions are implemented in NETBIOS Version 2.1.

To reduce the unnecessary interrupts to NETBIOS nodes, which are caused by broadcast frames to the NETBIOS functional address, Remote Name Directory (RND) function is used to send the frame to a specific node, whenever possible.

The local station locates a remote name (on-ring only if RND is used) by issuing a NAME\_QUERY on-ring once, and issues a NAME\_QUERY off-ring per transmit count and transmit timeout values which may be defined by configuration parameters. The range for NCB.TRANSMIT.COUNT is 1-10 and the default is 6. The range for NCB.TRANSMIT.TIMEOUT is 1/2 second to 10 seconds and the default is 1/2 second.

When RND is used, after the local station has located a remote name, the remote node address is saved and subsequent messages to that name will be to a specific node rather than a broadcast to all nodes. This is true for CALLS and STATUS QUERIES ( also, SEND DATAGRAMS if REMOTE.DATAGRAM.CONTROL is used).

**Note:** When the RND function is being used, a duplicate network name will not be detected by a CALL function unless it is the first time that the remote (CALL) name has been used.

## New NCB Commands

Two new commands: NCB.SEND.NO.ACK and NCB.CHAIN.SEND.NO.ACK are introduced in NETBIOS 2.2 version. The purpose of the NCB.SEND.NO.ACK/NCB.CHAIN.SEND.NO.ACK commands is to provide a SEND facility that does not require NETBIOS to transmit a data acknowledgement. This is to reduce the completion time of a "SEND" type command. There is also a new NCB.LAN.STATUS.ALERT command. This LAN status command allows application programs to be alerted of temporary ring error conditions that last more than one minute.

## Header Format Overview

All NETBIOS frames contain a header, which immediately follows the Data Link Control (DLC) header:



\*\*\*\*\*

### Notes:

1. With respect to the DLC frame, the NETBIOS header is data.
2. 'USER DATA' is applicable only on certain messages.

## NETBIOS Header

In this section the various NETBIOS frame headers will be described. See the *IBM Token-Ring Network Architecture Reference* for a complete description of the LAN header and DLC headers.

### General Format

NETBIOS Header Length	X'FFFF'	Command	Optional Data1	Optional Data2	Xmit/Resp Correlator	Dest ID	Source ID
-----------------------	---------	---------	----------------	----------------	----------------------	---------	-----------

### NETBIOS Non-Session Frame Header (DLC UI-Frame)

These frames are transmitted using DLC connectionless services to either a functional address or a specific address.

0	2	4	5	6	8	12	28	44
NETBIOS Header Length	X'FFFF'	Command	Optional Data1	Optional Data2	Xmit/Resp Correlator	Dest Name	Source Name	

## NETBIOS Session Frame Header (DLC I-Frame)

These frames are transmitted using DLC connection-oriented services to a specific address.

0	2	4	5	6	8	12	13	14
NETBIOS Header Length	X'FFFF'	Command	Optional Data1	Optional Data2	Xmit/Resp Correlator	Dest Num	Source Num	

### NETBIOS Frame Header Fields

#### *Bytes Meaning*

- 0-1** NETBIOS Header Length – Length of NETBIOS header (including the length field)
- 2-3** X'FFFF' – A delimiter that indicates that subsequent data is destined for the NETBIOS function.
- 4** Command – Specific protocol command – indicates the type or function of the frame. The command codes are partitioned as follows:
  - X'00' – X'13' are UI frames
  - X'14' – X'1F' are I frames
- 5** Data1 – 1 byte of optional data per a specific command
- 6-7** Data2 – 2 bytes of optional data per a specific command.
- 8-11** Correlator – One or two numbers in the range X'0001' – X'FFFF'. Used to associate received responses with transmitted requests.
  - Transmit Correlator: The value returned in a response to a given query (the value was received as the response correlator).
  - Response Correlator: The value expected (in the transmit correlator field) when the response to that message is received.

**Note:** Commands X'00' – X'07' will be discarded by the sending NETBIOS, upon return to the station. For example, when an ADD\_NAME\_QUERY is received by the station that transmitted it, the message is discarded.

Remaining bytes depend upon message type.

- Non-Session frame (DLC UI frame):

#### *Bytes Meaning*

**12-27** Destination Name – a 16-character destination name.

**28-43** Source Name – a 16-character source name.

- Session frame (DLC I frame):

#### *Bytes Meaning*

**12** Destination Number – a 1-byte destination session number.

**13** Source Number – a 1-byte source session number.



## NETBIOS Frame Summary

Each NETBIOS frame has been given a name that is suggestive of its use in implementing the various NETBIOS commands.

### Frames Listed Alphabetically

The following is an alphabetical listing of all the NETBIOS frames.

FRAME NAME	CODE	FUNCTION
ADD_GROUP_NAME_QUERY	X'00'	Check for duplicate group name on network
ADD_NAME_QUERY	X'01'	Check for duplicate name on network
ADD_NAME_RESPONSE	X'0D'	Negative response: add name is duplicate
DATA_ACK	X'14'	DATA_ONLY_LAST acknowledgment
DATA_FIRST_MIDDLE	X'15'	Session data message – first or middle frame
DATAGRAM	X'08'	Application – generated datagram
DATAGRAM_BROADCAST	X'09'	Application – generated broadcast datagram
DATA_ONLY_LAST	X'16'	Session data message – only or last frame
NAME_IN_CONFLICT	X'02'	Duplicate names detected
NAME_QUERY	X'0A'	Request to locate a name on the network
NAME_RECOGNIZED	X'0E'	Name Recognized: NAME_QUERY response
NO_RECEIVE	X'1A'	No receive command to hold received data
RECEIVE_CONTINUE	X'1C'	Indicates receive outstanding
RECEIVE_OUTSTANDING	X'1B'	Re – transmit last data – receive command up
SESSION_ALIVE	X'1F'	Verify session is still active
SESSION_CONFIRM	X'17'	SESSION_INITIALIZE acknowledgment
SESSION_END	X'18'	Session termination
SESSION_INITIALIZE	X'19'	A session has been set up
STATUS_QUERY	X'03'	Request remote node status
STATUS_RESPONSE	X'0F'	Remote node status information, STATUS_QUERY response
TERMINATE_TRACE	X'07'	Terminate traces at remote nodes
TERMINATE_TRACE	X'13'	Terminate traces at local and remote nodes

### Frames Listed Numerically

The NETBIOS frames are listed below according to the command value.

FRAME NAME	CODE	FUNCTION
ADD_GROUP_NAME_QUERY	X'00'	Check for duplicate group name on network
ADD_NAME_QUERY	X'01'	Check for duplicate name on network
NAME_IN_CONFLICT	X'02'	Duplicate names detected
STATUS_QUERY	X'03'	Request remote node status
TERMINATE_TRACE	X'07'	Terminate traces at remote nodes
DATAGRAM	X'08'	Application – generated datagram
DATAGRAM_BROADCAST	X'09'	Application – generated broadcast datagram
NAME_QUERY	X'0A'	Request to locate a name on the network

FRAME NAME	CODE	FUNCTION
ADD_NAME_RESPONSE	X'0D'	Negative response: add name is duplicate
NAME_RECOGNIZED	X'0E'	Name Recognized: NAME_QUERY response
STATUS_RESPONSE	X'0F'	Remote node status information, STATUS_QUERY response
TERMINATE_TRACE	X'13'	Terminate traces at local and remote nodes
DATA_ACK	X'14'	DATA_ONLY_LAST acknowledgment
DATA_FIRST_MIDDLE	X'15'	Session data message – first or middle frame
DATA_ONLY_LAST	X'16'	Session data message – only or last frame
SESSION_CONFIRM	X'17'	SESSION_INITIALIZE acknowledgment
SESSION_END	X'18'	Session termination
SESSION_INITIALIZE	X'19'	A session has been set up
NO_RECEIVE	X'1A'	No receive command to hold received data
RECEIVE_OUTSTANDING	X'1B'	Re – transmit last data – receive command up
RECEIVE_CONTINUE	X'1C'	Indicates receive outstanding
SESSION_ALIVE	X'1F'	Verify session is still active

## Frames Listed by Function

The NETBIOS frames are grouped below by function.

**Name Management Frames:** The following frames are used to provide name management functions.

FRAME NAME	CODE	FUNCTION
ADD_GROUP_NAME_QUERY	X'00'	Check for duplicate group name on network
ADD_NAME_QUERY	X'01'	Check for duplicate name on network
ADD_NAME_RESPONSE	X'0D'	Negative response: add name is duplicate
NAME_IN_CONFLICT	X'02'	Duplicate names detected

**Session Establishment and Termination Frames:** The following frames are used to establish, maintain and terminate sessions.

FRAME NAME	CODE	FUNCTION
NAME_QUERY	X'0A'	Request to locate a name on the network
NAME_RECOGNIZED	X'0E'	Name Recognized: NAME_QUERY response
SESSION_ALIVE	X'1F'	Verify session is still active
SESSION_CONFIRM	X'17'	SESSION_INITIALIZE acknowledgment
SESSION_END	X'18'	Session termination
SESSION_INITIALIZE	X'19'	A session has been set up

**Data Transfer Frames:** The following frames are used to transfer both session and non – session data.

FRAME NAME	CODE	FUNCTION
DATA_ACK	X'14'	DATA_ONLY_LAST acknowledgment
DATA_FIRST_MIDDLE	X'15'	Session data message – first or middle frame
DATAGRAM	X'08'	Application – generated datagram
DATAGRAM_BROADCAST	X'09'	Application – generated broadcast datagram
DATA_ONLY_LAST	X'16'	Session data message – only or last frame
NO_RECEIVE	X'1A'	No receive command to hold received data
RECEIVE_CONTINUE	X'1C'	Indicates receive outstanding
RECEIVE_OUTSTANDING	X'1B'	Re-transmit last data – receive command up

### Additional Frames

FRAME NAME	CODE	FUNCTION
STATUS_QUERY	X'03'	Request remote node status
STATUS_RESPONSE	X'0F'	Remote node status information, STATUS_QUERY response
TERMINATE_TRACE	X'07'	Terminate traces at remote nodes
TERMINATE_TRACE	X'13'	Terminate traces at local and remote nodes

### Frames Listed by Transmission Type

The NETBIOS frames are grouped below according to the type of transmission used for each frame.

#### UI – Frames to Functional Address, Single-Route Broadcast

FRAME NAME	CODE	FUNCTION
ADD_GROUP_NAME_QUERY	X'00'	Check for duplicate group name on network
ADD_NAME_QUERY	X'01'	Check for duplicate name on network
DATAGRAM *	X'08'	Application – generated datagram
DATAGRAM_BROADCAST	X'09'	Application – generated broadcast datagram
NAME_IN_CONFLICT	X'02'	Duplicate names detected
NAME_QUERY *	X'0A'	Request to locate a name on the network
STATUS_QUERY *	X'03'	Request remote node status
TERMINATE_TRACE	X'07'	Terminate traces at remote nodes
TERMINATE_TRACE	X'13'	Terminate traces at local and remote nodes
* If remote name directory function is used, UI frames are sent to the specific address with no broadcast.		

## UI – Frames to Specific Address, No Broadcast

FRAME NAME	CODE	FUNCTION
ADD_NAME_RESPONSE	X'0D'	Negative response: add name is duplicate
STATUS_RESPONSE	X'0F'	Remote node status information, STATUS_QUERY response

## UI – Frames to Specific Address, General Broadcast

FRAME NAME	CODE	FUNCTION
NAME_RECOGNIZED	X'0E'	Name Recognized: NAME_QUERY response

## I – Frames to Specific Address, No Broadcast

FRAME NAME	CODE	FUNCTION
DATA_ACK	X'14'	DATA_ONLY_LAST acknowledgment
DATA_FIRST_MIDDLE	X'15'	Session data message – first or middle frame
DATA_ONLY_LAST	X'16'	Session data message – only or last frame
NO_RECEIVE	X'1A'	No receive command to hold received data
RECEIVE_OUTSTANDING	X'1B'	Re-transmit last data – receive command up
SESSION_CONFIRM	X'17'	SESSION_INITIALIZE acknowledgment
SESSION_END	X'18'	Session termination
SESSION_INITIALIZE	X'19'	A session has been set up
RECEIVE_CONTINUE	X'1C'	Indicates receive outstanding
SESSION_ALIVE	X'1F'	Verify session is still active

# NETBIOS Frame Formats and Descriptions

This section describes each NETBIOS frame. The frame descriptions are organized by command value.

Hex 00

## ADD\_GROUP\_NAME\_QUERY

**Initiation:** This frame is initiated by an ADD.GROUP.NAME command.

**Function:** To verify that the group name to be added does not already exist as a unique name within the network.

**Transmission:** This frame is transmitted as a UI – frame to the NETBIOS functional address using single-route broadcast.

**Responses:** No responses within the system timeout period indicates that the name is not already in use as a unique name. An ADD\_NAME\_RESPONSE indicates that the name is already in use as a unique name and cannot be added as a group name.

**Note:** An ADD\_NAME\_RESPONSE from more than one remote adapter will cause the local adapter to transmit a NAME\_IN\_CONFLICT to all remote adapters.

**Retries:** The number of times the frame is transmitted can be defined by the NCB.TRANSMIT.COUNT parameter. The range for this count is 1 – 10 and the default is 6.

**Note:** The frame will be sent this number of times whether a response is received or not.

**Timeouts:** The timeout value between retransmissions of this frame may be defined by the NCB.TRANSMIT.TIMEOUT parameter. The range is 1/2 second to 10 seconds and the default is 1/2 second.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'00'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NAME	16	DB	reserved
28	SOURCE NAME	16	DB	Group name to be added

---

**Response Correlator**

**Explanation:** This is a 2-byte correlator generated by the sender and used to correlate any responses to this query. It is returned in any ADD\_NAME\_RESPONSE in the transmit correlator field.

---

**Source Name**

**Explanation:** This is a 16-byte field that contains the group name to be added. This is the NCB\_NAME field in the ADD.GROUP.NAME command.

Hex 01

**ADD\_NAME\_QUERY**

**Initiation:** This frame is initiated by an ADD.NAME command.

**Function:** To verify that the name to be added is a unique name within the network.

**Transmission:** This frame is transmitted as a UI – frame to the NETBIOS functional address using single-route broadcast.

**Responses:** No responses within the system timeout period indicates that the name is a unique name. An ADD\_NAME\_RESPONSE indicates that the name is not unique and can not be added.

**Note:** An ADD\_NAME\_RESPONSE from more than one remote adapter will cause the local adapter to transmit a NAME\_IN\_CONFLICT to all remote adapters.

**Retries:** The number of times the frame is transmitted can be defined by the NCB.TRANSMIT.COUNT parameter. The range for this count is 1 – 10 and the default is 6.

**Note:** The frame will be sent this number of times whether a response is received or not.

**Timeouts:** The timeout value between transmissions of this frame may be defined by the NCB.TRANSMIT.TIMEOUT parameter. The range is 1/2 second to 10 seconds and the default is 1/2 second.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'01'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NAME	16	DB	reserved
28	SOURCE NAME	16	DB	Name to be added

---

**Destination Name**

**Explanation:** This is a 2-byte correlator generated by the sender and used to correlate any responses to this query. It is returned in any ADD\_NAME\_RESPONSE in the transmit correlator field.

---

**Source Name**

**Explanation:** This is a 16-byte field that contains the name to be added. This is the NCB\_NAME field in the ADD.NAME command.

---

**Hex 02**

**NAME\_IN\_CONFLICT**

**Initiation:** This frame is sent if it is detected that more than one adapter has the same unique name registered or that a name is registered as both a group name and a unique name in the network (for example, if an ADD\_NAME\_RESPONSE is received from more than one node following transmission of an ADD\_NAME\_QUERY or ADD\_GROUP\_NAME\_QUERY or if a NAME\_RECOGNIZED is received from more than one adapter following transmission of a NAME\_QUERY).

**Function:** To indicate to all nodes that the specified name has been detected as being registered at more than one node.

**Transmission:** This frame is transmitted as a UI-frame to the NETBIOS functional address using single-route broadcast.

**Responses:** None

**Retries:** The number of times the frame is transmitted can be defined by the NCB.TRANSMIT.COUNT parameter. The range for this count is 1 – 10 and the default is 6.

**Timeouts:** The timeout value between retransmissions of this frame may be defined by the NCB.TRANSMIT.TIMEOUT parameter. The range is 1/2 second to 10 seconds and the default is 1/2 second.

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'02'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	Name in conflict
28	SOURCE NAME	16	DB	Sending node NAME_NUMBER_1

### Destination Name

**Explanation:** This is a 16-byte field containing the name that was determined to be in conflict.

### Source Name

**Explanation:** This is a 16-byte field that contains the NAME\_NUMBER\_1, which consists of 10 bytes of binary zeros followed by the 6-byte permanently encoded address for this adapter.

### Hex 03

#### STATUS\_QUERY

**Initiation:** This frame is sent due to a STATUS command.

**Function:** To request remote adapter status.

**Transmission:** This frame is transmitted as a UI – frame to the NETBIOS functional address using single-route broadcast.

**Responses:** No response within the system timeout period will cause the command to time out. A STATUS\_RESPONSE is used to return the remote adapter status.

If a status request is sent to a NETBIOS 2.1 station from a NETBIOS 2.1 station, and the remote adapter cannot transmit all the status data at once, upon receiving the STATUS\_RESPONSE, an additional STATUS\_QUERY frame is transmitted as a UI frame to the specific address of the node that sent the STATUS\_RESPONSE. This process is repeated until all the data are received or the application program buffer is filled up.

**Retries:** The number of times the frame is transmitted can be defined by the NCB.TRANSMIT.COUNT parameter. The range for this count is 1 – 10 and the default is 6.

**Timeouts:** The timeout value between transmissions of this frame may be defined by the NCB.TRANSMIT.TIMEOUT parameter. The range is 1/2 second to 10 seconds and the default is 1/2 second.

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'03'
5	DATA1	1	DB	X'nn'
6	DATA2	2	DW	Length of user's status buffer
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NAME	16	DB	Name of receiver
28	SOURCE NAME	16	DB	Sending node NAME_NUMBER_1

---

### Data1

**Explanation:** This is a 1-byte field generated by the sender.

X'nn' indicates the status request, where:

**X'nn' = 0**

Status request 1.x or 2.0

**X'nn' = 1**

Initial status request, NETBIOS 2.1

**X'nn' > 1**

NETBIOS 2.1 request for more names

**X'nn' > 1**

Indicates the total number of names received so far (names received in the first response and all additional name responses).

**Note:** X'nn' > 1 status request is not transmitted unless the remote station is NETBIOS 2.1.

---

### Data2

**Explanation:** This is a 2-byte field containing the length of the user's buffer area as defined by the NCB\_LENGTH field in the STATUS command.

---

### Response Correlator

**Explanation:** This is a 2-byte correlator generated by the sender and used to correlate any responses to this query. It is returned in any STATUS\_RESPONSE in the transmit correlator field.

---

### Destination Name

**Explanation:** This is a 16-byte field containing the name on the network for which the status is being requested. This is the name from the NCB\_CALL\_NAME field in the STATUS command.

---

### Source Name

**Explanation:** This is a 16-byte field which contains the NAME\_NUMBER\_1, consisting of 10 bytes of binary zeros followed by the 6-byte permanent adapter address for this adapter.



**Hex 07****TERMINATE\_TRACE**

**Initiation:** This frame is sent due to a TRACE command with the 'remote trace off' option set (NCB\_NUM = X'01').

**Function:** To terminate the trace at any remote nodes which had activated a trace with the TRACE command.

**Transmission:** This frame is transmitted as a UI-frame to the NETBIOS functional address using single-route broadcast.

**Responses:** None

**Retries:** None

**Timeouts:** None

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'EFFF'
4	COMMAND	1	DB	X'07'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	reserved
28	SOURCE NAME	16	DB	reserved

**Hex 08****DATAGRAM**

**Initiation:** This frame is sent due to a SEND.DATAGRAM command.

**Function:** To transmit a user specified datagram to a name.

**Transmission:** This frame is transmitted as a UI-frame to the NETBIOS functional address using single-route broadcast.

**Responses:** None

**Retries:** None

**Timeouts:** None

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'08'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	Name of receiver
28	SOURCE NAME	16	DB	Name of sender
44	USER DATA	nn	DB	Datagram

**Destination Name**

**Explanation:** This is a 16-byte field containing the name on the network to which the datagram is being sent. This is the NCB\_CALLNAME field in the SEND.DATAGRAM command.

**Source Name**

**Explanation:** This is a 16-byte field that contains the name of the originator of the datagram. This is the name associated with the NCB\_NUM field in the SEND.DATAGRAM command.

**User Data**

**Explanation:** This is the datagram that the user provided with the SEND.DATAGRAM command.

**Note:** If the adapter is opened by an application program other than NETBIOS, and the size of the transmit buffer is not sufficient for the size of the datagram, the datagram will be truncated with no indication given.

**Hex 09**

**DATAGRAM\_BROADCAST**

**Initiation:** This frame is sent due to a SEND.BROADCAST.DATAGRAM command.

**Function:** To transmit a user specified datagram to all names on the network.

**Transmission:** This frame is transmitted as a UI-frame to the NETBIOS functional address using single-route broadcast.

**Responses:** None

**Retries:** None

**Timeouts:** None

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'09'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	reserved
28	SOURCE NAME	16	DB	Name of sender
44	USER DATA	nn	DB	Datagram

---

### Source Name

**Explanation:** This is a 16-byte field that contains the name of the originator of the datagram. This is the name associated with the NCB\_NUM field in the SEND.BROADCAST.DATAGRAM command.

---

### User Data

**Explanation:** This is the datagram that the user provided with the SEND.BROADCAST.DATAGRAM command.

**Note:** If the adapter is opened by an application program other than NETBIOS, and the size of the transmit buffer is not sufficient for the size of the datagram, the datagram will be truncated with no indication given.

---

### Hex 0A

NAME\_QUERY

**Initiation:** This frame is sent due to a CALL command or a FIND.NAME command.

**Function:** To request the remote node to establish a session (CALL) or to locate a name on the network (FIND.NAME).

**Transmission:** This frame is transmitted as a UI – frame to the NETBIOS functional address using single-route broadcast.

**Responses:** No response within the system timeout period indicates the name is not on the network and the command gets an appropriate return code (X'14' for CALL and X'05' for FIND.NAME). A NAME\_RECOGNIZED response will indicate if a session can be established in the case of a CALL command or indicate the location of a name in the case of the FIND.NAME command. A NAME\_RECOGNIZED response for a CALL command will terminate (with a valid LISTEN on the remote side) the response timer and the NAME\_QUERY will not be retried.

**Retries:** The number of times the frame may be transmitted can be defined by the NCB.TRANSMIT.COUNT parameter. The range is 1 – 10 and the default is 6.

**Note:** If a NAME\_RECOGNIZED with no LISTEN response is received for a CALL command, after transmitting NAME\_QUERY frame NCB.TRANSMIT.COUNT times, NAME\_QUERY process is retried one more time.

**Timeouts:** The timeout value between transmissions of this frame may be defined by the NCB.TRANSMIT.TIMEOUT parameter. The range is 1/2 second to 10 seconds and the default is 1/2 second. The command timeout for the FIND.NAME case is NCB.TRANSMIT.COUNT times NCB.TRANSMIT.TIMEOUT – the default is 3 seconds.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'0A'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	X'ttss'
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NAME	16	DB	Name of receiver
28	SOURCE NAME	16	DB	Name of sender

**Data2**

**Explanation:** 'tt' indicates the type of name that is issuing the CALL command, where:

- X'00' unique name
- X'01' group name

'ss' indicates the local session number that will be assigned to refer to this session if the CALL completes. The number is assigned in round-robin fashion (1 – 254). A value of zero (0) is not a valid session number and indicates a FIND.NAME request.

**Response Correlator**

**Explanation:** This is a 2-byte field containing a correlator that is generated by the sender and used to correlate any responses with this query. The value is returned in the NAME\_RECOGNIZED response in the transmit correlator field.

**Destination Name**

**Explanation:** This is a 16-byte field containing the name on the network that is being called or located. This is the name in the NCB\_CALLNAME field of the command.

**Source Name**

**Explanation:** This is a 16-byte field containing the local name that is issuing the CALL command. This is the name in the NCB\_NAME field of the CALL command. This field is not used if this frame is for a FIND.NAME command ('ss' in DATA2 is zero).

Hex 0D

## ADD\_NAME\_RESPONSE

**Initiation:** This frame is sent as a response to an ADD\_NAME\_QUERY or an ADD\_GROUP\_NAME\_QUERY frame.

**Function:** To indicate that the identified name in the query frame is already in use. This response is generated to an ADD\_NAME\_QUERY if the name is already registered as either a group name or a unique name. This response is generated to an ADD\_GROUP\_NAME\_QUERY only if the name is already registered as a unique name.

**Transmission:** This frame is transmitted as a UI-frame to the specific address of the node that sent the query.

**Responses:** None

**Retries:** None

**Timeouts:** None

### Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'0D'
5	DATA1	1	DB	0 = add name not in process, 1 = add name in process
6	DATA2	2	DW	0 = unique name, 1 = group name
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	Name to be added
28	SOURCE NAME	16	DB	Name to be added

### Transmit Correlator

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the ADD\_NAME\_QUERY or ADD\_GROUP\_NAME\_QUERY. This is used to correlate the response with the original query at the originator.

### Destination Name

**Explanation:** This is a 16-byte field containing the name that the originator wanted to add.

### Source Name

**Explanation:** This is a 16-byte field containing the name that the originator wanted to add.

Hex 0E

## NAME\_RECOGNIZED

**Initiation:** This frame is sent as a response to a NAME\_QUERY frame.

**Function:** Used to indicate whether a session can be established with the queried name (CALL) or used to indicate the location of a name (FIND.NAME).

**Transmission:** This frame is transmitted as a UI – frame to the specific address of the node that sent the NAME\_QUERY using general broadcast.

**Responses:** If this NAME\_RECOGNIZED frame indicates that no session is to be established, then no response is expected. If this NAME\_RECOGNIZED frame indicates that a session is to be established, then a SESSION\_INITIALIZE frame is expected.

**Retries:** If multiple NAME\_QUERY frames are received from the same node, then a NAME\_RECOGNIZED frame is sent for each NAME\_QUERY received. This could happen if the originator of the NAME\_QUERY timed out and retried the NAME\_QUERY before the NAME\_RECOGNIZED was received by the originator.

**Timeouts:** If the NAME\_RECOGNIZED indicates that a session is to be established, then the SESSION\_INITIALIZE frame is expected within a timeout period determined by configuration parameters (NCB.TRANSMIT.COUNT times NCB.TRANSMIT.TIMEOUT). The default is 3 seconds.

### Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'0E'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	X'ttss'
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NAME	16	DB	Name of receiver
28	SOURCE NAME	16	DB	Name of sender

### Data2

#### Explanation:

The 'tt' in DATA2 indicates the type of name as it is registered locally, where:

**X'00'** is a unique name

**X'01'** is a group name

The 'ss' in DATA2 indicates the state of the queried name, where:

**X'00'** means no LISTEN command is outstanding for this name or this is a FIND.NAME response.

**X'FF'** means a LISTEN command is outstanding for this name, but there are insufficient resources to establish a session at this time.

**X'01' – X'FE'** means a locally assigned session number that will identify this session when the call process completes.

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the NAME\_QUERY frames. This is used to correlate the response with the original query at the originator.

---

**Response Correlator**

**Explanation:** This is a 2-byte field containing a correlator that is generated by the sender. This is used to correlate the SESSION\_INITIALIZE frame with this NAME\_RECOGNIZED frame. It will be returned in the transmit correlator field of the SESSION\_INITIALIZE frame. If this NAME\_RECOGNIZED frame indicates that no session is to be established, then this field is unused.

---

**Destination Name**

**Explanation:** This is a 16-byte field containing the name at the originator that issued the NAME\_QUERY.

---

**Source Name**

**Explanation:** This is a 16-byte field containing the name that was being queried.

---

Hex 0F

**STATUS\_RESPONSE**

**Initiation:** This frame is sent as a response to a STATUS\_QUERY frame.

**Function:** Used to return the status information of the adapter.

**Transmission:** This frame is transmitted as a UI – frame to the specific address of the node that sent the STATUS\_QUERY.

**Responses:** None

**Retries:** None

**Timeouts:** None

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X' EFFF'
4	COMMAND	1	DB	X'0F'
5	DATA1	1	DB	X'nn'
6	DATA2	2	DW	X'xybbbbbbbbbbbb'
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	Receiver's NAME_NUMBER_1
28	SOURCE NAME	16	DB	Name of sender

---

**Data1**

**Explanation:** This is a 1-byte field generated by the sender.

X'nn' indicates the status response, where:

**X'nn' = 0**

Status response 1.x or 2.0

**X'nn' > 0**

NETBIOS 2.1 status response

**X'nn' > 0**

Indicates the total number of names sent so far (names sent in the first response and additional name responses, including this one).

---

**Data2**

**Explanation:**

'x' 1 indicates the length of the status data exceeds the maximum UI-frame size (the data is truncated).

'y' 1 indicates the length of the status data exceeds the size of the user's buffer (sent in the STATUS\_QUERY).

'bbb...' indicates the length of the status data sent (interpreted as a 2-byte (DW) field with the 2 high order bits zeroed).

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the STATUS\_QUERY frames. This is used to correlate the response with the original query at the originator.

---

**Destination Name**

**Explanation:** This is a 16-byte field containing the NAME\_NUMBER\_1 of the originator of the STATUS\_QUERY. This was the source name field of the STATUS\_QUERY frame and is 10 bytes of binary zeros followed by the 6-byte permanent adapter address.

---

**Source Name**

**Explanation:** This is a 16-byte field containing the name that was being queried. This was the destination name in the STATUS\_QUERY.

---

Hex 13

TERMINATE\_TRACE

**Initiation:** This frame must be generated by some application program other than NETBIOS. This frame is never sent by NETBIOS, but it will react to the receipt of a X'13'.

**Function:** To terminate the traces at any remote nodes which had activated a trace with the TRACE command.

**Transmission:** This frame is transmitted as a UI-frame to the NETBIOS functional address using single-route broadcast.

**Responses:** None



**Retries:** None

**Timeouts:** None

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'002C'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'13'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NAME	16	DB	reserved
28	SOURCE NAME	16	DB	reserved

Hex 14

**DATA\_ACK**

**Initiation:** This frame is a response to a DATA\_ONLY\_LAST frame.

**Function:** Used to indicate a positive acknowledgment to a received DATA\_ONLY\_LAST frame.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the node that sent the DATA\_ONLY\_LAST frame.

**Responses:** None

**Retries:** All retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'14'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the DATA\_ONLY\_LAST frame. This is used to correlate this DATA\_ACK frame with the data frame at the originator.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session. This value was in the source number field of the DATA\_ONLY\_LAST frame.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

**Hex 15**

<b>DATA_FIRST_MIDDLE</b>
--------------------------

**Initiation:** This frame is transmitted as a result of a SEND, CHAIN.SEND, SEND.NO.ACK, or CHAIN.SEND.NO.ACK command.

**Function:** To transfer a user message across a session, where the message has to be segmented because the size of the message exceeds the transmit buffer size or the maximum frame size that was established for this session during session initialization. This is one frame of a message, but not the last one.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** If a RECEIVE\_CONTINUE frame is requested (on the first DATA\_FIRST\_MIDDLE) then a RECEIVE\_CONTINUE frame will be returned if all the data was accepted and only partially filled the RECEIVE buffer. If the data completely fills or exceeds the RECEIVE buffer, then a NO\_RECEIVE frame will be returned.

**Note:** In NETBIOS 2.X version or higher, if the data completely fills or exceeds the RECEIVE buffer, and if another RECEIVE is outstanding for the same session, instead of returning NO\_RECEIVE/RECEIVE\_OUTSTANDING sequence, a RECEIVE\_OUTSTANDING frame is returned to acknowledge the last byte received and also to indicate the RECEIVE is available to receive more data.

In NETBIOS 2.2 version, a NO\_RECEIVE frame will be returned, if the NCB.SEND.NO.ACK data or NCB.CHAIN.SEND.NO.ACK data was not received at all or only partially received by the remote application program.

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer. At the NETBIOS level there may be a SEND Timeout in effect for the entire message transfer, that was specified in the NCB\_STO field of the CALL command.

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'15'
5	DATA1	1	DB	B'rrrrrxy' where: x = NO.ACK indicator (NETBIOS 2.2) y = RECEIVE_CONTINUE request
6	DATA2	2	DW	re-synch indicator
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number
14	USER DATA	nn	DB	Message from SEND

---

### Data1

#### Explanation:

'x' is the NO.ACK indicator (NETBIOS 2.2)

x = 0: Earlier versions of NETBIOS 2.2

x = 1: Frame sent via NCB.SEND.NO.ACK or NCB.CHAIN.SEND.NO.ACK (NETBIOS 2.2)

'y' indicates RECEIVE\_CONTINUE request

y = 0: No RECEIVE\_CONTINUE requested

y = 1: Request for a RECEIVE\_CONTINUE

'r' is a reserved bit, always zero

This is a 1-byte field where a X'01' indicates that a RECEIVE\_CONTINUE frame is being requested and a X'00' indicates no request. The RECEIVE\_CONTINUE is requested on the first DATA\_FIRST\_MIDDLE frame if this is the first message on this session or if any frames in the previous message encountered a NO\_RECEIVE response. If all segments of a message are sent without receiving a NO\_RECEIVE frame, then the first segment of the next message will not have this request bit set. The RECEIVE\_CONTINUE frame indicates that all of the data in this DATA\_FIRST\_MIDDLE frame was accepted. When the first DATA\_FIRST\_MIDDLE is sent (with the request indicator on), subsequent segments are not sent until a response is received (RECEIVE\_CONTINUE or RECEIVE\_OUTSTANDING). This is to avoid sending all message segments only to have them discarded at the remote node if there is no RECEIVE outstanding or the RECEIVE buffer cannot accept the entire message.

---

### Data2

**Explanation:** This is a 2-byte field where a value of X'0001' indicates that this is the *first* DATA\_FIRST\_MIDDLE following receipt of a RECEIVE\_OUTSTANDING from the remote node. A RECEIVE\_OUTSTANDING is sent to indicate the ability to receive more data following a NO\_RECEIVE. The indicator is set so the remote node can re-synch on the first valid DATA\_FIRST\_MIDDLE following a NO\_RECEIVE/RECEIVE\_OUTSTANDING sequence. After the first DATA\_FIRST\_MIDDLE is responded to, all remaining segments are transmitted in succession.

---

**Response Correlator**

**Explanation:** This is a 2-byte field containing a correlator that is generated by the sender. This is used to correlate the response frame with this frame. The correlator will be returned in the transmit correlator field of the RECEIVE\_CONTINUE frame.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

**User Data**

**Explanation:** This is a segmented portion of the message that the user provided.

Hex 16

**DATA\_ONLY\_LAST**

**Initiation:** This frame is transmitted as a result of a SEND, CHAIN.SEND, SEND.NO.ACK., or CHAIN.SEND.NO.ACK command.

**Function:** To transfer a user message across a session. This is either the only frame in this message or this is the last segment of a segmented message.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** Responses could be one of the following to acknowledge receipt of this frame or portion of this frame

DATA\_ACK  
NO\_RECEIVE  
RECEIVE\_OUTSTANDING

A DATA\_ACK will acknowledge the receipt of this frame. A NO\_RECEIVE frame will be returned if the data exceeds the RECEIVE buffer.

**Note:** In NETBIOS 2.X version or higher, if the data exceeds the RECEIVE buffer, and if another RECEIVE is outstanding for the same session, instead of returning the NO\_RECEIVE/RECEIVE\_OUTSTANDING sequence, a RECEIVE\_OUTSTANDING frame is returned to acknowledge the last byte received and also to indicate that the RECEIVE is available to receive more data.

In NETBIOS 2.2 version, a NO\_RECEIVE frame will be returned, if the NCB.SEND.NO.ACK data or NCB.CHAIN.SEND.NO.ACK data was not received at all or only partially received by the remote application program.

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer. At the NETBIOS level there may be a SEND Timeout in effect for the entire message transfer, that was specified in the NCB\_STO field of the CALL command.

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'16'
5	DATA1	1	DB	B'rrrrrrxr' where x = NO.ACK indicator (NETBIOS 2.2)
6	DATA2	2	DW	re-synch indicator
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number
14	USER DATA	nn	DB	Message from SEND

---

### Data1

#### Explanation:

'x' is the NO.ACK indicator (NETBIOS 2.2)

x = 0: Earlier versions of NETBIOS 2.2

x = 1: Frame sent via NCB.SEND.NO.ACK or NCB.CHAIN.SEND.NO.ACK (NETBIOS 2.2)

'r' is a reserved bit, always zero

---

### Data2

**Explanation:** This is a 2-byte field where a value of X'0001' indicates that this is the first DATA\_ONLY\_LAST following receipt of a RECEIVE\_OUTSTANDING (otherwise this field is X'0000'). The RECEIVE\_OUTSTANDING is sent to indicate the ability to receive more data following a NO\_RECEIVE.

---

### Destination Number

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

### Source Number

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

### User Data

**Explanation:** This is the last or only segment of the message that the user provided.

**Hex 17****SESSION\_CONFIRM**

**Initiation:** This frame is transmitted in response to a SESSION\_INITIALIZE frame.

**Function:** To indicate a positive acknowledgment to a SESSION\_INITIALIZE to complete the session establishment process.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** None

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'EFFF'
4	COMMAND	1	DB	X'17'
5	DATA1	1	DB	B'yrrrrrx'
6	DATA2	2	DW	Max data receive size
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	X'nnnn' = Session init xmit correlator
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number

**Data1**

**Explanation:** This is a 1-byte field generated by the sender.

rrrrrr reserved bits, always zero

x = 0 indicates NETBIOS 1.x

x = 1 indicates NETBIOS 2.0 or higher version

y = 0 Earlier versions of NETBIOS 2.2

y = 1 Flag to indicate the ability to handle SEND.NO.ACK  
or CHAIN.SEND.NO.ACK

**Data2**

**Explanation:** This is a 2-byte field containing the maximum size user data in any frame that this node wants to receive on this session. NETBIOS determines this value based on the size and number of receive buffers available in the adapter such that this value is approximately one-half of the receive buffer space. The remote partner will limit the size of the user data in frames transmitted over this session to this size or the size available in its transmit buffer (DHB), whichever is smaller. This is to avoid having the receipt of one session frame consume all of the available receive buffers in the adapter. NETBIOS takes into account the maximum size message that bridges in the path will forward. It will never send a frame larger than the bridge will forward.

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the SESSION\_INITIALIZE frame. This is used to correlate this response with the SESSION\_INITIALIZE frame at the originator.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

Hex 18

SESSION\_END

**Initiation:** This frame is transmitted as a result of a HANGUP command, a SEND command that timed out, or some abnormal condition.

**Function:** To indicate the termination of a session to the remote partner.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** None

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'EFFF'
4	COMMAND	1	DB	X'18'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	Termination indicator
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	Remote session number
28	SOURCE NUMBER	1	DB	Local session number

---

**Data2**

**Explanation:** This is a 2-byte field used to indicate the type of termination, where X'0000' indicates a normal session end (as a result of a HANGUP command) and X'0001' indicates an abnormal session end (typically a SEND command has timed out).

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

**Hex 19****SESSION\_INITIALIZE**

---

**Initiation:** This frame is transmitted in response to a NAME\_RECOGNIZED frame that indicates that a session is to be established.

**Function:** To indicate that a session has been established.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** A SESSION\_CONFIRM is expected to indicate acknowledgment of the session establishment.

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer. At the NETBIOS level, the SESSION\_CONFIRM is expected within a timeout period of NCB.TRANSMIT.COUNT times NCB.TRANSMIT.TIMEOUT plus 1 second per bridge. The default is 3 seconds.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X' EFFF'
4	COMMAND	1	DB	X'19'
5	DATA1	1	DB	B'zrrrxyy'
6	DATA2	2	DW	Max data receive size
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	X'nnnn'
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number



---

**Data1**

**Explanation:** This is a 1-byte field generated by the sender.

rrrr reserved bits, always zero  
xxx indicates the largest frame value as seen by the MAC layer ( LF field in the routing information)  
y = 0 indicates NETBIOS 1.x  
y = 1 indicates NETBIOS 2.0 or higher version  
  
z = 0 Earlier versions of NETBIOS 2.2  
z = 1 Flag to indicate the ability to handle SEND.NO.ACK or CHAIN.SEND.NO.ACK

---

**Data2**

**Explanation:** This is a 2-byte field containing the maximum size user data in any frame that this node wants to receive on this session. NETBIOS determines this value based on the size and number of receive buffers available in the adapter such that this value is approximately one-half of the receive buffer space. The remote partner will limit the size of the user data in frames transmitted over this session to this size or the size available in its transmit buffer (DHB), whichever is smaller. This is to avoid having the receipt of one session frame consume all of the available receive buffers in the adapter. NETBIOS takes into account the maximum size message that bridges in the path will forward. It will never send a frame larger than the bridge will forward.

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the NAME\_RECOGNIZED frame. This is used to correlate this frame with the NAME\_RECOGNIZED frame at the remote node.

---

**Response Correlator**

**Explanation:** This is a 2-byte field containing a correlator that is generated by the sender. This is used to correlate the SESSION\_CONFIRM frame with this SESSION\_INITIALIZE frame. It will be returned in the transmit correlator field of the SESSION\_CONFIRM frame.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

**Hex 1A****NO\_RECEIVE**

**Initiation:** This frame is transmitted as a result of receiving a DATA\_ONLY\_LAST or a DATA\_FIRST\_MIDDLE frame.

**Function:** To acknowledge receipt of session data, and indicate how much of the data was accepted. This frame is sent when the RECEIVE buffer is filled. Upon receipt of this frame at the remote node, the remote node discontinues sending session data on this session until the local node transmits a RECEIVE\_OUTSTANDING frame. Any session data that continues to arrive at the local node will be ignored until a DATA\_FIRST\_MIDDLE or DATA\_ONLY\_LAST frame arrives with the re-synch indicator set (which means the local node has sent a RECEIVE\_OUTSTANDING frame).

**Notes:**

1. When a data frame is received, if there is no RECEIVE command outstanding then nothing is transmitted to the sender. When a RECEIVE command is issued, then a RECEIVE\_OUTSTANDING frame is transmitted.
2. In NETBIOS 2.X version or higher, if the data completely fills or exceeds the RECEIVE buffer, and if another RECEIVE is outstanding for the same session, instead of returning NO\_RECEIVE/RECEIVE\_OUTSTANDING sequence, a RECEIVE\_OUTSTANDING frame is returned to acknowledge the last byte received and also to indicate the RECEIVE is available to receive more data.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** None

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'1A'
5	DATA1	1	DB	B'rrrrrrxr'
6	DATA2	2	DW	Number of data bytes accepted
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number

---

**Data1****Explanation:**

- r : Reserved bit, always zero.
- x = 0: Earlier versions of NETBIOS 2.2
- x = 1: Flag to indicate that NCB.SEND.NO.ACK data or NCB.CHAIN.SEND.NO.ACK data either not received or partially received ( NETBIOS 2.2 ).

---

**Data2**

**Explanation:** This is a 2-byte field containing the number of bytes of user data that were accepted. When the remote node is informed to resume transmitting session data (via a RECEIVE\_OUTSTANDING frame) it will resume with the next byte following the last acknowledged byte.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

**Hex 1B**

<b>RECEIVE_OUTSTANDING</b>
----------------------------

**Initiation:** This frame is transmitted following transmission of a NO\_RECEIVE if a RECEIVE command is or becomes available.

**Function:** To inform the remote session partner that there is a RECEIVE outstanding and that the remote partner should resume sending session data. The remote node resumes with the first data byte following the last acknowledged byte indicated in the NO\_RECEIVE frame. The remote node indicates the first session data frame following the RECEIVE\_OUTSTANDING frame by setting the re-synch indicator.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** After transmitting this frame, the local node is expecting a DATA\_ONLY\_LAST or DATA\_FIRST\_MIDDLE frame with the re-synch indicator set.

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

## Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'EFFE'
4	COMMAND	1	DB	X'1B'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	Number of data bytes accepted
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number

### DATA2

**Explanation:** This field is defined for NETBIOS 2.1. For earlier versions, this is a reserved field.

This is a 2-byte field containing the number of bytes of user data that were accepted. When the remote node is informed to resume transmitting session data (via a RECEIVE\_OUTSTANDING frame) it will resume with the next byte following the last acknowledged byte.

### Destination Number

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

### Source Number

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

### Hex 1C

RECEIVE\_CONTINUE

**Initiation:** This frame is transmitted in response to a DATA\_FIRST\_MIDDLE that had the RECEIVE\_CONTINUE request bit set.

**Function:** To indicate that there is a RECEIVE command outstanding on this session. It also acknowledges receipt of all the session data in the DATA\_FIRST\_MIDDLE (a NO\_RECEIVE would be transmitted if the data filled or exceeded the RECEIVE buffer).

**Note:** In NETBIOS 2.X version or higher, if the data completely fills or exceeds the RECEIVE buffer, and if another RECEIVE is outstanding for the same session, instead of returning NO\_RECEIVE/RECEIVE\_OUTSTANDING sequence, a RECEIVE\_OUTSTANDING frame is returned to acknowledge the last byte received and also to indicate the RECEIVE is available to receive more data.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** None

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer.

**Frame Format:**

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'1C'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	X'nnnn'
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	Remote session number
13	SOURCE NUMBER	1	DB	Local session number

---

**Transmit Correlator**

**Explanation:** This is a 2-byte field containing the correlator that was in the response correlator field of the DATA\_FIRST\_MIDDLE frame. This is used to correlate this frame with the DATA\_FIRST\_MIDDLE frame at the originator.

---

**Destination Number**

**Explanation:** This is a 1-byte field containing the session number that was assigned at the remote node during session initialization to identify this session.

---

**Source Number**

**Explanation:** This is a 1-byte field containing the local session number that was assigned during session initialization to identify this session.

---

**Hex 1F**

**SESSION\_ALIVE**

**Initiation:** This frame is transmitted due to the periodic expiration of a link inactivity timer.

**Function:** To periodically determine, in the absence of session data, if the links to any remote nodes for which sessions are established are still available. This is to prevent keeping resources allocated when a link may have become unavailable without indication. A bad return code on the transmit of this frame will cause the user to be informed of abnormal session termination and the resources for every session on this link will be released. Receipt of this frame is ignored.

**Transmission:** This frame is transmitted as an I-frame to the specific address of the remote session partner.

**Responses:** None

**Retries:** Since this is an I-frame, all retries are handled by the 802.2 LLC layer.

**Timeouts:** Retry timeouts are handled by the 802.2 LLC layer. At the NETBIOS level, the periodic link inactivity timer is 30 seconds.

### Frame Format:

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	LENGTH	2	DW	X'000E'
2	DELIMITER	2	DW	X'FFFF'
4	COMMAND	1	DB	X'1F'
5	DATA1	1	DB	reserved
6	DATA2	2	DW	reserved
8	XMIT CORRELATOR	2	DW	reserved
10	RSP CORRELATOR	2	DW	reserved
12	DEST NUMBER	1	DB	reserved
13	SOURCE NUMBER	1	DB	reserved

---

## NETBIOS Protocol Examples Without RND

Following are NETBIOS 2.1 examples of typical protocol scenarios including name management, session establishment, and session data transfer. The direction of a frame on the network is indicated by an arrow. The frame type is above the arrow and the contents of the frame are below the arrow. If a frame is repeated, the data appears only on the first instance of the frame.

### Notes:

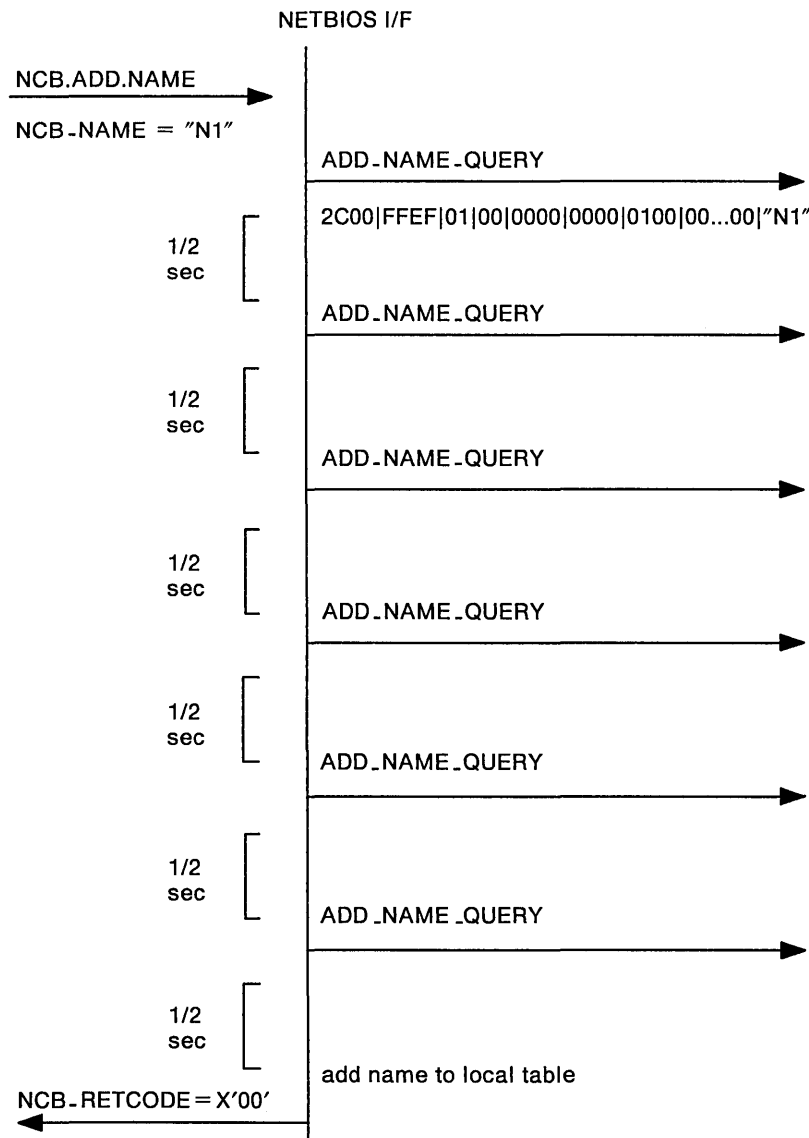
1. The numeric values shown in the various examples indicate *the true number*; in a trace, the high-order and low-order bytes may be reversed because of 8086 architecture.
2. The use of 'rrrr' within certain fields of the frame indicates a reserved field.

## Name Management Examples

Following are scenarios that show the adding of a name to the network.

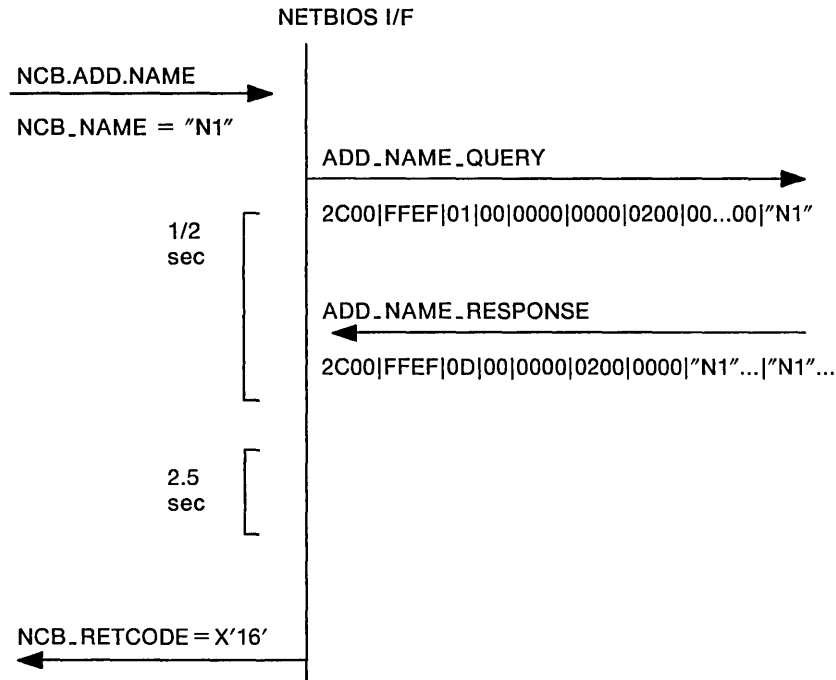
### Add a Name to the Network

The application program attempts to add a unique name to the network. The ADD\_NAME\_QUERY is sent at 1/2 second intervals for 6 times (by default). If no response is received in that period of time, the name is assumed to be unique and is added to the name table. The application program NCB gets a return code of X'00'.



## Add a Name: Name Already On Network

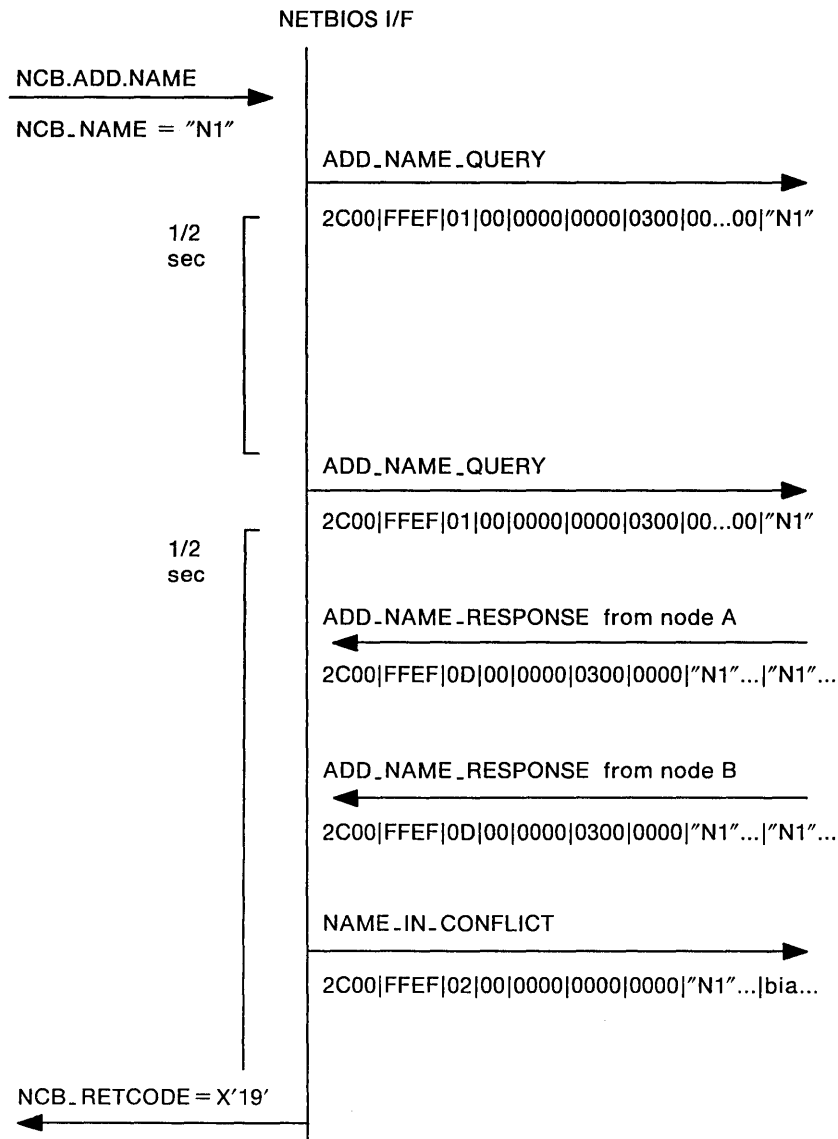
The application program attempts to add a unique name to the network and the name is already registered at some remote node. The remote node responds to the ADD\_NAME\_QUERY with an ADD\_NAME\_RESPONSE. The originating node stops transmitting ADD\_NAME\_QUERY when a response is received and waits for the total timeout period checking for responses from other nodes. If no other nodes respond, the NCB gets a return code of X'16' (name in use on remote NETBIOS).





## Add a Name: Receive Multiple Responses

The application program attempts to add a unique name to the network and the name is registered at more than one remote node (this is an error situation). The originating node stops transmitting ADD\_NAME\_QUERY when a response is received and waits for the total timeout period, checking for responses from other nodes. When more than one response is received from a different node, the NAME\_IN\_CONFLICT is detected by the originating node and sends a NAME\_IN\_CONFLICT to all remote nodes and no longer waits for the timeout period. The NCB gets a return code of X'19' (name conflict detected).

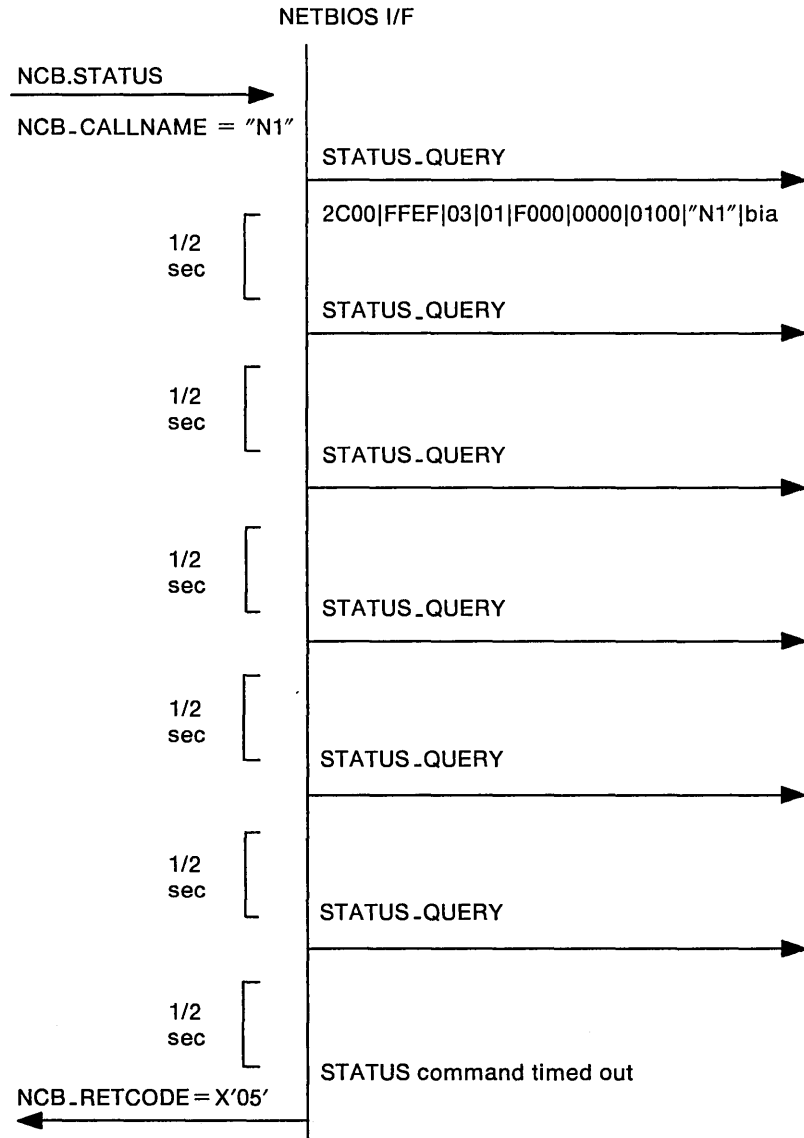


## Remote Adapter Status Examples

Following are examples of remote status requests.

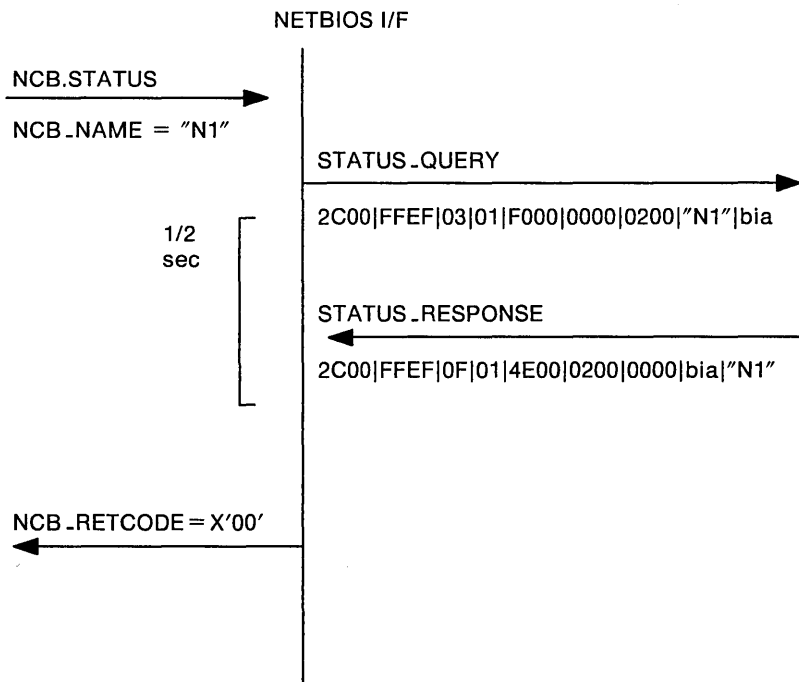
### Remote Adapter Status for a Name that Is Not on the Network

The application program attempts to request the remote status for a name that is not on the network. STATUS\_QUERY is sent at 1/2 second intervals for 6 times (by default). If a STATUS\_RESPONSE is not received the application program NCB gets a return code of X'05' (command timed out).



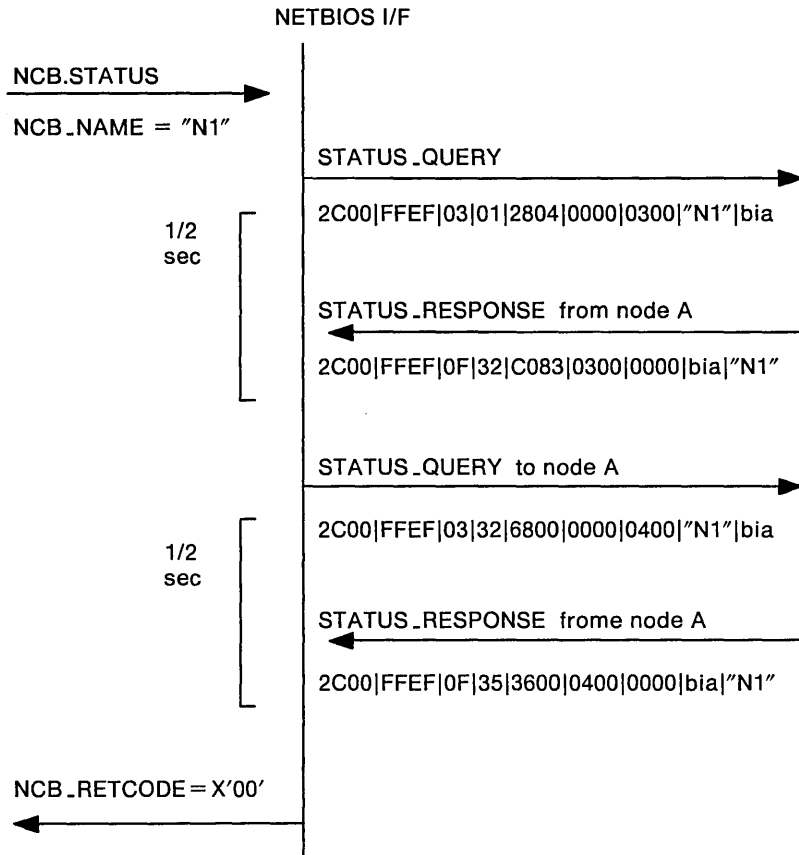
## Remote Adapter Status for a Name that Is on the Network

The application program attempts to request the remote status for a name that is on the network. STATUS\_QUERY is sent at 1/2 second intervals for 6 times (by default). If a STATUS\_RESPONSE is received the application program NCB gets a return code of X'00'. It is assumed that the application program buffer is big enough to hold the status data.



## Remote Adapter Status Data : Segmentation

The application program attempts to request the remote status for a name that is on the network. STATUS\_QUERY is sent at 1/2 second intervals for 6 times (by default). The example shown below describes how the status data are segmented. It is assumed that the remote station has 63 names and the maximum UI frame that the remote station can transmit is 962 bytes. The application program puts up a buffer for 1064 bytes. The application program NCB gets a return code of X'00'.

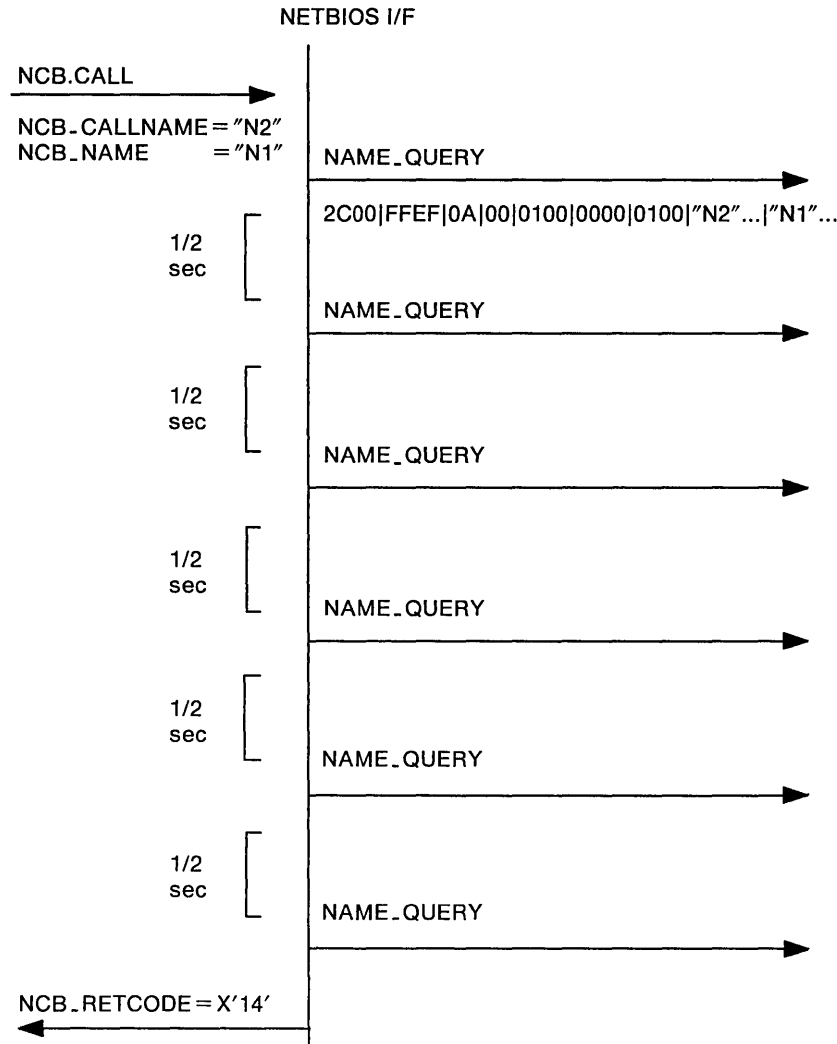


## Session Establishment Examples

Following are scenarios that show session establishment.

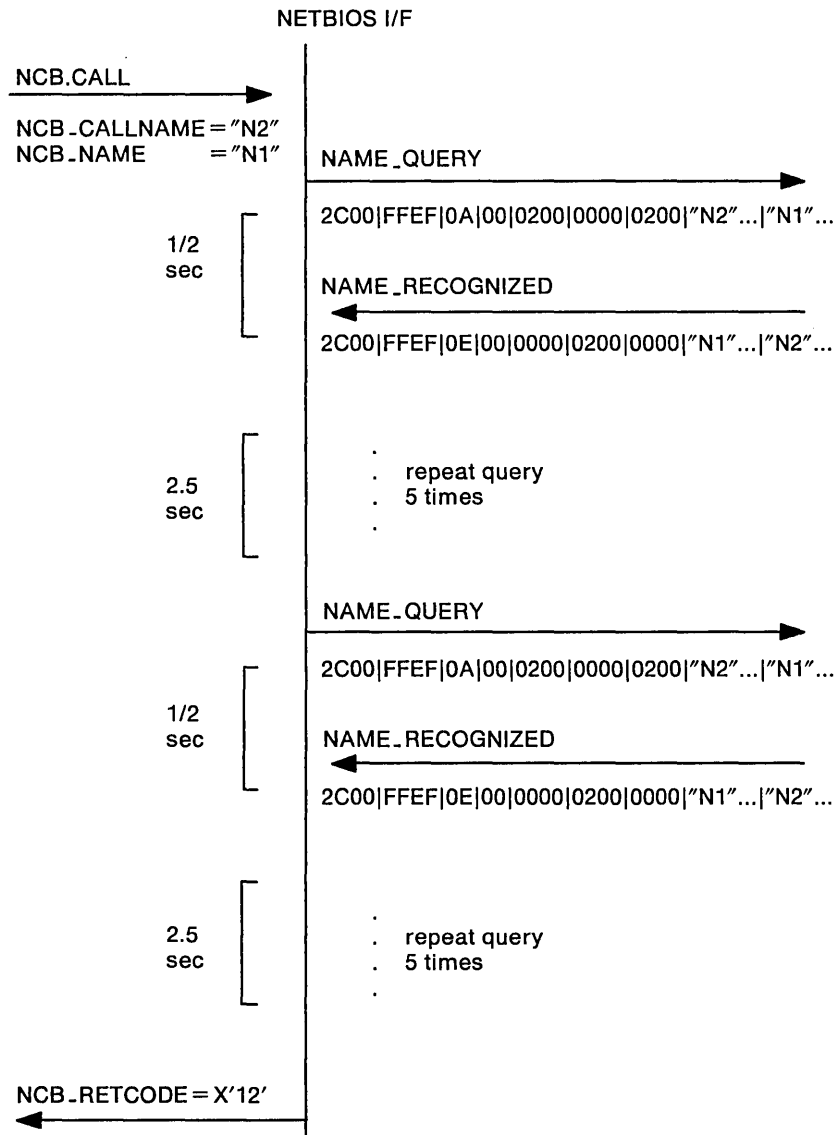
### Call a Name: Name Not on Network

The application program attempts to call a name on the network that does not exist. The NAME\_QUERY is sent at 1/2 second intervals for 6 times (by default). If a NAME\_RECOGNIZED is not received, the application program NCB gets a return code of X'14' (cannot find name or no answer).



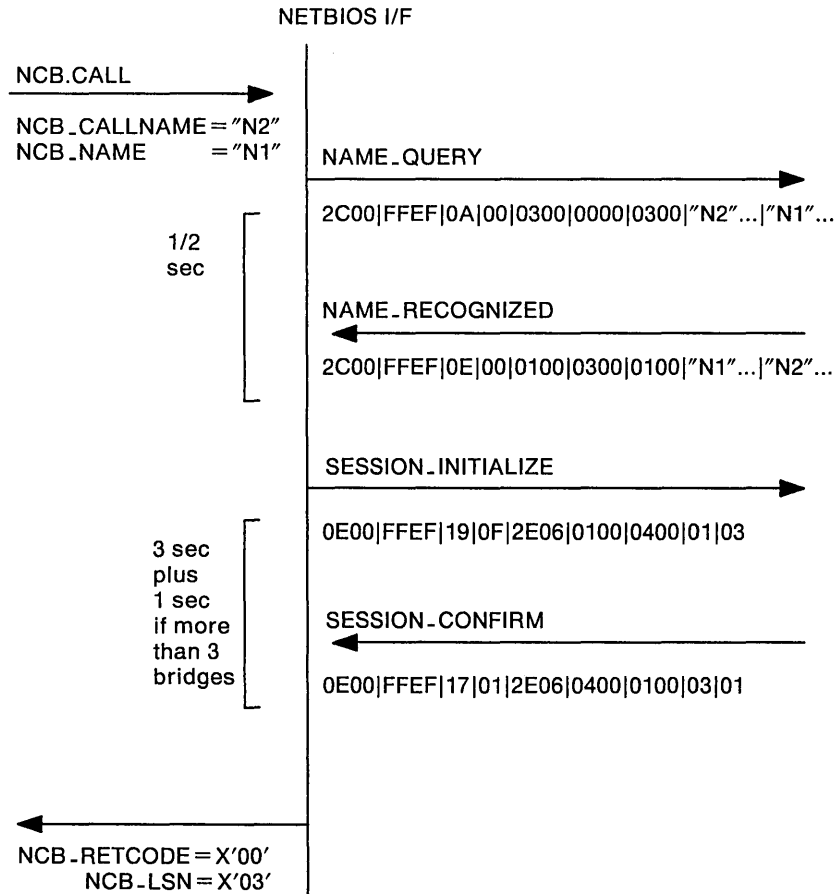
## Call a Name: Name on Network but No Listen

The application program attempts to call a name on the network that is registered but there is no LISTEN outstanding for that name. The NAME\_QUERY is sent at 1/2 second intervals for 6 times (by default). If a NAME\_RECOGNIZED is received which indicates no LISTEN, the query cycle is started over. If a good NAME\_RECOGNIZED is still not received, the application program NCB gets a return code of X'12' (session open rejected).



## Call a Name: Name Found – Start Session

The application program uses name "N1" and issues a call to name "N2" that is registered at a remote node that has a LISTEN outstanding for that name. The NAME\_QUERY and NAME\_RECOGNIZED contain the respective session numbers. Receipt of the NAME\_RECOGNIZED causes the originating node to send a SESSION\_INITIALIZE (as an I-frame) and run a timer waiting for a SESSION\_CONFIRM (sent as an I-frame). The SESSION\_INITIALIZE and SESSION\_CONFIRM each indicate the maximum amount of user data that they are prepared to receive on this session. The remote partner will then limit the size of the user data in frames transmitted over this session to this size or the size available in its transmit buffer (DHB), whichever is smaller. Receipt of the SESSION\_CONFIRM will complete the call and set the NCB\_RETCODE to X'00' and NCB\_LSN to the locally assigned session number.

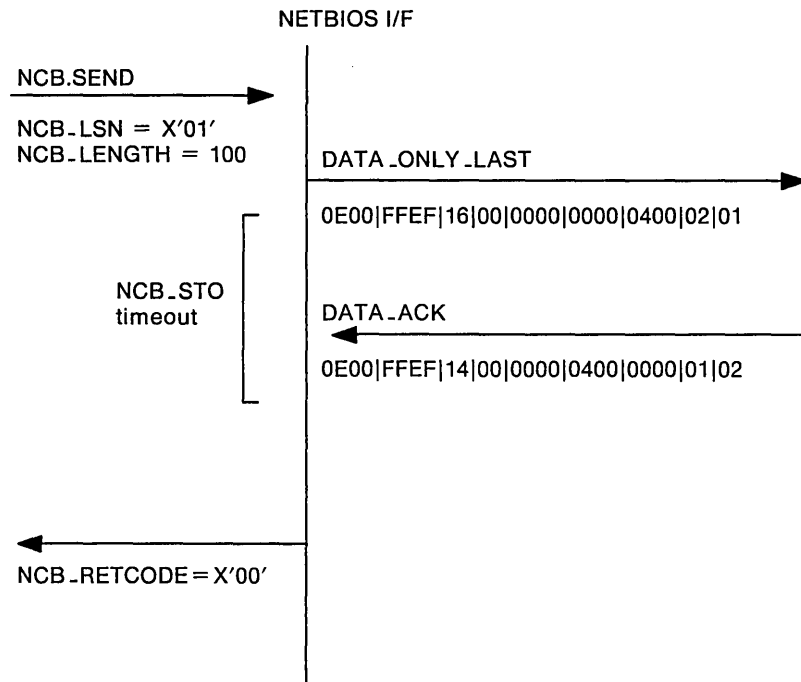


## Session Data Transfer Examples

Following are scenarios that show data transfer across a session including segmentation at the sender and multiple receives at the receiver.

### Send Session Data: One Send and One Receive

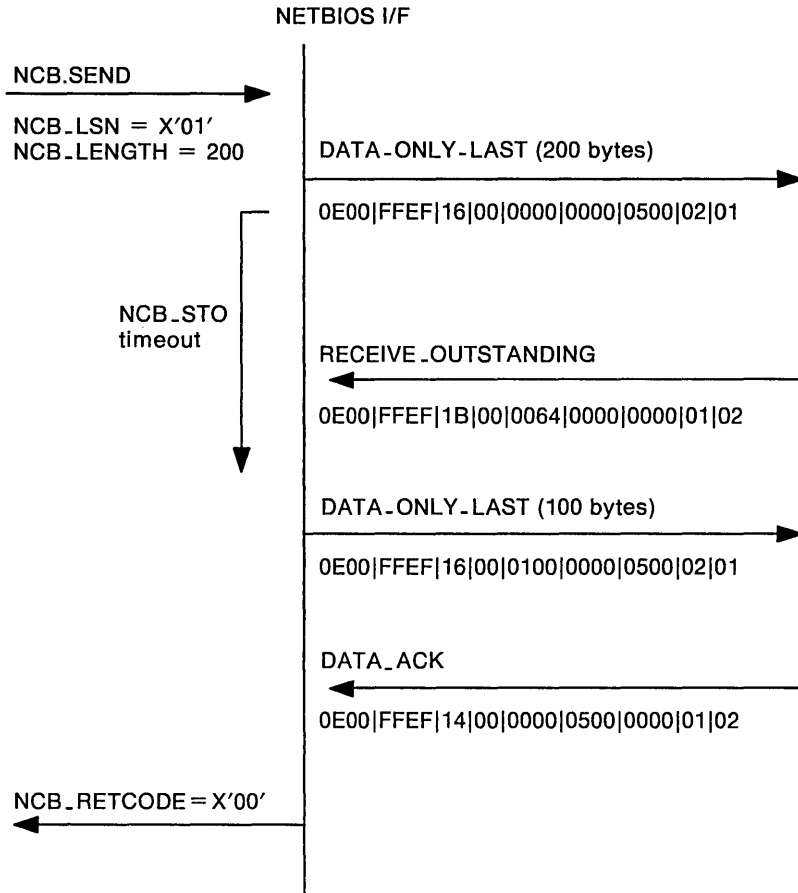
The session has been established as above. The local node issues a SEND command for 100 bytes and the remote node has a corresponding 100-byte RECEIVE outstanding. The local node sends a DATA\_ONLY\_LAST and the remote node responds with a DATA\_ACK.





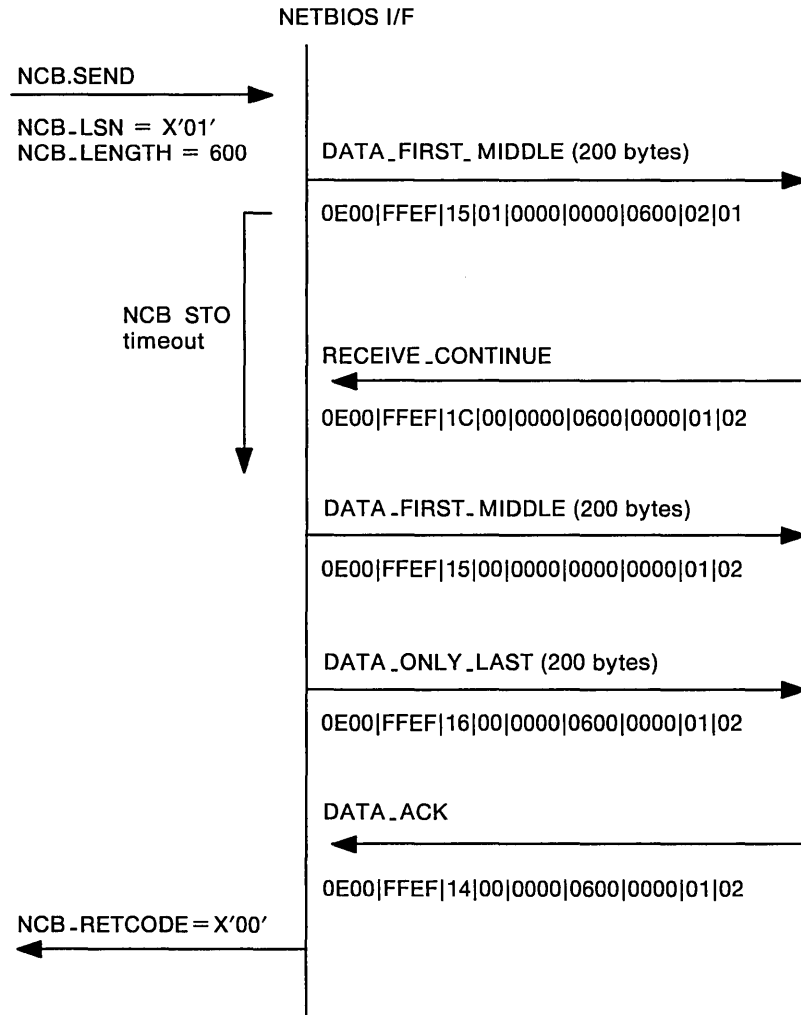
## Send Session Data: One Send and Multiple Receives

The local node sends a 200-byte message over the session to the remote node. The remote node has two 100-byte RECEIVES outstanding. The remote node sends a RECEIVE\_OUTSTANDING, acknowledging 100 bytes and indicating another RECEIVE is outstanding. The local node re-sends until the remote has acknowledged all the data.



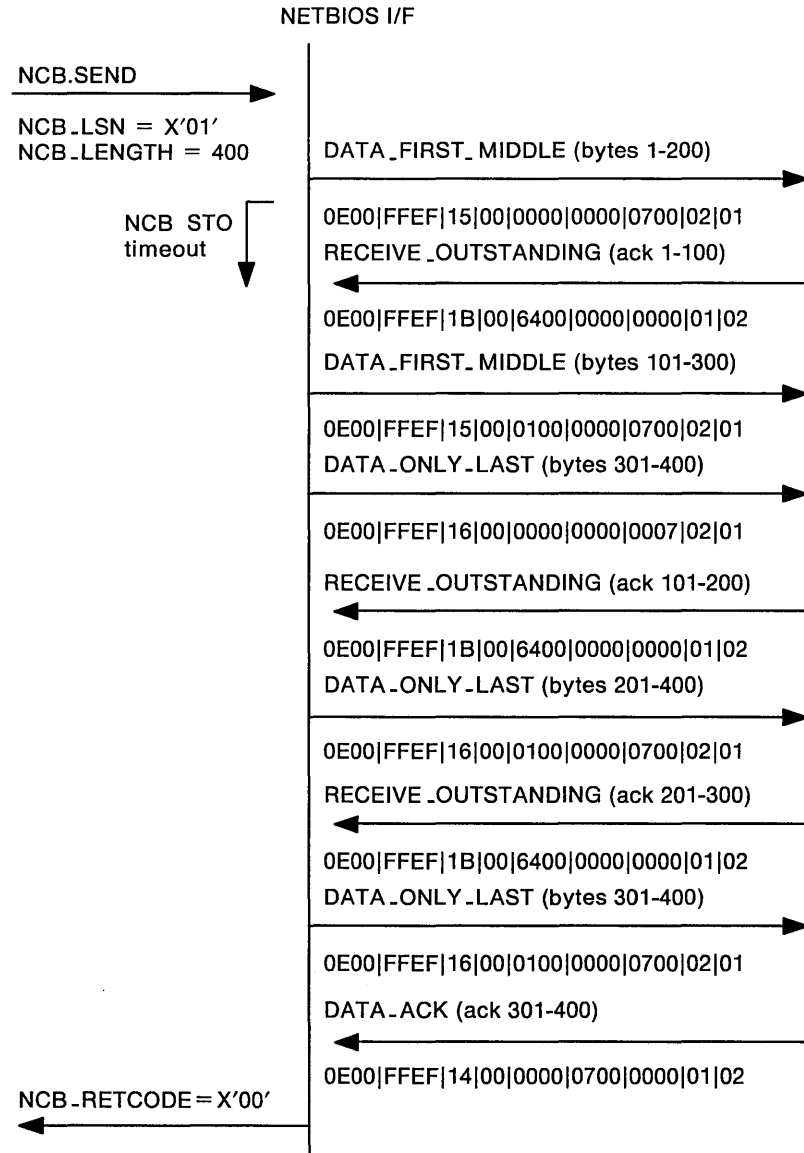
## Send Session Data: Segmentation and One Receive

The local node sends a 600-byte message over the session to the remote node that has a corresponding 600-byte RECEIVE outstanding. The local DHB size is such that the message must be segmented into 200-byte segments. The local node sends the first 200-byte segment with a bit requesting a RECEIVE\_CONTINUE from the remote node indicating it has a RECEIVE outstanding. The remote node responds with the RECEIVE\_CONTINUE which indicates acceptance of the 200-byte segment and indicates it can receive more data. The local node then sends the remainder.



## Send Session Data: Segmentation and Multiple Receives

The local node sends a 400-byte message over the session to the remote node. The remote node has 4 receives of 100 bytes each. The local DHB size is such that the message must be segmented into 200-byte segments. The local node sends the first 200-byte segment with a bit requesting a RECEIVE\_CONTINUE from the remote node indicating it has a RECEIVE outstanding. The remote node can only accept part of the first segment and responds with a RECEIVE\_OUTSTANDING, acknowledging 100 bytes and indicating more RECEIVES are outstanding. This continues until the entire message has been acknowledged.



## NETBIOS Protocol Examples with RND

Following are examples of NETBIOS 2.1 typical protocol scenarios that use Remote Name Directory function to communicate with a remote station.

The local station locates a remote name by issuing a NAME\_QUERY on-ring once and issues a NAME\_QUERY off-ring per transmit count and transmit timeout values that may be defined by NCB.TRANSMIT.COUNT and NCB.TRANSMIT.TIMEOUT parameters in the DIR.OPEN.ADAPTER command. The range for NCB.TRANSMIT.COUNT is 1-10 and the default is 6. The range for NCB.TRANSMIT.TIMEOUT is 1/2 second to 10 seconds and the default is 1/2 second.

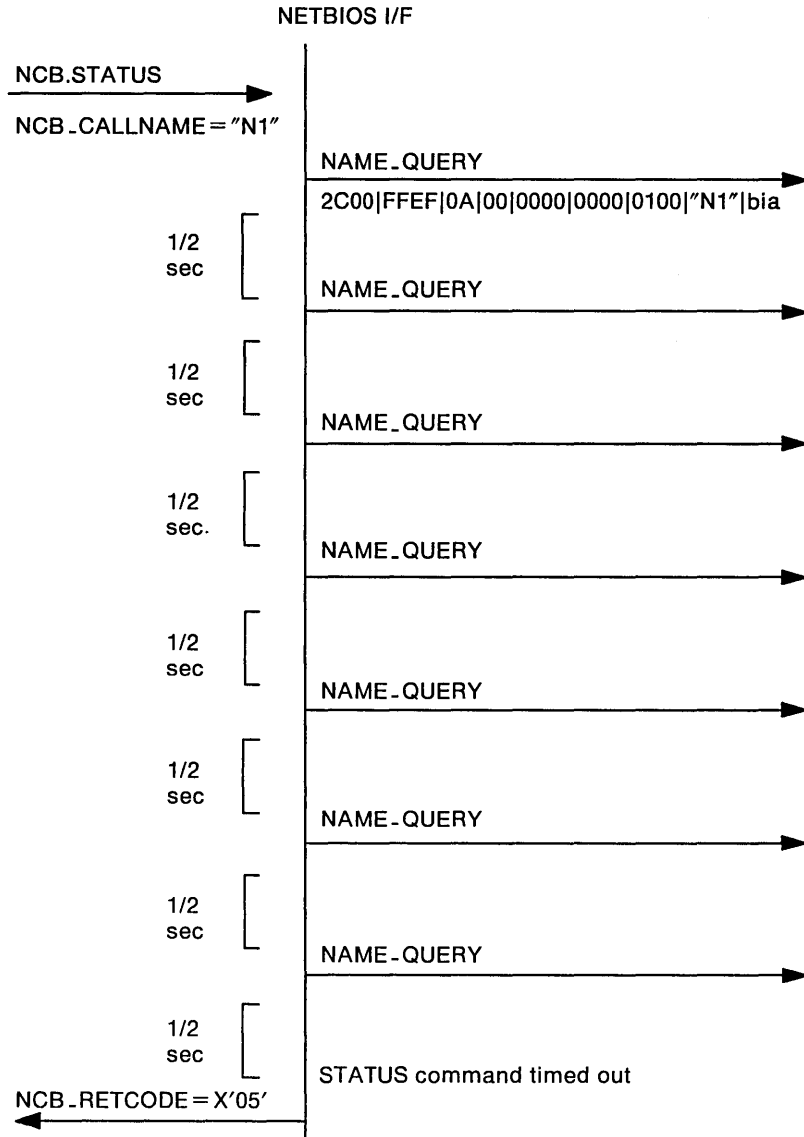
When RND is used, after the local station has located a remote name, the remote node address is saved and subsequent messages to that name will be to a specific node rather than a broadcast to all nodes.

The direction of a frame on the network is indicated by an arrow. The frame type is above the arrow and the contents of the frame are below the arrow. If a frame is repeated, the data appears only on the first instance of the frame.

**Note:** The numeric values shown in the various examples indicate the values as they appear on the line when transmitting the frame.

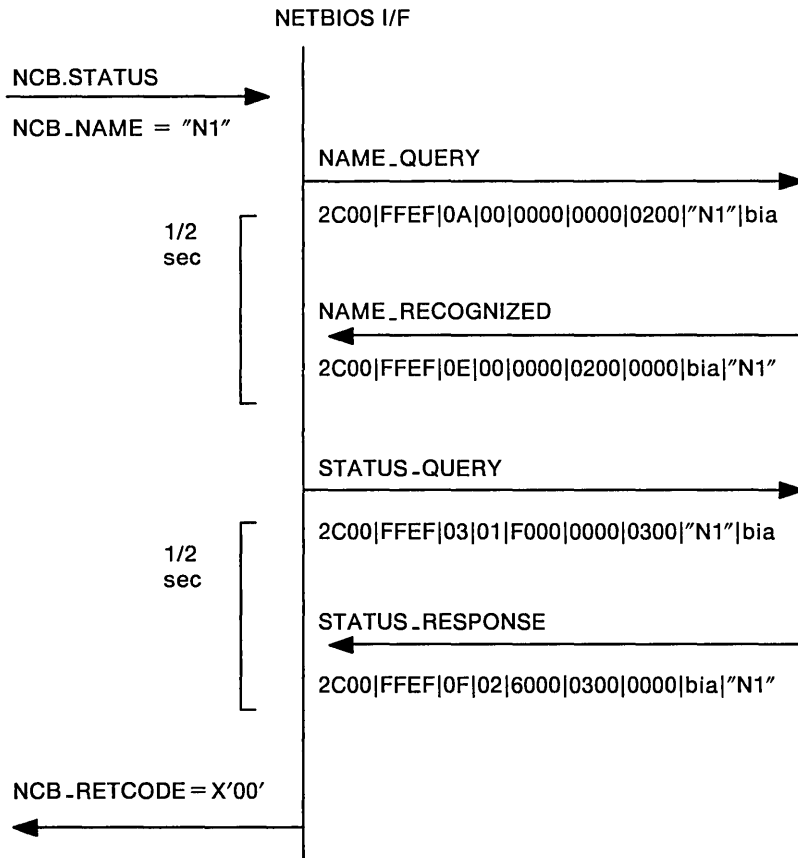
## Remote Adapter Status for a Name Not on Network or RND

The application program attempts to request the remote status for a name that is not on the network. NAME\_QUERY is sent to locate the name on the network on-ring once and off-ring at 1/2 second intervals for 6 times (by default). If a NAME\_RECOGNIZED is not received the application program NCB gets a return code of X'05' (command timed out).



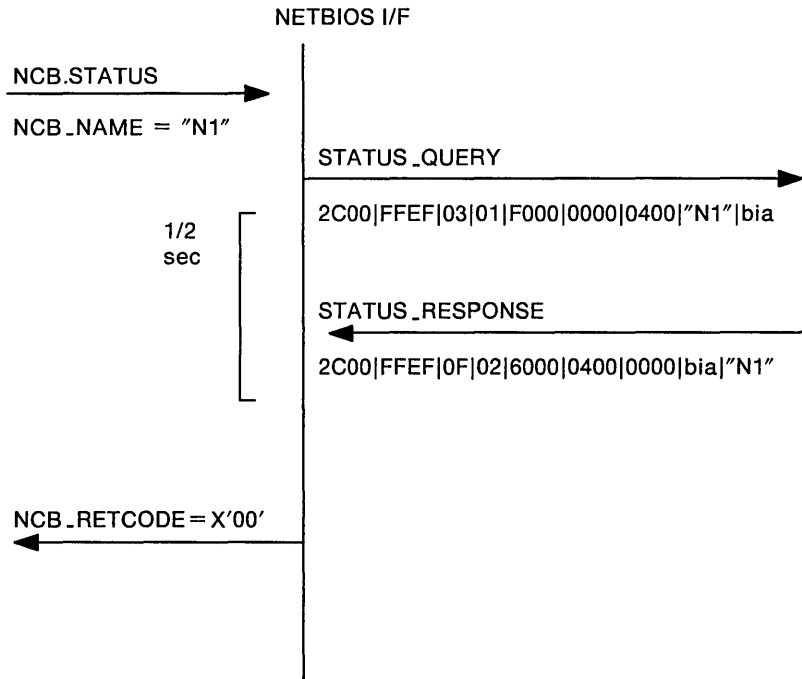
## Remote Adapter Status for a Name Not in RND but on the Network

The application program attempts to request the remote status for a name that is on the network. The local RND is checked to see if the remote name exists in the table. If the remote name does not exist in the RND, the NAME\_QUERY is sent to locate the name on the network on-ring once and off-ring at 1/2 second intervals for 6 times (by default). If a NAME\_RECOGNIZED is received, the local NETBIOS station adds the remote name and node address to the RND and sends a STATUS\_QUERY to that node. If a STATUS\_RESPONSE is received the application program NCB gets a return code of X'00'. It is assumed that the application program buffer is big enough to hold the status data.



## Remote Adapter Status for a Remote Name that Is in RND

The application program attempts to request the remote status for a name that is on the network. The local RND is checked to see if the remote name exists in the table. If the remote name exists in the RND, STATUS\_QUERY is sent to the specific node address that is saved in the RND. If a STATUS\_RESPONSE is received the application program NCB gets a return code of X'00'. It is assumed that the application program buffer is big enough to hold the status data.

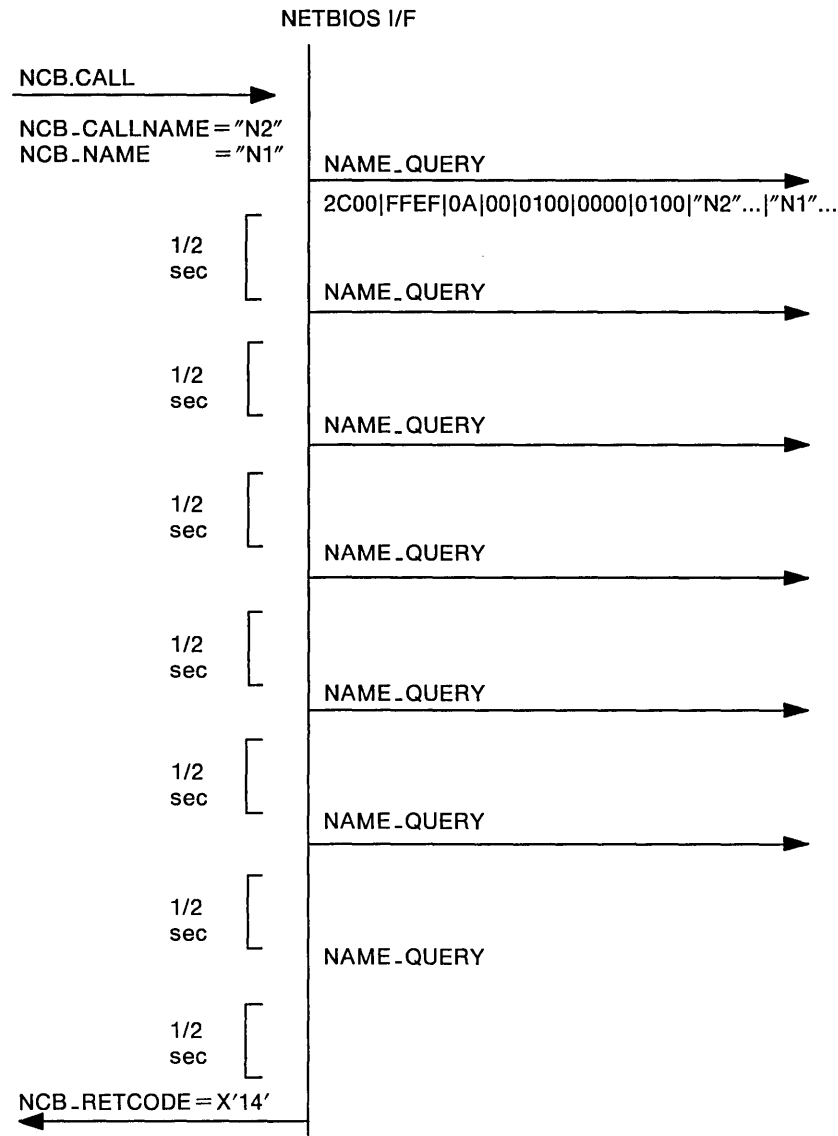


## Session Establishment RND Examples

Following are scenarios that show session establishment.

### Call a Name: Name Not on Network or RND

The application program attempts to locate a name on the network that does not exist. The NAME\_QUERY is sent on-ring once and at 1/2 second intervals for 6 times (by default). The NAME\_QUERY frame is sent to NETBIOS functional address (single-route broadcast). If a NAME\_RECOGNIZED is not received the application program NCB gets a return code of X'14' (cannot find name or no answer).

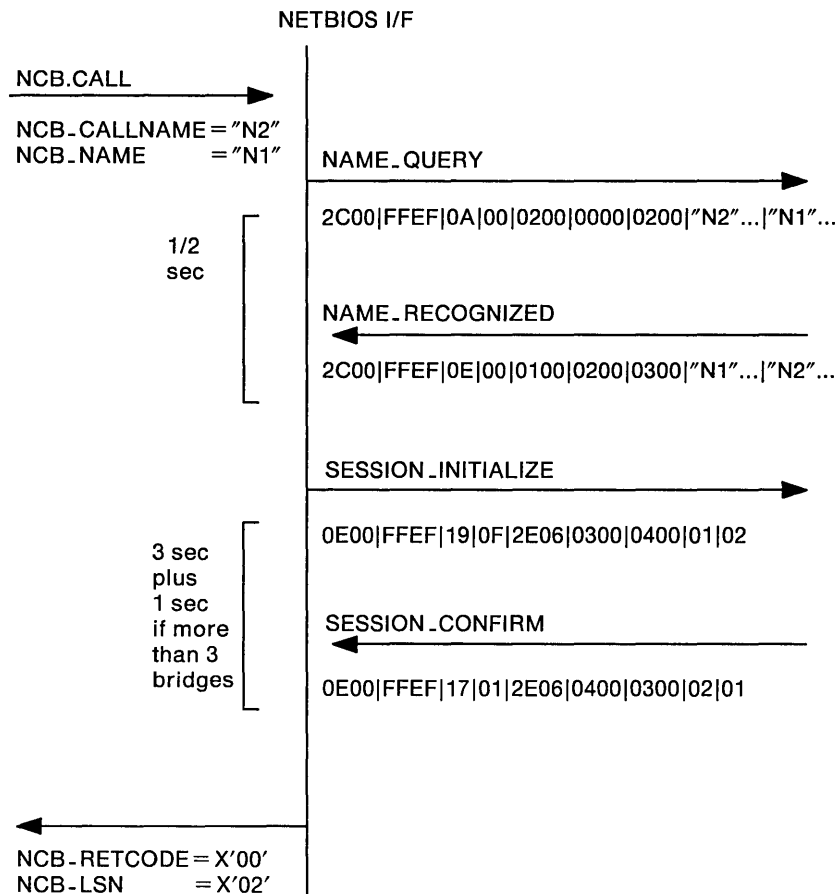




## Call a Name: Name Not in RND but Is on Network - Start Session

The application program uses name "N1" and issues a call to name "N2" which is registered at a remote node which has a LISTEN outstanding for that name. The NAME\_QUERY and NAME\_RECOGNIZED contain the respective session numbers. Receipt of the NAME\_RECOGNIZED causes the originating node to add the remote name "N2" and the remote node address to the RND, and to send a SESSION\_INITIALIZE (as an I-frame) and run a timer waiting for a SESSION\_CONFIRM (sent as an I-frame). The SESSION\_INITIALIZE and SESSION\_CONFIRM each indicate the maximum amount of user data that they are prepared to receive on this session. The remote partner will then limit the size of the user data in frames transmitted over this session to this size or the size available in its transmit buffer (DHB), whichever is smaller. Receipt of the SESSION\_CONFIRM will complete the call and set the NCB\_RETCODE to X'00' and NCB\_LSN to the locally assigned session number.

**Note:** If the remote name is already in RND, the NAME\_QUERY is sent to a specific node address rather than to a NETBIOS functional address.



## NETBIOS Protocol SEND.NO.ACK Examples

Following are NETBIOS 2.2 examples of typical protocol scenarios that use the SEND.NO.ACK/CHAIN.SEND.NO.ACK function to communicate with a remote station.

The direction of a frame on the network is indicated by an arrow. The frame type is above the arrow and the contents of the frame are below the arrow. If a frame is repeated, the data appears only on the first instance of the frame.

**Note:** The numeric values shown in the various examples indicate the values as they appear on the line when transmitting the frame.

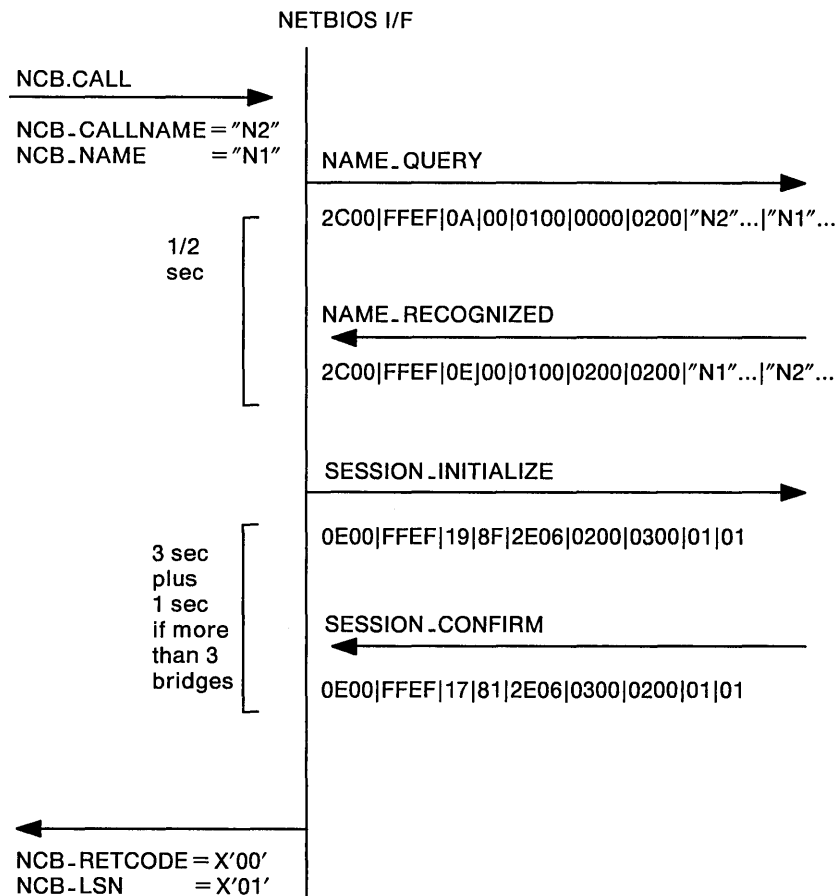
## Session Establishment Examples

Following are scenarios that show session establishment.

### Call a Name: Name Found - Start Session

The application program uses name "N1" and issues a call to name "N2" which is registered at a remote node which has a LISTEN outstanding for that name. The NAME\_QUERY and NAME\_RECOGNIZED contain the respective session numbers. Receipt of the NAME\_RECOGNIZED causes the originating node to send a SESSION\_INITIALIZE (as an I-frame) and run a timer waiting for a SESSION\_CONFIRM (sent as an I-frame). The SESSION\_INITIALIZE and SESSION\_CONFIRM each indicate the maximum amount of user data that they are prepared to receive on this session. The remote partner will then limit the size of the user data in frames transmitted over this session to this size or the size available in its transmit buffer (DHB), whichever is smaller.

A NO\_ACK indicator flag will be set in the DATA1 field of SESSION\_INITIALIZE and SESSION\_CONFIRM frames to indicate that the sending node has the ability to handle SEND.NO.ACK or CHAIN.SEND.NO.ACK commands. Receipt of the SESSION\_CONFIRM will complete the call and set the NCB\_RETCODE to X'00' and NCB\_LSN to the locally assigned session number.



## Session Data Transfer Examples

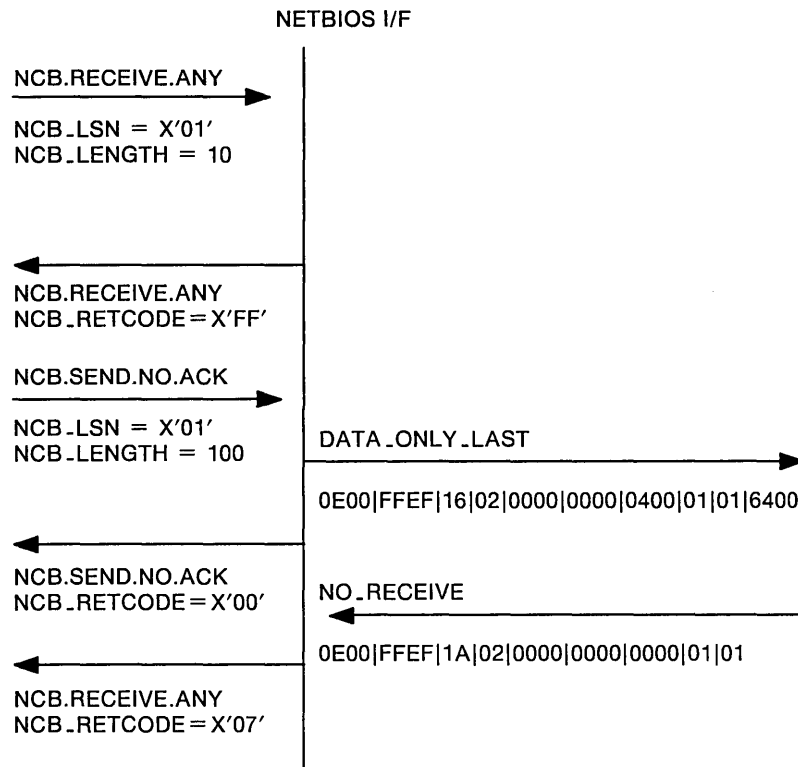
Following are scenarios that show data transfer across a session.

### Send Session Data: One SEND.NO.ACK and No Receive

The session has been established as above. The local side issues a RECEIVE\_ANY no-wait NCB for 10 bytes and a SEND.NO.ACK command for 100 bytes and the remote node has no RECEIVE outstanding. The local side sends a DATA\_ONLY\_LAST frame. Soon after the frame is transmitted, SEND.NO.ACK command completes with a X'00' return code.

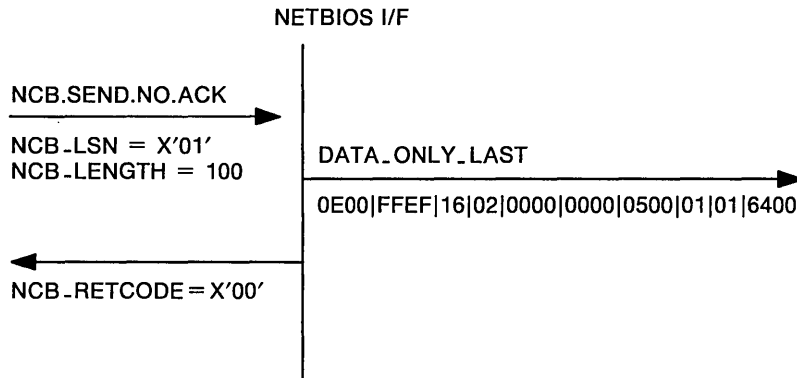
The remote side recognizes the DATA\_ONLY\_LAST frame and that the data is of SEND.NO.ACK or CHAIN.SEND.NO.ACK type. Since the remote side does not have any RECEIVE outstanding for the session, a NO\_RECEIVE frame is sent in response to indicate that the SEND.NO.ACK or CHAIN.SEND.NO.ACK data being either not received or partially received. When this frame is received, the local station will terminate a pending session type command (in this example, RECEIVE\_ANY) with a X'07' return code.

**Note:** If there is no pending receive command for the session at the receipt of a NO\_RECEIVE frame, the next RECEIVE command (or other applicable command) issued will get a X'07' return code.



### Send Session Data: One SEND.NO.ACK and One Receive

The session has been established as above. The local node issues a SEND.NO.ACK command for 100 bytes and the remote node has a corresponding 100-byte RECEIVE outstanding. The local node sends a DATA\_ONLY\_LAST and the SEND.NO.ACK completes with a X'00' return code.

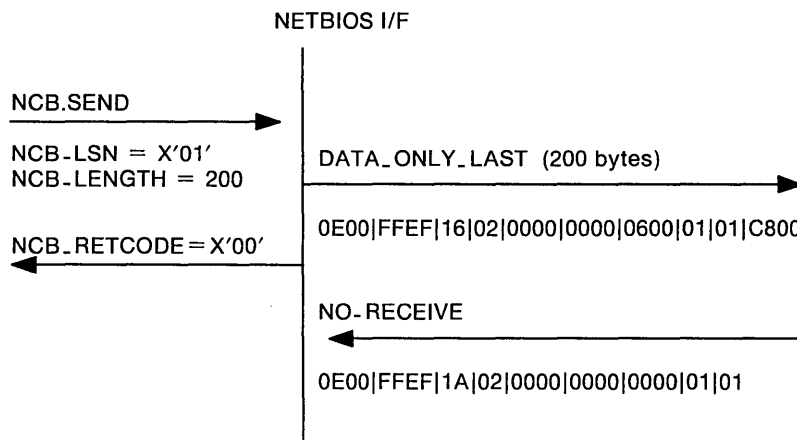


### Send Session Data: One SEND.NO.ACK and Multiple Receives

The local node sends a 200-byte message over the session to the remote node via the SEND.NO.ACK command. The remote node has two 100-byte RECEIVES outstanding. The remote node accepts 100-byte and transmits a NO\_RECEIVE frame indicating that only part of the SEND.NO.ACK or CHAIN.SEND.NO.ACK data received by the remote application program.

If there is no pending receive command for the session at the receipt of a NO\_RECEIVE frame, the next RECEIVE ( or other applicable command) command issued will get a X'07' return code.

**Note:** For data transmitted via SEND.NO.ACK or CHAIN.SEND.NO.ACK, only one receive command will be satisfied: the SEND.NO.ACK or CHAIN.SEND.NO.ACK data buffer must be no longer than the remote receive buffer for successful completion.



---

## **Chapter 6. The PC Network and PC Network (Baseband) Adapters**

See the following Technical References for information about PC Network and PC Network (Baseband) Adapters.

- *IBM PC Network Adapter Technical Reference*
- *IBM PC Network Adapter II Technical Reference*
- *IBM PC Network Adapter III/A Technical Reference*
- *IBM PC Network Adapter II - Frequency 2 Technical Reference*
- *IBM PC Network Adapter III/A - Frequency 2 Technical Reference*
- *IBM PC Network Adapter II - Frequency 3 Technical Reference*
- *IBM PC Network Adapter III/A - Frequency 3 Technical Reference*
- *IBM PC Network Baseband Adapter Technical Reference*
- *IBM PC Network Baseband Adapter/A Technical Reference*

For assistance in obtaining IBM manuals, see your place of purchase.



---

## Chapter 7. The Token-Ring Network Adapter Interface

Token-Ring Network Adapters	7-3
The Adapter Interface	7-3
An Overview of the Adapter	7-3
Data Transfer	7-4
Using Interrupts	7-5
Programmed I/O (PIO) Commands	7-5
Primary Adapter Addresses	7-5
Alternate Adapter Addresses	7-6
Adapter Setup Information	7-7
MMIO Domain	7-7
Interrupt Level Selection	7-8
Shared RAM Domain X'0A22' (X'0A26')	7-8
BIOS/MMIO Domain	7-9
BIOS/MMIO Layout	7-10
BIOS Area	7-10
The Attachment Control Area	7-10
Communication	7-11
The Registers	7-11
Shared RAM Relocation Register (RRR) Pair	7-11
Write Management Register Pairs	7-12
Read-Only Area	7-13
Interrupt Register Pairs	7-13
Adapter Check Data	7-15
The Adapter Identification Area	7-20
Contents of the AIP	7-20
Shared RAM	7-23
Shared RAM Layout	7-23
Shared RAM Paging	7-24
Uninitialized Shared RAM	7-25
Operating the Adapter	7-26
Commands	7-29
Adapter SRB Commands	7-30
Direct Interface Commands	7-30
DIR.CLOSE.ADAPTER	7-30
DIR.INTERRUPT	7-31
DIR.MODIFY.OPEN.PARMS	7-32
DIR.OPEN.ADAPTER	7-32
DIR.READ.LOG	7-37
DIR.RESTORE.OPEN.PARMS	7-38
DIR.SET.FUNCT.ADDRESS	7-39
DIR.SET.GROUP.ADDRESS	7-39
DLC SRB Commands	7-40
DLC.CLOSE.SAP	7-41
DLC.CLOSE.STATION	7-42
DLC.CONNECT.STATION	7-43
DLC.FLOW.CONTROL	7-44
DLC.MODIFY	7-45
DLC.OPEN.SAP	7-47
DLC.OPEN.STATION	7-50
DLC.REALLOCATE	7-52
DLC.RESET	7-53
DLC.STATISTICS	7-54



Transmit Commands .....	7-56
Transmit Summary .....	7-57
Adapter-to-PC system Commands .....	7-58
DLC.STATUS .....	7-59
RECEIVED.DATA .....	7-60
RING.STATUS.CHANGE .....	7-62
TRANSMIT.DATA.REQUEST .....	7-62
Additional Adapter Functions .....	7-64
Adapter's Shared RAM .....	7-64
The Bridge Functions .....	7-64
Shared RAM Layout for Bridge Use .....	7-65
The Bridge Commands .....	7-65
DIR.CONFIG.BRIDGE.RAM .....	7-66
DIR.SET.BRIDGE.PARMS .....	7-67
The Additional Adapter-to-PC System Command .....	7-68
RECEIVED.BRIDGE.DATA .....	7-69
Transmitting Bridge Forwarded Frames .....	7-70
Initiating Transmission .....	7-71
The Bridge Transmit Buffer Layout .....	7-72
POS Registers in Token-Ring Network Adapters with Micro Channel .....	7-75
Early Token Release .....	7-77
Additional Factors to Consider .....	7-78
ETR Conclusion .....	7-78

---

## Token-Ring Network Adapters

There are two types of Token-Ring Network adapters:

- Token-Ring Network adapters with PC I/O Bus and
- Token-Ring Network adapters with Micro Channel.

When both types are discussed in this chapter, they are referred to as “adapters.”

---

## The Adapter Interface

The hardware interface provided by Token-Ring Network adapters is used to allow the PC system software to communicate to the adapter hardware and software. PC system software (application programs) use the hardware interface to issue commands, store and retrieve data, receive commands, use timing facilities, and identify the adapter.

The Token-Ring Network adapters contain several areas that allow communication between PC system software and the adapter. These areas include the following:

- Programmed I/O (PIO)
- Memory Mapped I/O (MMIO)
- shared RAM
- BIOS.

The PIO areas are accessed using IN and OUT instructions. The MMIO and shared RAM areas are accessed using memory instructions. The BIOS area, which is read only, may contain an optional program that is executed by the PC system during power-on.

When IBM adapter support software or equivalent is not used, the application program must perform the control functions and interrupt handling for the adapter. The application program must load the commands and parameters into the shared RAM, and also control interrupt bits in the adapter MMIO domain where the interrupt status registers are assigned. The application program must then interrogate control blocks and registers when the adapter has updated shared RAM.

Descriptions of the sequence of operations, the shared RAM assignments, the process of initializing the adapter, and related responses are included in this chapter.

## An Overview of the Adapter

- The PIO area is used for adapter configuration information and control. The PIO area uses I/O addresses that range from X'0A20' to X'0A27' for both the Token-Ring Network adapters with PC I/O Bus and the Token-Ring Network adapters with Micro Channel, and X'02F0' to X'02F7' for the Token-Ring Network adapters with PC I/O Bus only. These addresses are also used to obtain the MMIO address and interrupt level used by the adapter and provide the ability to reset and release the adapter. For the Token-Ring Network adapters with Micro Channel, they are used to obtain the shared RAM address for the adapter.
- The MMIO area is a movable section of memory mapped I/O that is 512 bytes and consists of the following:
  - The Attachment Control Area contains registers for adapter operations control.

**RRR** The shared RAM relocation register is used to get information for the shared RAM on the adapter. For a PC system with PC I/O Bus adapters it is also used to set the shared RAM address in PC system memory.

### WRBR/WWOR/WWCR

The write region base register, write window open register, and the write window close register are used to control the read/write access to the shared RAM on the adapter. The R/O access is used to protect the Token-Ring Network parameters. The R/W areas are used to pass data to the adapter.

### ISRA/ISRP

The interrupt status register-adapter and the interrupt status register-PC system are used for the main communication between the adapter and the PC system.

- The adapter's PC System Programmable Timer is a general purpose timer for use by the PC system software. The offset addresses of the timer registers are X'1E0C' and X'1E0D' within the BIOS/MMIO segment.
- The adapter ID PROM (AIP) is used to provide the adapter type and the adapter's encoded address.
- The shared RAM is a movable section of memory and is either 8K, 16K, 32K, or 64K bytes, and is used for data transfer and control blocks. Shared RAM is where data and all of the software commands are stored.
- Each adapter has an optional BIOS area that is a movable section of mapped memory that is 7.5K bytes and can be used for bring-up or initialization code that is executed at power-on time.

## Data Transfer

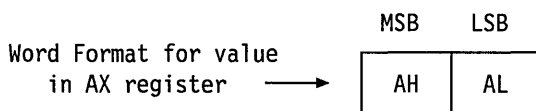
Data may be obtained from three general sections of the adapter.

- The PIO area, which is 8 bits wide. Byte read and write I/O instructions may be used.
- The MMIO area, which is 8 bits wide. Both byte and word read and write memory instructions may be used except when accessing the AIP. The AIP is 8 bits wide, but only 4 bits are valid and should be accessed using a byte read memory instruction to an even address.
- The shared RAM area which is 8 bits wide for Token-Ring Network adapters with PC I/O Bus and 16 bits wide for Token-Ring Network adapters with Micro Channel. Both byte and word read and write memory instructions may be used.

### Notes:

1. All word variables placed by the adapter in the adapter's shared RAM are in IBM format. All word variables stored in the shared RAM of the adapter by the PC system must also be in the IBM format. The Intel method is the reverse of the IBM method. See the example below.

Example:



MSB = Most Significant Byte

LSB = Least Significant Byte

If the value is stored at location 100 using Intel format the registers would be placed into memory as follows:

**AH (MSB)** stored at address 101

**AL (LSB)** stored at address 100

If the value is stored at location 100 using IBM format the registers would be placed into memory as follows:

**AH (MSB)** stored at address 100

**AL (LSB)** stored at address 101

As seen from the above example, the MSB and LSB using the Intel method is the reverse of the IBM method. The PC system software must store word values into shared RAM using the IBM method.

2. In the MMIO area there are several 8-bit registers. These registers are in pairs. For example, there are the RRR-even and RRR-odd pair. These will be referred to as the RRR-even (the register at the even address) and the RRR-odd (the register at the odd address).

## Using Interrupts

Interrupts are used by the adapter to signal the PC system software of events that have occurred on the adapter. The PC system software uses adapter interrupts to communicate with the adapter microprocessor.

---

## Programmed I/O (PIO) Commands

There are four I/O addresses dedicated for PIO operations to each possible adapter (primary or alternate). Read (IN) or write (OUT) operations to these addresses either cause an action or transfer data. The addresses and their uses are described below.

### Primary Adapter Addresses

Address	Operation	Result	Adapter Type
X'02Fn'	Write	Global interrupt enable	Token-Ring Network adapters with PC I/O Bus
X'0A20'	Read	Setup Read 1	Both
X'0A20'	Write	Reserved	Both
X'0A21'	Read	Reserved	Both
X'0A21'	Write	Adapter reset latch	Both
X'0A22'	Read	Setup Read 2	Token-Ring Network adapters with Micro Channel
X'0A22'	Write	Adapter reset release	Both
X'0A23'	Read	Reserved	Both
X'0A23'	Write	Interrupt enable	Token-Ring Network adapters with PC I/O Bus

## Alternate Adapter Addresses

Address	Operation	Result	Adapter Type
X'02Fn'	Write	Global interrupt enable	Token-Ring Network adapters with PC I/O Bus
X'0A24'	Read	Setup Read 1	Both
X'0A24'	Write	Reserved	Both
X'0A25'	Read	Reserved	Both
X'0A25'	Write	Adapter reset latch	Both
X'0A26'	Read	Setup Read 2	Token-Ring Network adapters with Micro Channel
X'0A26'	Write	Adapter reset release	Both
X'0A27'	Read	Reserved	Both
X'0A27'	Write	Interrupt enable	Token-Ring Network adapters with PC I/O Bus

The commands are:

### GLOBAL INTERRUPT ENABLE

For PC system with PC I/O Bus: An I/O Write (OUT instruction) to X'02Fn' is used as a global interrupt enable for the Token-Ring Network adapters with PC I/O Bus. This resets interrupt generating circuits in all adapters sharing the PC system interrupt facilities. The specific interrupt level is defined by the value of "n". This command performs no function for the Token-Ring Network adapters with Micro Channel.

### SETUP READ 1

For PC system with Micro Channel: An I/O Read (IN instruction) from X'0A20' (Primary) or X'0A24' (Alternate) returns all but the high-order bit of the 1-byte value for the BIOS/MMIO domain address<sup>1</sup> and two bits of interrupt level information. This information must have been set during the setup function of POST. The address specifies where, in a 512K portion of 1M of PC system with Micro Channel-addressable memory, the adapter registers will be located. An I/O Write to this address is reserved. See the chart on page 7-7 for bit assignments.

For PC system with PC I/O Bus: An I/O Read (IN instruction) from X'0A20' (Primary) or X'0A24' (Alternate) returns all but the high-order bit of the 1-byte value for the BIOS/MMIO domain address and two bits of interrupt level information. The switches must be set when the adapter is installed to define where in the PC system-addressable memory the adapter registers will be located. An I/O Write to this address is reserved. See the adapter's *Guide to Operations* for more information on setting adapter switches.

### ADAPTER RESET LATCH

An I/O Write (OUT instruction) to X'0A21' (Primary) or X'0A25' (Alternate) causes an unconditional adapter reset to be latched on. The entire adapter is held reset until an Adapter Reset Release is received from the PC system. The adapter reset state is similar to a power-on reset and is used to start the adapter in a known state. While the adapter is held reset, the PC system cannot access either the shared RAM or the MMIO region (except for the BIOS area). An I/O Read to this address is reserved.

### SETUP READ 2

For PC system with Micro Channel: An I/O Read (IN instruction) from X'0A22' (Primary) or X'0A26' (Alternate) returns the 1-byte value containing the shared RAM address and the high-order bit of the BIOS/MMIO domain address is also included. This information must have been set during the setup function of POST. The address specifies where, in a 1M space of PC

<sup>1</sup> In this document the term *domain address* refers to the address from the perspective of the PC system.

system with Micro Channel-addressable memory, the shared RAM on the adapter will be located. The BIOS/MMIO address bit specifies which 512K portion of 1M PC system with Micro Channel-addressable memory the BIOS/MMIO domain is in. See the chart on page 7-8 for bit assignments.

### ADAPTER RESET RELEASE

An I/O Write (OUT instruction) to X'0A22' (Primary) or X'0A26' (Alternate) turns off an adapter reset condition that has been previously latched on by an Adapter Reset Latch from the PC system. Before the adapter can be completely reset, at least 50 milliseconds must elapse between the Adapter Reset Latch and the Adapter Reset Release I/O instructions. If the adapter is not latched in a reset condition, the command is ignored.

### ADAPTER INTERRUPT ENABLE

For PC system with PC I/O Bus: An I/O Write (OUT instruction) to X'0A23' (adapter 0) or X'0A27' (adapter 1) resets and re-enables the adapter interrupt generation circuitry. Since this leaves all other PC system adapters disabled, this adapter is able to monopolize the interrupt facilities. An I/O read to this address is reserved.

## Adapter Setup Information

### MMIO Domain

X'0A20' (X'0A24')

The BIOS/MMIO domain is mapped to any contiguous 8K block within a 1M PC system address space.

If an optional BIOS module is installed on the adapter that executes at power-on time, the BIOS/MMIO domain must be limited to the 96K of BIOS space in the PC system (X'0C8000-0DFFFF').

The value read from X'0A20' (Primary) or from X'0A24' (Alternate) is defined in the following chart.

Boundary *	Bit in the PC system address	Bit in the Byte read	Switch Number if PC system with PC I/O Bus
256K	18	7	1
128K	17	6	2
64K	16	5	3
32K	15	4	4
16K	14	3	5
8K	13	2	6
Interrupt Level	Encoded Bits	1	7
		0	8
* For PC system with Micro Channel: See bit zero of X'0A22' (X'0A26') for the value of address bit 19 (512K). For PC system with PC I/O Bus: Bit 19 is always equals to one.			

## Interrupt Level Selection

The interrupt level select bit values are:

Bit 1	Bit 0	Interrupt Level	
		PC system with PC I/O Bus	PC system with Micro Channel
0	0	IRQ 2	IRQ 2
0	1	IRQ 3	IRQ 3
1	0	IRQ 6	IRQ 10
1	1	IRQ 7	IRQ 11

The interrupt levels in order of precedence are as follows:

Priority	PC system with PC I/O Bus	PC system with Micro Channel
Highest priority	2	2
Second priority	3	10
Third priority	6	11
Lowest priority	7	3

## Shared RAM Domain X'0A22' (X'0A26')

For PC system with PC I/O Bus: The shared RAM domain is set by software using the RRR-even register.

For PC system with Micro Channel: The value read from X'0A22' (Primary) or X'0A26' (Alternate) is defined in the following chart.

Boundary	Bit in the PC system address	Bit in the Byte read
512K	19	7
256K	18	6
128K	17	5
64K	16	4
32K	15	3
16K	14	2
8K	13	1 (Always 0)
512K	Bit 19 of the BIOS/MMIO domain address	0

## BIOS/MMIO Domain

The BIOS/MMIO domain is a contiguous 8K region containing BIOS space, registers, and permanent ID information.

The address driven to the adapter by the PC system contains several fields when used in the MMIO area.

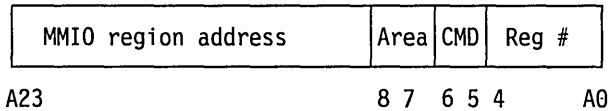


Figure 7-1. PC Address Bits

Bits 23 through 9 select the MMIO region.

Bits 8 and 7 select the 128-byte area within the region.

- B'00' is the attachment control area
- B'01' (Do not access. This is a reserved area.)
- B'10' is the adapter identification area A
- B'11' is the adapter identification area B

Bits 6 and 5 select the specific MMIO command (CMD) to be performed.

Bits 4 through 0 select the specific register (Reg #) and byte.

With the BIOS/MMIO domain address from the setup information obtained by using the PIO instructions, MMIO read or write instructions can be used to put data into or read data from the adapter registers. Four MMIO commands are used. These four commands are controlled by the specific address used.

### READ

Read the contents of an adapter control register into the PC system register. A READ is performed by issuing a read instruction in the PC system with an address pointing to the appropriate MMIO register of the adapter.

### WRITE

Transfer the contents of a PC system register directly into the selected adapter register. A WRITE is performed by issuing a write instruction in the PC system with an address pointing to the appropriate MMIO register of the adapter with the 2 bits in the address assigned as CMD being set to B'00'.

### OR

OR the contents of a PC system register into the selected adapter register. An OR is performed by issuing a write instruction in the PC system with an address pointing to the appropriate MMIO register of the adapter with the 2 bits in the address assigned as CMD being set to B'10'.

### AND

AND the contents of a PC system register into the selected adapter register. An AND is performed by issuing a write instruction in the PC system with an address pointing to the appropriate MMIO register of the adapter with the 2 bits in the address assigned as CMD being set to B'01'.

See "BIOS/MMIO Layout" and Figures 7-2, 7-3, and 7-4 beginning on page 7-18.



## BIOS/MMIO Layout

Offset From BIOS/MMIO Segment	Contents	
0000	X'55' *	BIOS Area
0001	X'AA' *	
0002	Length /512 *	
0003	BIOS code - first instruction *	
	Reserved for BIOS code, 7675 bytes *	
1DFF	BIOS code - last location *	
1E00	Attachment Control Area 128 bytes	MMIO Area
1E80	Reserved 128 bytes (do not access)	
1F00	AIP 256 bytes (read only access)	
* If BIOS is present. Otherwise, undefined.		

### BIOS Area

All but the last 512 bytes of the BIOS/MMIO domain are reserved for PC system **BIOS** program storage.

### The Attachment Control Area

This area in the MMIO region contains the registers for controlling operations of the adapter. It is located at X'1E00' beyond the MMIO domain origin and is 128 bytes long. This area contains 9 pairs (even and odd) of 1-byte registers.

These registers are listed below.

<b>RRR</b>	shared RAM relocation register
<b>WRBR</b>	write region base register
<b>WWCR</b>	write window close register
<b>WWOR</b>	write window open register
<b>ISRP</b>	interrupt status register-PC system
<b>ISRA</b>	interrupt status register-adapter
<b>TCR</b>	timer control register
<b>TVR</b>	timer value register
<b>SRPR</b>	shared RAM paging register. This register is available only on the 16/4 adapters.

These registers are accessed by both the PC system and the adapter. The PC system cannot *lock* a register to prevent adapter access.

## Communication

The two registers in the attachment control area that are used by the adapter support software and the adapter respectively to cause interrupts to one another are the ISRA and the ISRP.

Other registers in the attachment control area and control blocks in shared RAM are also used for communication between the PC system and the adapter. The control blocks are explained in "Shared RAM" on page 7-23.

The majority of communication actions use the following areas:

From PC System to Adapter	From Adapter to PC System	Usage	Located in
ISRA	ISRP	Interrupts	MMIO
SRB	ARB	Requests	Shared RAM
SSB	ASB	Status	Shared RAM

## The Registers

All of the register pairs in the Attachment Control Area consist of two 1-byte registers, the register at the even address and the register at the odd address.

### Shared RAM Relocation Register (RRR) Pair

**RRR-even** For Token-Ring Network adapters with PC I/O Bus, bits 7 through 1 of the even register are used to set the shared RAM starting address. This location must be set before the shared RAM can be accessed, and must be set to a location in the memory map that does not cause a conflict. These register bits default to zero on power-up or after an adapter reset. If the register contains zero, the shared RAM is not mapped into the memory map. This register must be set to a correct address boundary as indicated below:

8K shared RAM should be on an 8K address boundary.

16K shared RAM should be on a 16K address boundary.

32K shared RAM should be on a 32K address boundary.

64K shared RAM should be on a 64K address boundary.

**Note:** For shared RAM paging, the address boundary can be on a 16K boundary since only 16K of PC address space is used.

For Token-Ring Network adapters with Micro Channel, bits 7 through 1 of the even register are reserved and should not be changed.

**RRR-odd** Bits 3 and 2 indicate the shared RAM size; the amount of shared RAM in the PC system mapped memory. These bits are coded as shown below.

Both bits zero (00) for 8K.

Bit 3 is zero and bit 2 is one (01) for 16K.

Bit 3 is one and bit 2 is zero (10) for 32K.

Bit 3 is one and bit 3 is one (11) for 64K.

This shared RAM may not be the total amount of shared RAM on the adapter, but this value indicates the amount of shared RAM for the PC system to map into its memory. (For example, the 16/4 Adapter has 64K of available shared RAM, but may be set for 16K to allow shared RAM paging. If the RRR-odd bit 2 is set to one and bit 3 set to zero, this would indicate 16k of shared RAM in the PC system's memory map.)

**Warning:** The remaining RRR-odd bits, though readable, are controlled by the adapter. These bits should not be changed.

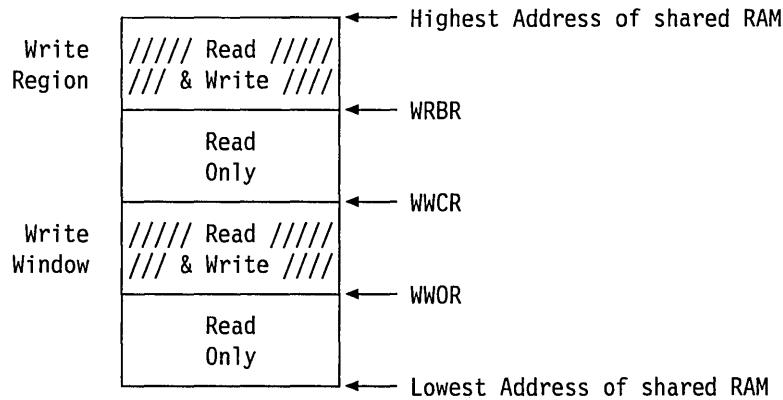
## Write Management Register Pairs

The WWCR pair, the WWOR pair, and the WRBR pair are used by the adapter to dynamically control the PC system write access to shared RAM. Only the adapter can write into these three register pairs.

These management register pairs specify an offset into shared RAM. The offsets are 16-bit values. The even register contains the most significant byte of this value. For example:

WRBR-even = 47  
 WRBR-odd = 24  
 WRBR value = 4724

All references to a register without the odd/even notation indicate this 16-bit offset value.



The adapter can concurrently define two separate and independent computer write areas within the shared RAM domain: the *write region* and the *write window*. The size of each of these areas can be individually defined in word (2-byte) increments from 2 bytes to the maximum size of the shared RAM domain.

The two areas differ only in how they are bound. The write region always extends from the highest address of the shared RAM domain down to a variable origin specified by the WRBR. The write window extends from a variable base defined by the WWOR pair to a variable limit defined by the WWCR pair. Also, the low-order bit in each odd register is zero since all write boundaries are word (2-byte) aligned.

The interface mechanism allows the PC system read-only access to the entire shared RAM domain until the adapter is initialized and PC system write-access areas are defined by the adapter.

The WRBR contains either zero or the offset of the beginning of the write region. When this field is zero, no write region is available.

The WWOR contains either zero or the offset of the beginning of the write window. This field will contain zero until the adapter is opened, and when it is zero, no write window is available.

The WWCR contains either zero or the first offset after the last writeable offset. This field is reserved until the adapter is opened, and when it is zero, no write window is available.

## Read-Only Area

Any address in the shared RAM not given specific PC system write access by the shared RAM management registers will be PC system read-only access. A PC system write to any of these read-only memory addresses or to any shared RAM management register MMIO address will not be completed and will activate the PC Access error interrupt condition (ISRP-even bit 2). Since the origin of the write region (WRBR) and the write window (WWOR) must be greater than zero if either write area is to be defined, the first 2 bytes of the shared RAM domain are always read-only to the PC system.

## Interrupt Register Pairs

The ISRA pair and the ISRP pair are used by the application program running in the PC system and by the adapter, respectively, to cause interrupts to one another.

There are also bits used by the adapter in these registers that can be interrogated by the program in the PC system to determine adapter conditions.

**Interrupt Status Register-Adapter Pair:** The adapter support software sets bits in the ISRA-odd to interrupt the adapter. These are bits 6 through 0. The bits in ISRA-even are controlled by the adapter and may be interrogated by the program for information about the adapter.

The ISRA-even

BIT	NAME	DESCRIPTION
7	Internal Parity Error	If this bit was on, there was parity error on the adapter's internal bus.
6	Timer interrupt	At least one of the TCR-odd timers has an interrupt to present to the adapter.
5	Access interrupt	When this bit is on, it indicates a shared RAM access violation or an illegal MMIO operation by the adapter to an Attachment Control Area register has occurred.
4	Deadman timer has expired	The deadman timer has expired, indicating an adapter microcode problem. This bit is one of the conditions that can set ISRP-even bit 3.
3	Adapter processor check	This bit does not latch on but follows the state of the adapter processor machine check indication. This bit is one of the conditions that can set ISRP-even bit 3.
2	Reserved	
1	Adapter hardware interrupt mask	When this bit is on, it prevents adapter hardware interrupts (ISRA-even bit 7) from being presented to the adapter processor.
0	Adapter software interrupt mask	When this bit is on, it prevents adapter software interrupts (ISRA-odd bits 0-6) from being presented to the adapter processor.

## The ISRA-odd

BIT	NAME	DESCRIPTION
7	Reserved	
6	Bridge frame forward request	The application program has placed a frame in the bridge transmit buffers and is requesting that the frame be forwarded (Bridge function is not available on the Token-Ring Network PC Adapter).
5	Command in SRB	The application program has placed a command in the SRB and is informing the adapter.
4	Response in ASB	The application program has placed the response to an ARB request in the ASB and is informing the adapter.
3	SRB free request	The application program wants to use the SRB, but a previous request is still being processed by the adapter. The adapter will return an "SRB free" interrupt when the SRB return code field has been set.
2	ASB free request	The application program wants to use the ASB, but a previous response is still being processed by the adapter. The adapter will return an "ASB free" interrupt when the ASB return code field has been set.
1	ARB free	The command in the ARB has been read by the application program and the ARB is available. If the command requires a response from the application program (receive and transmit only), it will be provided in the ASB later.
0	SSB free	The response in the SSB has been read by the application program and the SSB is available.

**Interrupt Status Register-PC System Pair:** The adapter sets bits in the ISRP pair to interrupt the PC system. Provided in these registers are methods of interrupting and blocking interrupts to the PC system.

## The ISRP-even

BIT	NAME	DESCRIPTION
7	CHCK/IRQ steering control	This bit is used to control error interrupts. If 0, the adapter will issue a CHCK. If 1, the adapter will issue IRQ. CHCK is not supported on the Adapter/A and this bit must be set to 1.
6	Interrupt enable	When this bit is on, interrupt requests will be presented to the PC system. When this bit is off, all interrupts are masked off. The bit can be set by either the adapter or the PC system.
5	Reserved	
4	Timer interrupt	When this bit is on, the TVR-even has expired.
3	Error interrupt	The adapter has had either a machine check occur, the adapter deadman timer expire, or the adapter timer overrun.
2	Access interrupt	When this bit is on, it indicates a shared RAM access violation or an illegal MMIO operation by the computer to an Attachment Control Area register pair has occurred. The following conditions will set this bit: <ul style="list-style-type: none"> <li>Any PC system write to a write-protected location in the shared RAM domain</li> <li>Any PC system write to a shared RAM management (WRBR, WWCR, WWOR) register</li> <li>Any PC system write to ISRA-even</li> <li>Any PC system write to a nonzero interrupt field of ISRP-odd or ISRA-odd. Nonzero interrupt fields of ISRP-odd and ISRA-odd must be manipulated using OR and AND MMIO commands.</li> </ul>
1		For PC system with PC I/O Bus: Interrupt block bit
0	Primary/Alternate address	This bit reflects the setting of the adapter primary/alternate setup information. If this bit is off, the primary adapter address is selected. If this bit is on, the alternate adapter address is selected.

## The ISRP-odd

BIT	NAME	DESCRIPTION
7	Reserved	
6	Adapter check	The adapter has encountered an unrecoverable error and is closed. The reason for the check may be read from the shared RAM using the address in the write window close management register pair in the attachment control area of the MMIO region. The information returned is defined in "Adapter Check Data."
5	SRB response	The adapter has recognized an SRB request and has set the return code in the SRB. A return code of: <ul style="list-style-type: none"><li>• X'00' : Indicates successful completion of the SRB request.</li><li>• X'01'-X'FD' : Indicates unsuccessful completion of the SRB request.</li><li>• X'FF' : Indicates that the request has been accepted and is in process. A subsequent SSB response will be issued at the command completion. This interrupt bit is set for this return code only if the PC system has set the "SRB Free Request" bit in the ISRA-odd.</li></ul>
4	ASB free	The adapter has read the response provided in the ASB, and the ASB is available for another response. This interrupt bit is set only if the PC system has set the "ASB Free Request" bit in the ISRA-odd or if an error has been detected in the response.
3	ARB command	The ARB contains a command for the PC system to act on.
2	SSB response	The SSB contains a response to a previous SRB command from the PC system.
1	Bridge frame forward complete	The adapter has completed transmitting a frame forwarded by the bridge application program (All Token-Ring Network adapters except the original Token-Ring Network PC Adapter)
0	Reserved	

## Adapter Check Data

When the ISRP-odd bit 6 is set, indicating an adapter check, the address in the write window close management register pair will point to data defining the reason for the check.

The data is in the following format:

**Bytes 0-1** Adapter check reason code

**Bytes 2-3** Parameter 1

**Bytes 4-5** Parameter 2

**Bytes 6-7** Parameter 3

See "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43 in Appendix B, "Return Codes." The reason codes and parameter values can be useful when the adapter is returned for service.

**Timer Control Register (TCR) Pair:** This register pair controls timing for both PC system and the ring. The even register is used along with the TVR pair to control the PC system programmable timer. The odd register controls the fixed-duration timers provided as a hardware base for the adapter microcode timing routines.

## The TCR-even

BIT	NAME	DESCRIPTION
7	PC system programmable timer interrupt mask	This bit controls the timer interrupt operation. When this bit is on, the timer interrupts the PC system when the programmable count expires. When the bit is off, the timer will not interrupt the computer, and the timer status must be obtained by polling either the ISRP-even or the TVR-even. The timer interrupt, like all PC system interrupts, is also subject to the interrupt enable bit (ISRP-even bit 6).
6	PC system programmable timer reload mode	If this bit is on, the timer is automatically reloaded from the TVR-odd when the countdown expires (reaches zero). When this bit is off, the timer must be reprogrammed or restarted after each countdown.
5	PC system programmable timer count gate	This bit enables or disables timer counting and also allows reloading of the initial countdown from the value stored in the TVR-odd. When the bit is set to one, the timer is enabled and counting commences. When reset to zero, the timer is disabled, and decrementing of the timer count is halted. The countdown may be resumed by writing a one back to this bit, since the count contained in the timer is not changed when the gate bit is cleared. However, if a gate set is received when the gate bit is already on and timer count is at zero, the countdown value is reloaded from the TVR-odd and a full countdown begins.
4	PC system programmable timer overrun status	This bit is set when an overrun condition is detected with the PC system timer interrupt. If the timer interrupt has not been reset before the end of the next timing period, the overrun bit is set at the end of that period. Once set, this status bit remains active until reset to zero by the PC system.
3	PC system programmable timer count status	This bit is set by the adapter when the timer contains a nonzero countdown value (the timer is loaded but not necessarily counting). If this bit is a one, the nonzero value contained in the timer counter can be obtained by reading the TVR-even. Otherwise, reads to the TVR-even return zeros. When the timer countdown is halted by clearing of the gate bit and the count value is not zero, this bit will remain active (one). The PC system can only read this bit.
2	PC system interlock	This interlock allows the adapter's internal diagnostic routine to check the functional capability of the PC system timing facility without interference from the PC system. When set to one, this bit prevents PC system MMIO writes from updating the contents of the TVR-even and TVR-odd and the PC system portion (except this bit) of the TCR-even. This bit will be set only when the adapter's internal diagnostic procedures require exclusive use of the PC system programmable timer.
1	Reserved	
0	Adapter microprocessor timer control	Reserved

## The TCR-odd

BIT	NAME	DESCRIPTION
7-0	Adapter microprocessor timer control	Reserved

**Timer Value Register (TVR) Pair:** This register pair contains the PC system timer initial countdown value in the TVR-odd and the state of the PC system timer count in the TVR-even (referred to as "the timer.") For each byte, the possible values range from 10 milliseconds (X'01') to 2.55 seconds (X'FF') in 10-millisecond increments.

If the timer contains zeros, writing a byte to the TVR-odd will transfer that value to the timer. Counting is then subject to the state of the TCR-even gate bit. A read of the TVR-even will return the actual contents of the PC system timer counter at the time the read is received by the adapter. Writes to the TVR-even are ignored.

If the counter is loaded (nonzero), a write to the TVR-odd register will not cause the timer to be reloaded. The loading of the new TVR-odd value to the timer is governed by the state of the TCR-even gate and TCR-even reload bits.

The TCR-even count status bit and the TCR-even gate bit are used with the TVR-even. When the timer is loaded (the TCR-even count status bit is one), the value returned from the TVR-even is the actual timer count at the time of the read. If the TCR-even gate bit is one, then the counter will be counting and the value returned will reflect the current instantaneous counting state.

Reading the TVR-odd will always return the last value written to it (zero following initial power-on). Both TVR-odd and TVR-even are cleared after power-on reset.

#### TVR-even

BIT	NAME	DESCRIPTION
0-7	PC system programmable current count	As described above

#### TVR-odd

BIT	NAME	DESCRIPTION
0-7	PC system programmable timer initial count	As described above

**Shared RAM Page Register (SRPR) Pair:** This register pair only exists on the adapters that support RAM Paging. They are used to communicate to the adapter microcode the total amount of RAM to use, and they are also used to “page” the shared RAM into the PC system’s memory map.

Before the initialization of the adapter, the adapter support code must write the SRPR to the appropriate value in order to communicate to the adapter microcode how much total shared RAM to use. If the value of X'0000' is written to the SRPR, the adapter will use only the amount of RAM indicated by the shared RAM size bits in the RRR-odd register (bits 2 and 3). If the shared RAM size bits are set to the page size indicated in the ID PROM under the RAM Paging function, the PC system adapter support software can write X'C000' to the SRPR and the adapter microcode will use all 64K of shared RAM. The adapter support software can then access the entire 64K of shared RAM using RAM Paging.

If RAM Paging is selected, the SRPR can be used to “page” the PC system “window” into the full 64K of shared RAM after the adapter is initialized. See “Shared RAM Paging” on page 7-24 for more details.

#### SRPR-even

BIT	NAME	DESCRIPTION
7	Page select bit 1	This bit is set by the PC system to indicate total RAM or page selected.
6	Page select bit 0	This bit is set by the PC system to indicate total RAM or page selected.
0-5	Reserved	

#### SRPR-odd

BIT	NAME	DESCRIPTION
0-7	Reserved	



SRPR values and meanings are shown on the table below.

SRPR-Even	BEFORE Initialization	AFTER Initialization
X'00'	Use RRR(3,2) as total RAM	Map Area 1 into memory map
X'40'	Reserved	Map Area 2 into memory map
X'80'	Reserved	Map Area 3 into memory map
X'C0'	Use 64K as total RAM	Map Area 4 into memory map

SRPR Shared Ram Page Register	Total Ram/Page Selected															
	B1	B0														

TVR Timer Value	Current Count								Initial Count							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

TCR Timer Control	PC Timer						Adapter Microprocessor Timer					
	Program			Status			Reserved					
	M	R	G	O	C	SI						

ISRA ADPT. IRPT/STAT	I/F			Status		Mask		Software Encoded							
	P	T	A	D	M	H	S	7	6	5	4	3	2	1	0

ISRP PC IRPT/STAT	IRT		Interface			STAT		Software Encoded							
	SC	EN	T	E	A	B	PA	7	6	5	4	3	2	1	0

WRBR WWOR-WWCR RAM Management	PC Offset Address															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RRR RAM Relocation	Shared RAM Origin														
	19	18	17	16	15	14	13						3	2	

Bits 3 and 2 indicate Shared RAM size  
00=8K, 01=16K 10=32K, and 11=64K

Figure 7-2. Attachment Control Area Registers

PC I/O Bus Adapter

Register Pairs	Even 7.....0	Odd 7.....0
0 RRR	wwwwwwwo	000orroo
1 WRBR	rrrrrrrr	rrrrrrr0
2 WWOR	rrrrrrrr	rrrrrrr0
3 WWCR	rrrrrrrr	rrrrrrr0
4 ISRP	xxoyyyrr	0xxxxxx0
5 ISRA	rrrrrorr	0xxxxxxx
6 TCR	xxxyrr00	00000000
7 TVR	rrrrrrrr	wwwwwww
8 SRPR*	ww000000	00000000

Micro Channel Adapter

Register Pairs	Even 7.....0	Odd 7.....0
0 RRR	00000000	000orroo
1 WRBR	rrrrrrrr	rrrrrrr0
2 WWOR	rrrrrrrr	rrrrrrr0
3 WWCR	rrrrrrrr	rrrrrrr0
4 ISRP	xxoyyyor	0xxxxxx0
5 ISRA	rrrrrorr	0xxxxxxx
6 TCR	xxxyrr00	00000000
7 TVR	rrrrrrrr	wwwwwww
8 SRPR*	ww000000	00000000

Legend: 0 = Reserved, do not modify  
 R = Read-Only, No Write  
 W = Read-Write  
 X = Read-Write-Set-Reset  
 Y = Read-Reset  
 0 = Read-Only, always zero

Figure 7-3. Attachment Control Area Access Summary

ACA Byte Addressing									
Register Pairs		Even Byte				Odd Byte			
		Read	Write	Reset	Set	Read	Write	Reset	Set
0	RRR	1E00	1E00	--	--	1E01	1E01	--	--
1	WRBR	1E02	--	--	--	1E03	--	--	--
2	WWOR	1E04	--	--	--	1E05	--	--	--
3	WWCR	1E06	--	--	--	1E07	--	--	--
4	ISRP	1E08	1E08	1E28	1E48	1E09	1E09	1E29	1E49
5	ISRA	1E0A	--	--	--	1E0B	1E0B	1E2B	1E4B
6	TCR	1E0C	1E0C	1E2C	1E4C	1E0D	--	--	--
7	TVR	1E0E	--	--	--	1E0F	1E0F	--	--
8	SRPR*	1E18	1E18	--	--	1E19	--	--	--

Figure 7-4. Attachment Control Area Addressing Summary. \* This register is valid only on adapters that support shared RAM paging. See "Shared RAM Paging" on page 7-24.

---

## The Adapter Identification Area

This section describes the various fields in the adapter ID PROM (AIP) located on Token-Ring Network adapters. These fields are used to identify the adapter and indicate what functions are supported by each adapter. The information contained in the AIP is listed below.

- The adapter's encoded address
- The one's complement of the encoded address
- An identifier for PC I/O Bus or Micro Channel support
- A PROM checksum # 1
- A test pattern
- The adapter's supported functions
- A PROM checksum # 2

The AIP specifications should be used by any adapter support software that need to determine the adapter type, the supported functions, or the adapter's encoded address. This adapter support software could include test code, diagnostics, and adapter handlers.

The following information needs to be considered when using the AIP.

- All AIPs are 256 locations with each location being 4 bits wide (256 x 4 bit PROM).
- The AIP will be mapped into the BIOS/MMIO segment with offsets of 1F00-1FFF.
- The valid locations (where parameters will be retrieved) will be on **even** locations from the PC system.
- The valid bits are read from the four least significant bits of even bytes (PC system data bits 0-3).

**Note:** The upper four bits of each byte are not guaranteed to be zero and therefore should always be masked off.

- Adapter support software should only use the values in the AIP that have been defined. Reserved locations should not be used.

### Contents of the AIP

A typical AIP could have the following contents as seen from the PC system. This example is from a Token-Ring Network Adapter/A.

BIOS/MMIO Offset Address	AIP Contents															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1F00	01	00	00	00	00	00	00	00	05	00	0A	00	02	00	00	00
1F10	00	00	01	00	0E	00	03	00	0E	00	0F	00	0F	00	0F	00
1F20	0A	00	05	00	0D	00	0F	00	0F	00	0E	00	01	00	0C	00
1F30	04	00	0D	00	04	00	01	00	05	00	02	00	05	00	03	00
1F40	03	00	06	00	03	00	03	00	05	00	08	00	03	00	04	00
1F50	03	00	05	00	03	00	01	00	03	00	08	00	02	00	00	00
1F60	0B	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1F70	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1F80	00	00	02	00	04	00	06	00	08	00	0A	00	0C	00	0E	00
1F90	0F	00	0D	00	0B	00	09	00	07	00	05	00	03	00	01	00
1FA0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1FB0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1FC0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1FD0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1FE0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00
1FF0	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00	0F	00

Below is a list of all AIP fields.

**Adapter's Encoded Address** PC system even locations from X'1F00' to X'1F16'. The format is 12 nibbles from the even bytes with each nibble representing a hex digit. The most significant nibble (MSN) is the nibble at X'1F00', and the least significant nibble (LSN) is at X'1F16'.

**Example:** From the typical AIP above, read the even bytes from X'1F00' to X'1F16' as x1,x0,x0,x0,x5,xA,x2,x0,x0,x1,xE,x3. The resulting address is X'10005A2001E3'.

This field contains a 48-bit universally administered address as defined by the IEEE 802 committee on local area networks. Each IBM Token-Ring Network adapter is programmed with its own unique address in this field. This address will be used as the adapter's node address unless specifically overridden.

**One's Complement of the Adapter Encoded Address** The method for determining this value is the same method as for the address, but using the locations X'1F18' to X'1F2E'. X'1F18' is the MSN and X'1F2E' is the LSN. In the typical AIP above read xE,xF,xF,xF,xA,x5,xD,xF,xF,xE,x1,xC for a one's complement address of X'EFFFA5DFFE1C'.

This field contains the one's complement of the universally administered address as defined above.

**Channel Identifier** This field determines whether the adapter is one of the Token-Ring Network adapters with PC I/O Bus or one of the Token-Ring Network adapters with Micro Channel. It uses PC system even locations from X'1F30' to X'1F5E'. The format is 24 nibbles from the even bytes with each nibble representing a hex digit. The MSN is the nibble at X'1F30', and the LSN is at X'1F5E'.

From the example above read:

x4,xD,x4,x1,x5,x2,x5,x3,x3,x6,x3,x3,x5,x8,x3,x4,x3,x5,x3,x1,x3,x8,x2,x0 for a value of X'4D41 5253 3633 5834 3531 3820'

The two channel identifiers are defined below:

- X'5049 434F 3631 3130 3939 3020' is for Token-Ring Network adapters with PC I/O Bus.

- X'4D41 5253 3633 5834 3531 3820' is for Token-Ring Network adapters with Micro Channel.

**AIP Checksum # 1** The checksum is a PC system even location of X'1F60'. The format is a hex nibble.

This field contains the first checksum for the AIP. If a 4-bit checksum (addition) of all valid (even) locations from 1F00 up to and including 1F60 is calculated, the result should be zero. This checksum is used to validate the encoded address and channel identifier. If an invalid checksum is obtained (non-zero value), then the previous values should be considered inaccurate.

**Test Pattern** The test pattern is PC system even and odd locations X'1F80' to X'1F9F'. The format is 32 nibbles from all bytes with each nibble representing a hex digit. In the example above read:

```
x0,x0,x2,x0,x4,x0,x6,x0,
x8,x0,xA,x0,xC,x0,xE,x0,
xF,x0,xD,x0,xB,x0,x9,x0,
x7,x0,x5,x0,x3,x0,x1,x0.
```

This field contains a test pattern used during adapter diagnostics.

**Supported Functions Identifiers** These identifiers are PC system even locations from X'1FA0' to X'1FEE'. The format is a hex nibble from the even bytes. The example above reads xF,xF,xF . . .

These nibbles should be used to determine what functions the adapter supports. The adapter support software should read these nibbles to determine the capabilities of each specific adapter. The nibbles from the hex locations are defined as follows.

**X'1FA0'** Adapter Type where: F = 1st, E = 2nd, D = 3rd, .... 0 = 16th.

This field is used to identify different adapters within a given I/O bus or channel type. For example, the Token-Ring Network Adapter/A uses X'F'.

**X'1FA2'** Data Rate where: F = 4 Mbps, E = 16 Mbps, D = 4 & 16 Mbps, C to 0 = reserved.

This field is used to identify the Data Rates supported by the adapter.

**X'1FA4'** Early Token Release where F = No, E = 4 Mbps, D = 16 Mbps, C = 4/16 Mbps, B to 0 = reserved.

This field is used to identify which Data Rates support Early Token Release.

**X'1FA6'** Total available shared RAM where F = use RRR(3,2), E = 8K, D = 16K, C = 32K, B = 64K (top 512 reserved and must be set to zero), A = 64K (top 512 usable) 9-0 = reserved.

This field is used to identify the total shared RAM installed on the adapter. Either use the encoded value in the RRR register, or use the specified value in the AIP. For value "B," the last 512 bytes (offset address FE00-FFFF) are reserved and must be set to '0' during Initialization of the adapter.

**X'1FA8'** Shared RAM Paging where F = No, E = 16K page, D = 32K page, C = 16K & 32K page, B-0 = reserved.

This field is used to identify whether or not the adapter supports shared RAM Paging and if so, at which page sizes.

**X'1FAA'** The DHB size available at 4 Mbps where F = 2048, E = 4096, D = 4464, C-0 = reserved.

This field is used to identify the maximum DHB size at a 4 Mbps Data Rate (N/A if adapter does not support 4 Mbps).

**X'1FAC'** The DHB size available at 16 Mbps where F = 2048, E = 4096, D = 8192, C = 16384, B = 17960, A-0 = reserved.

This field is used to identify the maximum DHB size at a 16 Mbps Data Rate (N/A if adapter does not support 16 Mbps).

**X'1FAE' to X'1FEE'** These locations are reserved.

**AIP Checksum # 2** A PC system even location of X'1FF0'. The format is a hex nibble.

This field contains the second checksum for the AIP. If a 4-bit checksum (addition) of all valid (even) locations from 1F00 up to and including 1FF0 is calculated, the result should be 0. This checksum is used to validate all defined areas of the AIP. If an invalid checksum is obtained (non-zero value), then the previous values should be considered inaccurate. If an incorrect value is encountered, the only values used should be "default" values for the type adapter identified by the channel identifier field.

**Reserved** All undefined locations (both even and odd) are reserved and should not be used.

## Shared RAM

### Shared RAM Layout

The adapter will assign locations in shared RAM when the adapter is opened in a format such as this:

#### Beginning of shared RAM (as seen by the adapter)

PC read-only address space	Adapter private variables and work areas Length: 1496 bytes
	System status block (SSB) Length: 20 bytes
	Adapter request block (ARB) Length: 28 bytes
	Receive buffers Length: space remaining after all SAPs/Stations are defined
	SAP and link station control blocks Length: as defined by maximum number of SAPs/Stations
PC read/write address space	Data holding buffer (DHB) Length: as specified at open adapter time. There may be one or more DHBs.
	System request block (SRB) Length: 28 bytes
	Adapter status block (ASB) Length: 12 bytes
Reserved area on 64K Shared RAM adapters	Length: 512 bytes

#### End of shared RAM (as seen by the adapter)

**Note:** On 64K adapters, the 512 bytes at the end is reserved.

If the bridge function is used, shared RAM is formatted with an additional area. See "The Bridge Functions" on page 7-64 for shared RAM layout and a description of the bridge functions available.

## Shared RAM Paging

Shared RAM Paging is a technique that allows the PC system software to access all the RAM on the adapter, without having to map the entire shared RAM into the PC system's memory map. The shared RAM on the adapter is paged into the PC system's memory map one area at a time.

Shared RAM paging is only available on the adapters that indicate that function within their ID PROM and only when they have the indicated amount of shared RAM mapped into the PC system memory map. Shared RAM paging is controlled by the PC system software using the shared RAM page register and the information in the shared RAM size bits in RRR-odd (bits 2 and 3).

The PC system adapter support software must follow these steps in order to use RAM Paging:

1. Determine if the adapter has the appropriate amount of shared RAM mapped into the PC system's memory map using information in the ID PROM and the shared RAM size bits (bits 2 and 3) in the RRR-odd register.
2. Set the SRPR bits to the desired value (X'C0' for Paging) before issuing the Initialize command.
3. Initialize the adapter to indicate to the adapter microcode the desired amount of "total shared RAM" to use.

The adapter's microcode uses the total shared RAM to determine where buffers, control blocks, and other information is placed in the adapter's shared RAM.

4. Use the SRPR to page to the desired section of shared RAM as required for operation.

Once the adapter has been initialized for RAM Paging, the SRPR should be used to page the PC system's mapped memory into the appropriate area of shared RAM as shown in the example below.

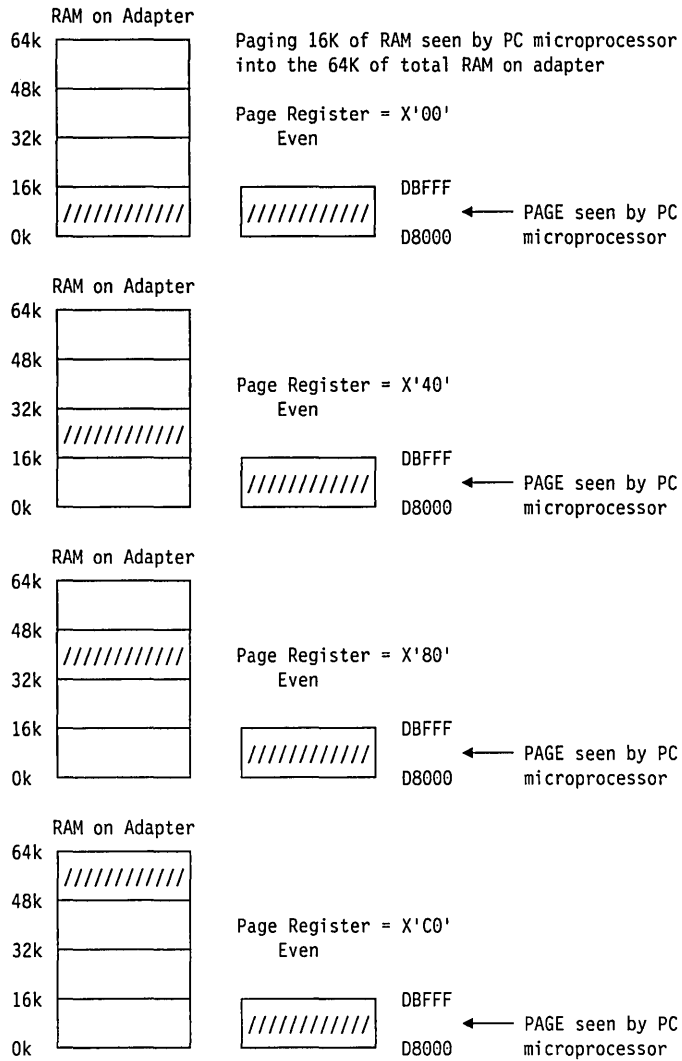


Figure 7-5. Shared RAM Paging after Initialization

## Uninitialized Shared RAM

All adapters that offer 64K of shared RAM and indicate in their AIP a "Total available shared RAM=B" need a portion of the shared RAM initialized to all zeros.

The area of uninitialized shared RAM is illustrated below. This 512 bytes must be written to all zeros after initialization but before opening the adapter.



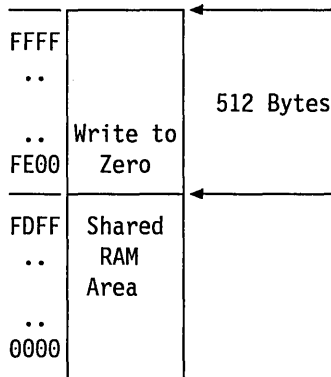


Figure 7-6. Uninitialized Shared RAM

## Operating the Adapter

The adapter must be enabled and initialized before any commands may be processed. Initializing the adapter is performed using PIO and MMIO operations. Commands thereafter are performed using read and write memory instructions.

Communication between the adapter and the PC system is by means of interrupts initiated in registers in the MMIO region and by control blocks in the shared RAM.

There are four control blocks used to pass requests and the status of requests between the PC system and the adapter.

**The system request block (SRB)** The SRB is used to pass a command from the PC system to the adapter. If the command is completed upon receipt by the adapter, either successfully or with an error, the return code for the command will be passed back to the PC system in the SRB with an interrupt raised. If further processing is required by the adapter, a return code of X'FF' and a command correlator will be placed in the SRB, but no interrupt to the PC system will result unless the SRB free request is set. The adapter will later update the system status block (SSB) with status related to that command and interrupt the PC system. When the SRB is "filled in," the return code field must be set to X'FE' by the PC system.

The SRB can be initially located by using the address in the write region base register (WRBR) in the attachment control area of the MMIO domain. When the adapter is opened and the future SRB location is indicated by the SRB response to a successful OPEN.ADAPTER SRB. The following three commands change the location of the SRB: DIR.CLOSE.ADAPTER, DIR.CONFIGURE.BRIDGE.RAM, and DIR.OPEN.ADAPTER. The SRB address is also changed if the adapter closes automatically due to an error condition.

Initially the SRB will be large enough to contain the 60 bytes (X'3C') needed to issue an DIR.OPEN.ADAPTER command, but will thereafter be 28 bytes (X'1C') in length. The SRB location after an DIR.OPEN.ADAPTER command is issued will be returned in the SRB upon completion of the DIR.OPEN.ADAPTER command.

**The system status block (SSB)** The SSB is used to pass the results of an SRB command to the PC system when the SRB has been returned initially with an in-process return code X'FF'. If multiple commands of the same type are outstanding, the Station ID and command correlator provided in the SRB with the X'FF' return code may be used to identify the command(s) being completed.

The location of the SSB is returned by the adapter in the SRB upon completion of a DIR.OPEN.ADAPTER command.

**The adapter request block (ARB)** The ARB is used by the adapter to pass information or issue a command to the PC system.

If information is passed with the ARB, no response is expected other than an indication that the information has been read and the ARB is available for reuse by the adapter.

If a command is passed with the ARB, a response is expected from the PC system in the adapter status block (ASB) when the command is complete.

The location of the ARB is returned by the adapter in the SRB upon completion of an DIR.OPEN.ADAPTER command.

**The adapter status block (ASB)** The ASB is used by the PC system to respond to a command received from the adapter in the ARB. The response may indicate either successful completion of the command or that an error has occurred.

The location of the ASB is returned by the adapter in the SRB upon completion of an DIR.OPEN.ADAPTER command. The return code field of the ASB is initialized to X'FF' by the adapter when the DIR.OPEN.ADAPTER command is completed.

The method of initializing and obtaining the results is:

1. Issue an Adapter Reset PIO command.
2. Delay for at least 50 milliseconds.
3. Issue an Adapter Release PIO operation.
4. Set the interrupt enable bit (bit 6) in ISRP-even.
5. Read the write region base register (WRBR) contents in response to the interrupt initiated by the adapter setting bit 5 in ISRP-odd. Currently this takes between one and three seconds depending on adapter type.
6. Use the offset in the WRBR and the shared RAM segment address to obtain address the initial location of the SRB where the adapter will have posted the results of the initialization.

The SRB will contain the following response:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'80', Initialization complete
1	INIT_STATUS	1	The bit of this INIT_STATUS byte have the following meanings. <ul style="list-style-type: none"> <li>• Bits 0 through 5 are reserved.</li> <li>• Bit 6, if on, indicates open option bit is supported by this adapter.</li> <li>• Bit 7 indicates the adapter data rate. If bit 7 is 0 the adapter data rate is 4Mbps and if it is 1 the data rate is 16Mbps.</li> </ul>
2		4	Reserved
6	BRING_UP_CODE	2	The bring-up diagnostics result code
8	ENCODED_ADDRESS	2	Shared RAM address offset of the adapter's encoded address
10	LEVEL_ADDRESS	2	Shared RAM offset to the adapter microcode level
12	ADAPTER_ADDRESS	2	Shared RAM address offset of the adapter addresses
14	PARMS_ADDRESS	2	Shared RAM address offset of the adapter parameters
16	MAC_ADDRESS	2	Shared RAM address offset of the adapter MAC buffer

---

### BRING\_UP\_CODE

**Explanation:** One of the following codes will be provided to indicate the results of the bring-up tests.

Code	8086 Type	Meaning
'0020'	DW	Diagnostics could not execute
'0022'	DW	ROM (ROS) diagnostics failed
'0024'	DW	Shared RAM diagnostics failed
'0026'	DW	Processor instruction test failed
'0028'	DW	Processor interrupt test failed
'002A'	DW	Shared RAM interface register diagnostics failed
'002C'	DW	Protocol-handler diagnostics failed
'0040'	DW	Adapter's programmable timer for the PC system failed (set by the PC system code)
'0042'	DW	Cannot write to shared RAM (set by the PC system code)
'0044'	DW	Reading from shared RAM read-only area caused an invalid error indication (interrupt) (set by the PC system code)
'0046'	DW	Writing into shared RAM read-only area did not cause an error indication (interrupt) (set by the PC system code)
'0048'	DW	Initialization timed out

### ADAPTER\_ADDRESS

**Explanation:** This parameter provides the shared RAM offset to the following information. The NODE\_ADDRESS is accessible as long as the adapter is initialized and/or open. The GROUP\_ADDRESS and FUNCTIONAL\_ADDR are not valid until after the adapter is open.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	NODE_ADDRESS	6	Adapter node address
6	GROUP_ADDRESS	4	Adapter group address
10	FUNCTIONAL_ADDR	4	Adapter functional address

### PARMS\_ADDRESS

**Explanation:** This parameter provides the shared RAM offset to the following information. This information is accessible as long as the adapter is initialized and/or open.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	PHYS_ADDR	4	Adapter physical address
4	UP_NODE_ADDR	6	Next active upstream node address
10	UP_PHYS_ADDR	4	Next active upstream physical address
14	POLL_ADDR	6	Last poll address
20		2	Reserved
22	ACC_PRIORITY	2	Transmit access priority
24	SOURCE_CLASS	2	Source class authorization
26	ATT_CODE	2	Last attention code
28	SOURCE_ADDR	6	Last source address
34	BEACON_TYPE	2	Last beacon type
36	MAJOR_VECTOR	2	Last major vector
38	NETW_STATUS	2	Network status

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
40	SOFT_ERROR	2	Soft error timer value
42	FE_ERROR	2	Front end error counter
44	LOCAL_RING	2	Ring number
46	MON_ERROR	2	Monitor error code
48	BEACON_TRANSMIT	2	Beacon transmit type
50	BEACON_RECEIVE	2	Beacon receive type
52	FRAME_CORREL	2	Frame correlator save
54	BEACON_NAUN	6	Beaconing station NAUN
60		4	Reserved
64	BEACON_PHYS	4	Beaconing station physical address

---

## Commands

There are three general categories of PC system-to-adapter commands:

- Direct
- DLC (IEEE 802.2 SAP and station interfaces)
- Data transmission.

These commands have certain qualities in common:

1. The command request is made by loading information in the SRB and setting ISRA-odd bit 5.
2. The adapter checks the validity of the SRB contents and either:
  - Completely processes the command, sets a return code other than X'FF', and interrupts the PC system by setting ISRP-odd bit 5, or
  - Performs initial processing only, sets the return code to X'FF', and provides a command correlator. ISRP-odd bit 5 will be set only if an SRB Free Request Interrupt is initiated by the PC system setting bit 3 of ISRA odd.
3. Depending on the command the adapter may request further data using the ARB and DHB. The PC system will use the ASB to indicate that the requested data has been moved.
4. When processing for a request that is in process (return code = X'FF') is complete, the adapter will put the final return code in the SSB and interrupt the PC system by setting ISRP-odd bit 2.
5. After the PC system has read the return code from the SSB, it will interrupt the adapter by setting ISRA-odd bit 0.

## Adapter SRB Commands

The commands placed into the SRB by the PC system are similar to those provided to the adapter support software. The SRB is also used in a similar format to the CCB used with the direct interface of the adapter support software.

## Direct Interface Commands

The following commands, listed in alphabetical order, may be issued by the application program to the adapter. These commands affect the state of the adapter as a whole, rather than specific SAPs or link stations, and do not involve LLC processing.

The adapter must have been successfully initialized before any of these commands can be performed. After initialization, or a successful DIR.CLOSE.ADAPTER command, the only acceptable commands are DIR.OPEN.ADAPTER, DIR.CLOSE.ADAPTER, DIR.INTERRUPT, and DIR.CONFIG.BRIDGE.RAM. After successful completion a DIR.OPEN.ADAPTER, any of the other direct interface commands may be issued.

All direct interface commands will be returned with ISRP-odd low bit 5 set and return information located in the SRB. Return code X'FF' (in-process) is never set for these commands.

COMMAND NAME	CODE
DIR.CLOSE.ADAPTER	X'04'
DIR.INTERRUPT	X'00'
DIR.MODIFY.OPEN.PARMS	X'01'
DIR.OPEN.ADAPTER	X'03'
DIR.READ.LOG	X'08'
DIR.RESTORE.OPEN.PARMS	X'02'
DIR.SET.FUNCTIONAL.ADDRESS	X'07'
DIR.SET.GROUP.ADDRESS	X'06'

A description of the SRB content for each of the commands follows. The command is explained and the fields provided by the PC system and those returned by the adapter are shown.

See "The Bridge Functions" on page 7-64 for the direct interface commands used for bridge functions.

## DIR.CLOSE.ADAPTER

Hex 04

**DIR.CLOSE.ADAPTER**

**Command Description:** Close the adapter and terminate all ring communication or the "open wrap test."

**Command Specifics:** This command will be accepted anytime after the adapter has been initialized. Commands that have been accepted by the adapter and not completed will remain incomplete and will not be returned to the PC system. The adapter will be removed from the ring, if it was active, and the write region base register (WRBR) will be reset to the value set prior to the OPEN.ADAPTER command.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'04', DIR.CLOSE.ADAPTER
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

X'00' Operation completed successfully

X'01' Invalid command code

## DIR.INTERRUPT

Hex 00

DIR.INTERRUPT

**Command Description:** Force an adapter interrupt. This command performs no function.

**Command Specifics:** The adapter must have been initialized, but does not have to be opened for this command to be accepted.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'00', DIR.INTERRUPT
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return

When the adapter detects this SRB, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

X'00' Operation completed successfully

X'01' Invalid command code

## DIR.OPEN.ADAPTER

## DIR.MODIFY.OPEN.PARMS

Hex 01

### DIR.MODIFY.OPEN.PARMS

**Command Description:** Used to modify the OPEN\_OPTIONS set by the DIR.OPEN.ADAPTER command.

**Command Specifics:** The wrap option, remote program load, and token release bits will be ignored.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'01', DIR.MODIFY.OPEN.PARMS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	OPEN_OPTIONS	2	New options (wrap bit left unaltered in adapter)

See the DIR.OPEN.ADAPTER command for a description of the OPEN\_OPTIONS parameter.

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

**X'00'** Operation completed successfully

**X'01'** Invalid command code

**X'04'** Adapter closed, should be open

## DIR.OPEN.ADAPTER

Hex 03

### DIR.OPEN.ADAPTER

**Command Description:** Prepare the adapter for either normal ring communication or an adapter wrap test.

**Command Specifics:** This command will be accepted after successful initialization of the adapter. Once an open adapter command has been completed successfully, the adapter must be closed or reset before another open adapter command will be accepted. After this command has been returned with a X'00' return code, the adapter will be in automatic receive mode and frames may be transmitted and received using the direct interface. DLC interface commands may also be issued.

The information provided along with this command is used to configure shared RAM. See "Shared RAM Layout" on page 7-23. Space is allocated for:

- The adapter work areas
- The communication areas
- The requested individual and group SAP control blocks
- The requested link station control blocks
- The requested number of DHBs.

The remaining shared RAM space will be configured as receive buffers using the supplied receive buffer length parameter. The adapter will then check that the number of available receive buffers is equal to or

greater than the number requested. If the number of receive buffers is inadequate, the open adapter command will be rejected. See "Additional Adapter Functions" on page 7-64 for more about shared RAM space and buffers.

### Length of SRB

The SRB in shared RAM is defined as 28 bytes in length and all PC system commands to the adapter except the DIR.OPEN.ADAPTER require 28 or fewer bytes. The SRB after initialization and before an open command has been completed starts at the initial write region base address and can accept enough information for the open parameters.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'03', DIR.OPEN.ADAPTER
1		7	Reserved
8	OPEN_OPTIONS	2	Open options, See description
10	NODE_ADDRESS	6	This adapter's ring address
16	GROUP_ADDRESS	4	The group address to set
20	FUNCT_ADDRESS	4	The functional address to set
24	NUM_RCV_BUF	2	Number of receive buffers
26	RCV_BUF_LEN	2	Length of receive buffers
28	DHB_LENGTH	2	Length of transmit buffers
30	NUM_DHB	1	Number of DHBs
31		1	Reserved
32	DLC_MAX_SAP	1	Maximum number of SAPs
33	DLC_MAX_STA	1	Maximum number of link stations
34	DLC_MAX_GSAP	1	Maximum number of group SAPs
35	DLC_MAX_GMEM	1	Maximum members per group SAP
36	DLC_T1_TICK_ONE	1	DLC timer T1 interval, group one
37	DLC_T2_TICK_ONE	1	DLC timer T2 interval, group one
38	DLC_Ti_TICK_ONE	1	DLC timer Ti interval, group one
39	DLC_T1_TICK_TWO	1	DLC timer T1 interval, group two
40	DLC_T2_TICK_TWO	1	DLC timer T2 interval, group two
41	DLC_Ti_TICK_TWO	1	DLC timer Ti interval, group two
42	PRODUCT_ID	18	Product Identification

### OPEN\_OPTIONS

**Explanation:** Various options, each defined by a bit. A bit on (1) indicates that the option is to be taken. Bit 15 is the high-order (leftmost) bit.

**Bit 15 - Wrap interface** The adapter will not attach itself to the network. Instead it will cause all user transmit data to be wrapped as received data.

**Bit 14 - Disable Hard Error** Prevents network status changes involving "Hard Error" and "Transmit Beacon" bits from causing interrupts.

**Bit 13 - Disable Soft Errors** Prevents network status changes involving "Soft Error" bit from causing interrupts.

**Note:** This bit is not applicable to the PC Network



## DIR.OPEN.ADAPTER

**Bit 12 - Pass Adapter MAC Frames** Pass, as direct interface data to the PC system, all adapter class MAC frames that are received but not supported by the adapter. If this bit is off, these frames will be ignored.

**Bit 11 - Pass Attention MAC Frames** Pass, as direct interface data to the PC system, all attention MAC frames that are not the same as the last received attention MAC frame. If this option is off, these frames will not be passed to the application program.

**Bit 10 - Reserved** Should be zero.

**Bit 8 - Contender** When the contender bit is on, the adapter will participate in monitor contention (claim token) if the opportunity occurs. When the contender bit is off, and the need is detected by another adapter, this adapter will not participate.

If the need for determining a new active monitor is detected by this adapter, monitor contention (claim token) processing will be initiated by this adapter in either case.

**Bit 7 - Pass Beacon MAC frames** Pass, as direct interface data to the PC system, the first beacon MAC frame and all subsequent beacon MAC frames that have a change in the source address or the beacon type.

**Bit 6 Reserved** Should be zero, but is not checked.

**Bit 5 - Remote Program Load** This bit is only implemented in 16/4 adapters. It prevents the adapter from becoming a monitor during the open process. If this bit is on, the adapter will fail the open process if there is no other adapter on the ring when it attempts to insert on the ring.

**Bit 4 - Token Release** This bit is only available for 16/4 adapters when operating at 16 Mbps. If not set, 16Mbps adapters will get early token release; that is the default. Setting this bit on, selects no early token release for an adapters operating at 16Mbps.

**Bits 0 through 3 - Reserved** Should be zero, but are not checked by the adapter support software.

---

### NODE\_ADDRESS

**Explanation:** The 6-byte specific node address of this station on the ring. The high-order (leftmost) bit must be zero. If the value is zero, the adapter's encoded address will be the node address by default.

---

### GROUP\_ADDRESS

**Explanation:** Sets the group address that the adapter will receive messages for. If the value is zero, no group address is set. The group address can also be set, or changed, by a SET.GROUP.ADDRESS command.

---

### FUNCT\_ADDRESS

**Explanation:** Sets the functional address that the adapter will receive messages for. Bits 31, 1, and 0 are ignored. If the value is zero, no functional address is set. The functional address can also be set, or changed, by a SET.FUNCT.ADDRESS command. The functional address is also affected by the DIR.CONFIG.BRIDGE.RAM command.

---

### NUM\_RCV\_BUF

**Explanation:** The number of receive buffers in shared RAM needed for the adapter to open. The adapter will configure as receive buffers all remaining shared RAM after other memory requirements have been met. If the number available is less than the number requested, the DIR.OPEN.ADAPTER command will fail. If the number available is greater than the number requested, no action will occur. If this value is less than 2, the default of 8 will be used.

---

**RCV\_BUF\_LEN**

**Explanation:** The length of each of the receive buffers in the shared RAM. Receive buffers will be chained together to hold a frame that is too long for one buffer. However, only one frame will be put into a single buffer.

The value must be a multiple of 8; 96 is the minimum and 2048 is the maximum. If the value is zero, the default of 112 will be used. Each buffer holds 8 fewer bytes of data than the specified size. Therefore, a buffer defined as 112 bytes long can hold only 104 bytes of data. The 8 bytes are overhead needed by the adapter.

---

**DHB\_LENGTH**

**Explanation:** The length of each of the transmit buffers in the shared RAM. Only one buffer is used to hold transmit data, including header information, for a given frame for the direct interface and SAP interface. For the link station interface, this length applies to the information field of I frames.

The value must be a multiple of 8 with 96 as minimum.

For the following Token-Ring Network adapters the maximum DHB size is 2048 bytes.

- Token-Ring Network PC Adapter
- Token-Ring Network PC Adapter II
- Token-Ring Network Adapter/A

For all 16/4 Token-Ring Network adapters operating at 4Mbps the maximum DHB size will be 4464.

For all 16/4 Token-Ring Network adapters operating at 16Mbps, the maximum DHB size will be 17960 bytes.

**Note:** If a size greater than 2048 is used, it is important to make sure that all adapters receiving these frames will also be able to handle the larger size.

If the value is zero, the default of 600 will be used. Each buffer holds 6 fewer bytes of data than the specified size. Therefore, a buffer defined as 600 can hold only 594 bytes.

---

**NUM\_DHB**

**Explanation:** This defines the number of transmit buffers in the adapter shared RAM in which the data from the PC system may be stored.

The adapter will accept any value between zero and 255, but the integrity of adapter operation cannot be guaranteed if the value is greater than 2. Requesting two buffers may improve adapter performance by allowing a frame to be moved into the second buffer while the adapter is transmitting from the first. However, this will reduce the storage available for receive buffers. If the value is zero, the default of 1 will be used.

---

**DLC\_MAX\_SAP**

**Explanation:** The maximum number of individual SAPs that can be opened at one time. The maximum value allowed is 126. Each individual SAP control block requires 64 bytes of shared RAM. If this parameter is set to zero, no open SAP commands will be accepted and the DLC SAP and the DLC link station interfaces will not be available. However, the null and the global SAPs will be activated.

---

**DLC\_MAX\_STA**

**Explanation:** The maximum number of link stations that can be opened at one time. It does not determine the number of link stations that may be open for any one SAP. Each link station control block requires 144 bytes of shared RAM. If this parameter is not zero, the DLC\_MAX\_SAP parameter must not be zero.

## DIR.OPEN.ADAPTER

---

### DLC\_MAX\_GSAP

**Explanation:** The maximum number of group SAPs that can be opened at one time. Each group SAP control block requires 14 bytes plus two times the DLC\_MAX\_GMEM parameter value in shared RAM. If the value is zero, no group SAPs are allowed, but the global SAP will be activated.

The corresponding individual SAP control block, requiring 64 bytes, is required in order to open a group SAP. That is, group SAP X'05' requires that individual SAP X'04' must also be allocated.

---

### DLC\_MAX\_GMEM

**Explanation:** The maximum number of SAPs that can be assigned to any given group. This parameter will be ignored if the DLC\_MAX\_GSAP parameter is zero and may not be zero if that field is not zero.

---

### DLC\_T1\_TICK\_ONE

**Explanation:** The number of 40-millisecond intervals that make up a "tick" for DLC timer T1 (T1 timer values 1 - 5). If the value is zero, the default of 5 (200 - 400 milliseconds) is used.

---

### DLC\_T2\_TICK\_ONE

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for DLC timer T2 (T2 timer values 1 - 5). If the value is zero, the default of 1 (40 - 80 milliseconds) is used.

---

### DLC\_TI\_TICK\_ONE

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for DLC timer Ti (Ti timer values 1 - 5). If the value is zero, the default of 25 (1 - 2 seconds) is used.

---

### DLC\_T1\_TICK\_TWO

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for DLC timer T1 (timer values 6 - 10). If the value is zero, the default of 25 (1 - 2 seconds) is used.

---

### DLC\_T2\_TICK\_TWO

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for DLC timer T2 (timer values 6 - 10). If the value is zero, the default of 10 (400 - 800 milliseconds) is used.

---

### DLC\_TI\_TICK\_TWO

**Explanation:** The number of 40-millisecond intervals between timer "ticks" for DLC timer Ti (timer values 6 - 10). If the value is zero, the default of 125 (5 - 10 seconds) is used.

---

### PRODUCT\_ID

**Explanation:** This is the PC system 18-byte product ID.

**SRB Response**

When the adapter completes the open command, bytes 6 through 15 in the SRB are set with return parameters and the return code is placed in the RETCODE field. The adapter then interrupts the PC system by setting ISRP-odd bit 5.

The SRB content will then be as follows.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'03', DIR.OPEN.ADAPTER
1		1	Reserved
2	RETCODE	1	Return code, see below
3		3	Reserved
6	OPEN_ERROR_CODE	2	Valid if RETCODE is X'07'. See "Token-Ring Network Adapter Open Errors for All CCBs" on page B-49.
8	ASB_ADDRESS	2	Offset to the beginning of the ASB
10	SRB_ADDRESS	2	Offset to the beginning of the SRB
12	ARB_ADDRESS	2	Offset to the beginning of the ARB
14	SSB_ADDRESS	2	Offset to the beginning of the SSB

**Valid Return Codes:**

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'03' Adapter open, should be closed
- X'05' Required parameter(s) not provided
- X'07' Command canceled, unrecoverable failure
- X'30' Inadequate receive buffers for adapter to open
- X'32' Invalid NODE\_ADDRESS
- X'33' Invalid adapter receive buffer length defined
- X'34' Invalid adapter transmit buffer length defined

**DIR.READ.LOG**

Hex 08

DIR.READ.LOG

**Command Description:** Read and reset the adapter error counters.

**Command Specifics:** This command should be issued if a ring status change ARB is received with the counter overflow set. This ARB is issued if one of the adapter error counters reaches a count of 255. The adapter will accept this command anytime after the adapter is opened and before a close adapter command is issued.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'08', DIR.READ.LOG
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		3	Reserved

## DIR.RESTORE.OPEN.PARMS

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
6	LOG_DATA	14	14 bytes of log data set by the adapter

### Adapter Error Counters

Refer to the *IBM Token-Ring Network Architecture Reference* for more about these error counters.

<b>Byte 0</b>	Line errors
<b>Byte 1</b>	Internal errors
<b>Byte 2</b>	Burst errors
<b>Byte 3</b>	A/C errors
<b>Byte 4</b>	Abort delimiters
<b>Byte 5</b>	Reserved
<b>Byte 6</b>	Lost frames
<b>Byte 7</b>	Receive congestion count
<b>Byte 8</b>	Frame copied errors
<b>Byte 9</b>	Frequency errors
<b>Byte 10</b>	Token errors
<b>Byte 11</b>	Reserved
<b>Byte 12</b>	Reserved
<b>Byte 13</b>	Reserved

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

### Valid Return Codes:

X'00'	Operation completed successfully
X'01'	Invalid command code
X'04'	Adapter closed, should be open

## DIR.RESTORE.OPEN.PARMS

Hex 02

### DIR.RESTORE.OPEN.PARMS

**Command Description:** Used to modify the OPEN\_OPTIONS set by the DIR.OPEN.ADAPTER command.

**Command Specifics:** The wrap option, remote program load, and modified token release bits will be ignored.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'02', DIR.RESTORE.OPEN.PARMS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	OPEN_OPTIONS	2	New options (wrap bit left unaltered in adapter)

See the DIR.OPEN.ADAPTER command for a description of the OPEN\_OPTIONS parameter.

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open

**DIR.SET.FUNCT.ADDRESS**

Hex 07

**DIR.SET.FUNCT.ADDRESS**

**Command Description:** Set the functional address for the adapter to receive messages.

**Command Specifics:** If this command is issued with the FUNCT\_ADDRESS field containing all zeros, any previously set functional address will be disabled. Bits 31, 1, and 0 will be ignored. The adapter will accept this command anytime after the adapter is opened and before a close adapter command is issued. See the DIR.CONFIGURE.BRIDGE.RAM command, which can also alter the functional address.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'07', DIR.SET.FUNCTIONAL ADDRESS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		3	Reserved
6	FUNCT_ADDRESS	4	New functional address to set

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open

**DIR.SET.GROUP.ADDRESS**

Hex 06

**DIR.SET.GROUP.ADDRESS**

**Command Description:** Set the group address for the adapter to receive messages.

**Command Specifics:** The adapter will accept this command anytime after the adapter is opened and before a close adapter command is issued.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'06', DIR.SET.GROUP.ADDRESS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return

## DIR.SET.GROUP.ADDRESS

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
3		3	Reserved
6	GROUP_ADDRESS	4	New group address to set

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

### Valid Return Codes:

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open

## DLC SRB Commands

The following commands, listed in alphabetical order, may be issued by the application program to the adapter. These commands apply to service access points (SAPs) and link stations, and make use of LLC protocols. Some of these commands apply only to the SAP interface (DLC.OPEN.SAP and DLC.CLOSE.SAP), and some apply only to the station interface (DLC.OPEN.STATION, DLC.CONNECT.STATION, DLC.CLOSE.STATION, DLC.STATISTICS). The remainder apply to both interfaces.

Refer to Chapter 2, "How to Write Programs for Local Area Networks" for more details about Logical Link Control.

The adapter must have been initialized and opened with direct interface commands before any of these commands will be accepted by the adapter.

COMMAND NAME	CODE
DLC.CLOSE.SAP	X'16'
DLC.CLOSE.STATION	X'1A'
DLC.CONNECT.STATION	X'1B'
DLC.FLOW.CONTROL	X'1D'
DLC.MODIFY	X'1C'
DLC.OPEN.SAP	X'15'
DLC.OPEN.STATION	X'19'
DLC.RESET	X'14'
DLC.STATISTICS	X'1E'

## DLC.CLOSE.SAP

Hex 16

### DLC.CLOSE.SAP

**Command Description:** Close (deactivate) a service access point (SAP) and free the associated control block(s).

**Command Specifics:** This command will be rejected if any links are open for the specified SAP, or if the SAP was opened with the group option specified and there are any active members in the group. If the specified SAP is a group member, its membership should be canceled using a DLC.MODIFY command prior to this command being issued. If an adapter command to the PC system is outstanding for the specified SAP when the DLC.CLOSE.SAP command is issued, the PC system must complete that action before this command will be completed.

**Note:** If a X'47' error code results when a DLC.CLOSE.SAP command closely follows a DLC.CLOSE.STATION command for the last open station for that SAP, reissue the DLC.CLOSE.SAP command.

Any frames directed to the specified SAP that have been received by the adapter and for which the adapter has not posted a receive ARB will be discarded.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'16', DLC.CLOSE.SAP
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the SAP to be closed

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

- X'00'** Operation completed successfully
- X'01'** Invalid command code
- X'04'** Adapter closed, should be open
- X'40'** Invalid STATION\_ID
- X'47'** SAP cannot close unless all link stations are closed
- X'48'** Group SAP cannot close until all member SAPs are closed
- X'4C'** Unable to close, commands outstanding



## DLC.CLOSE.STATION

## DLC.CLOSE.STATION

Hex 1A

### DLC.CLOSE.STATION

**Command Description:** Close one link station. This command will not complete while the ring is beaconing.

**Command Specifics:** The link control block will be freed for use by another link station on the same SAP.

This command will be rejected if there is a DLC.CLOSE.STATION or a DLC.CONNECT.STATION command outstanding for the specified link station. If the command is accepted, the adapter will either:

- Transmit a DISC command to the remote station and enter disconnecting mode while waiting for an acknowledgment, or
- Send a DM response if there is a SABME or DISC command outstanding, or if the link is in the disconnecting state, and close the link station when the response has been transmitted.

If there are outstanding Transmit I Frame requests when this command is accepted, they will not be returned by the adapter. If an adapter command to the PC system is outstanding for the specified Link Station when DLC.CLOSE.STATION command is issued, the PC system must complete that action before this command will be completed. Any frames directed to the specified link station that have been received by the adapter but not processed will be handled according to the state the adapter enters upon receipt of this command. Those link station states would be either disconnecting state or link closed state.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'1A', DLC.CLOSE.STATION
1	CMD_CORRELATE	1	Set by the adapter upon return
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the link station to be closed

If there is no immediate error, the adapter sets the RETCODE field to X'FF' (command in process), sets the CMD\_CORRELATE field in the SRB, and interrupts the PC system by setting ISRP-odd bit 5 if an SRB Free Request interrupt is received by the adapter. When the command is completed later, the PC system will be interrupted with a response in the SSB.

If there is an immediate error, the adapter sets the RETCODE field with the error code and interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

- X'FF' Command in process
- X'01' Invalid command code
- X'02' Duplicate command, one already outstanding
- X'04' Adapter closed, should be open
- X'40' Invalid STATION\_ID
- X'4C' Unable to close, commands outstanding

#### Final RETCODE in SSB

- X'00' Operation completed successfully
- X'4B' Station closed, no remote acknowledgment

## DLC.CONNECT.STATION

Hex 1B

### DLC.CONNECT.STATION

**Command Description:** To initiate a SABME-UA exchange to place both the local and remote link stations in data transfer state, or to complete such an exchange initiated by the remote station.

**Command Specifics:** This command will not be accepted if the link station is in the Disconnecting or Link Closed state, or if a DLC.CLOSE.STATION or DLC.CONNECT.STATION command is in process. Any outstanding transmit commands queued to the link station will be lost.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'1B', DLC.CONNECT.STATION
1	CMD_CORRELATE	1	Set by the adapter upon return
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the link station to be closed
6	ROUTING_INFO	18	See below

### ROUTING\_INFO

**Explanation:** See the ROUTING\_ADDR field of DLC.CONNECT.STATION command on page 3-7.

If there is no immediate error, the adapter sets the RETCODE field to X'FF' (command in process), sets the CMD\_CORRELATE field in the SRB, and interrupts the PC system by setting ISRP-odd bit 5 if an SRB Free Request interrupt is received by the adapter. When the command is completed later, the PC system will be interrupted with a response in the SSB. A successful return code indicates that the local link station has entered the Link Opened state. An unsuccessful return code indicates that it has entered the Disconnected state.

If there is an immediate error, the adapter sets the RETCODE field with the error code and interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

- X'FF' Command in process
- X'01' Invalid command code
- X'02' Duplicate command, one already outstanding
- X'04' Adapter closed, should be open
- X'40' Invalid STATION\_ID
- X'41' Protocol error, link in invalid state for command
- X'44' Invalid routing information
- X'4A' Sequence error, command in process

## DLC.FLOW.CONTROL

### Final RETCODE in SSB

X'00' Operation completed successfully

X'4D' Unsuccessful link station connection attempt

## DLC.FLOW.CONTROL

Hex 1D

### DLC.FLOW.CONTROL

**Command Description:** To control the flow of data across a specified link station on a SAP, or every link station on a SAP.

**Command Specifics:** Local Busy state is set either because of a user request, or because a RECEIVED.DATA command from the adapter to the PC system has been rejected due to a buffer shortage. In the latter case, the condition must be reset by the PC system program when buffers become available, by using this command with option bit 6 set.

This command affects the secondary state of target link stations, causing the Local Busy states to be set or reset. The command will complete successfully even if it makes no change to the existing state. That is, a request to reset Local Busy will be accepted even if the link is not in Local Busy state.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'1D', DLC.FLOW.CONTROL
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the link station or SAP
6	FLOW_OPTIONS	1	Option byte

### STATION\_ID

**Explanation:** If the STATION\_ID is a SAP (X'nn00'), the command will be applied to all link stations included in the SAP. If the STATION\_ID is a link station (X'nsss'), the command will be applied only to the specified link station.

### FLOW\_OPTIONS

**Explanation:** The flow option byte is described below.

- Bit 7: Set/Reset Local Busy State

If this bit is zero, the related link station(s) will enter the Local Busy link secondary state. If the station is in the Link Opened primary state and not already in Local Busy state, a Receiver Not Ready supervisory frame will be transmitted. Then I Frames received for this station will be discarded until this condition is reset by the application program.

If this bit is on, option bit 6 will be checked to determine whether Local Busy (user set) or Local Busy (buffer set) should be reset. If both Local Busy states are reset after this command has been accepted and the primary link state is Link Opened, the link will enter either the checkpointing or clearing secondary state to ensure that the remote station is aware that the condition has been reset.

- Bit 6: User/Buffer Reset

If bit 6 is 0 and option bit 7 is 1, Local Busy (user set) will be reset.

If bit 6 is 1 and option bit 7 is 1, Local Busy (buffer set) will be reset.

If option bit 7 is zero, this bit is ignored.

- Bits 5-0: Reserved

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

**X'00'** Operation completed successfully

**X'01'** Invalid command code

**X'02'** Duplicate command, one already outstanding

**X'04'** Adapter closed, should be open

**X'40'** Invalid STATION\_ID

## DLC.MODIFY

Hex 1C

DLC.MODIFY

**Command Description:** To modify certain working values of an open link station or the default values of a SAP.

**Command Specifics:** The values to be updated are included in SRB.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'1C', DLC.MODIFY
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	Set by the adapter upon return
6	TIMER_T1	1	T1 value, response timer
7	TIMER_T2	1	T2 value, acknowledgment timer
8	TIMER_Ti	1	Ti value, inactivity timer
9	MAXOUT	1	Max transmits without a receive acknowledgment
10	MAXIN	1	Max receives without a transmit acknowledgment
11	MAXOUT_INCR	1	Dynamic window increment value
12	MAX_RETRY_COUNT	1	N2 value
13	ACCESS_PRIORITY	1	New access priority for transmission
14	SAP_GSAP_MEM	1	Number of following group SAPs
15	GSAPS	n	GSAP list, maximum 13

---

**STATION\_ID**

**Explanation:** If this is a SAP STATION\_ID, the command will affect the default values held in the SAP control block, but not the current values of open link stations. If it is a link station STATION\_ID, the command will affect the current values of the designated (open) link station.

---

**TIMER\_T1, TIMER\_Ti**

**Explanation:** These values must be less than 11 for T1 and Ti. If a value greater than 10 is provided for T2, the acknowledgment timer will not run. If the field is zero, the existing value will remain unchanged.

---

**MAXOUT**

**Explanation:** This parameter may not exceed 127. If the field is zero, the existing value will remain unchanged.

---

**MAXIN**

**Explanation:** This parameter may not exceed 127. If the field is zero, the existing value will remain unchanged.

---

**MAXOUT\_INCR**

**Explanation:** This parameter may not exceed 255. If the field is zero, the existing value will remain unchanged.

---

**MAX\_RETRY\_COUNT**

**Explanation:** This parameter may not exceed 255. If the field is zero, the existing value will remain unchanged.

---

**ACCESS\_PRIORITY**

**Explanation:** If the requested access priority exceeds the limit authorized for the adapter it will be rejected. The access priority is contained in the 3 low-order bits of this byte.

---

**SAP\_GSAP\_MEM**

**Explanation:** The number of SAP\_VALUES in the GSAPS field. This field is only checked and used if the SAP was opened as a group member. The maximum value is 13. (The most SAP\_VALUES that the SRB length will accommodate).

---

**GSAPS**

**Explanation:** This field is used for an individual SAP to request membership in additional group SAPs to request that membership be canceled. If the low-order bit of a SAP\_VALUE is zero, it indicates that membership in the the associated group SAP is being requested. If the low-order bit of a SAP\_VALUE is 1, it indicates that membership should be cancelled. The group SAPs must be open when the assignment is requested, and all members of a group SAP must have the same XID handling option selected. If an error is found while processing the list of group SAPs, an error return code will be set and processing will stop. The SAP\_GSAP\_MEM field will be overwritten with the value of the failing group SAP. Other parameter changes will take place as requested.

**SRB Response**

When the adapter completes the modify command, the return code is placed in the RETCODE field. The adapter then interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open
- X'08' Unauthorized access priority
- X'40' Invalid STATION\_ID
- X'42' Parameter exceeded maximum allowed
- X'45' Membership requested in non-existent group SAP
- X'49' Group SAP has reached maximum membership
- X'4E' Member SAP not found in group SAP list

**DLC.OPEN.SAP**

Hex 15

DLC.OPEN.SAP

**Command Description:** Open (activate) a service access point (SAP) and allocate an individual SAP control block.

**Command Specifics:** A group SAP control block and one or more link station control blocks may also be allocated by this command.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'15', DLC.OPEN.SAP
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	Set by the adapter upon return
6	TIMER_T1	1	T1 value, response timer
7	TIMER_T2	1	T2 value, acknowledgment timer
8	TIMER_Ti	1	Ti value, inactivity timer
9	MAXOUT	1	Max transmits without a receive acknowledgment
10	MAXIN	1	Max receives without a transmit acknowledgment
11	MAXOUT_INCR	1	Dynamic window increment value
12	MAX_RETRY_COUNT	1	N2 value
13	GSAP_MAX_MEM	1	Maximum number of SAPs for a group SAP
14	MAX_I_FIELD	2	Maximum received information field length
16	SAP_VALUE	1	SAP value to be assigned
17	SAP_OPTIONS	1	Option byte, see below
18	STATION_COUNT	1	Number of link stations to reserve
19	SAP_GSAP_MEM	1	Number of entries in GSAP list
20	GSAPS	n	GSAP list, maximum = 8

**STATION\_ID**

**Explanation:** The adapter will set this to the STATION ID to be used in future commands referencing this SAP.

**TIMER\_T1, TIMER\_T2, TIMER\_Ti**

**Explanation:** These values must be less than 11 for T1 and Ti. If a value greater than 10 is provided for T2, the acknowledgment timer will not run. If the field is zero, the adapter will provide defaults. The default values are fixed, but their effect depends on the "TICK" values or defaults selected with the DIR.OPEN.ADAPTER command.

**MAXOUT**

**Explanation:** This parameter may not exceed 127. If a zero is provided, the default of 2 will be used.

**MAXIN**

**Explanation:** This parameter may not exceed 127. If a zero is provided, the default of 1 will be used.

**MAXOUT\_INCR**

**Explanation:** This parameter may not exceed 255. If a zero is provided, the default of 1 will be used.

**MAX\_RETRY\_COUNT**

**Explanation:** This parameter may not exceed 255. If a zero is provided, the default of 8 will be used.

**SAP\_VALUE**

**Explanation:** The value that will be used as the source SAP in transmitted frames and recognized as the destination SAP in received frames. The low-order bit of this field will be ignored. A DLC.OPEN.SAP command always allocates an individual SAP control block. A value of X'00' will always be rejected and a value of X'FE' will be rejected if the group SAP option is requested. If option bit 1 is a 1, the SAP\_VALUE with the low-order bit set to 1 will be the group SAP value. In other words, the next higher (odd-numbered) SAP control block will be allocated to be a group SAP.

**MAX\_I\_FIELD**

**Explanation:** This parameter defines the maximum length of a received I frame for a link station. If the STATION\_COUNT parameter is zero, this field is ignored. If this field is zero, the default will be 600 bytes long. The maximum length is 4905 bytes on a 4Mbps ring and 18000 bytes on a 15Mbps ring.

**SAP\_OPTIONS**

**Explanation:**

Bits 7-5	Priority	The transmission priority for this SAP. If the requested priority exceeds the limit authorized for the adapter, the command will be rejected.
Bit 4	Reserved	Should be zero. Not checked.
Bit 3	XID handling option	If this is zero, XID commands are handled by the adapter. If this is 1, XID commands are passed to the application program.
Bit 2	Individual option	If this bit is 1, the SAP will handle frames as an individual SAP.
Bit 1	Group option	If this bit is 1, the SAP will handle frames as a group SAP.

<b>Bit 0</b>	Group member	If this bit is 1, the SAP may be a member of a group SAP.
<b>Note:</b> If bit 0 is a 1, bit 2 must also be a 1.		

---

### STATION\_COUNT

**Explanation:** This parameter specifies the maximum number of link stations that may be open for this SAP at the same time, and applies only if the SAP is an individual SAP. If the number of link stations requested for this SAP, together with those already requested for previously opened SAPs, exceeds the DLC\_MAX\_STATIONS parameter value from the DIR.OPEN.ADAPTER command, the DLC.OPEN.SAP command will be rejected.

---

### GSAP\_MAX\_MEM

**Explanation:** The maximum number of individual SAPs that may be assigned membership in the group SAP if this SAP is designated to be a group SAP. Membership is assigned in the group SAP as the individual SAPs are opened. This parameter may not exceed the similar parameter provided with the DIR.OPEN.ADAPTER command and will default to that value if it is zero.

---

### SAP\_GSAP\_MEM

**Explanation:** The number of SAP\_VALUES in the GSAPS field. The maximum value is 8.

---

### GSAPS

**Explanation:** This field is used for an individual SAP to request membership in group SAPs. The SAP\_GSAP\_MEM parameter indicates the number of valid values in this field. If additional membership is required, the DLC.MODIFY command may be used for the requests. The group SAPs must be open when the assignment is requested, and all members of a group SAP must have the same XID handling option selected. If an error is found while processing the list of group SAPs, an error return code will be set and processing will stop. The SAP\_GSAP\_MEM field will be overwritten with the value of the failing group SAP. This will not affect the status of the SAP.

### SRB Response

When the adapter completes the open command, the return code is placed in the RETCODE field. The adapter then interrupts the PC system by setting ISRP-odd bit 5.

#### Valid Return Codes:

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open
- X'06' Option(s) missing, invalid, or incompatible
- X'08' Unauthorized access priority
- X'42' Parameter exceeded maximum allowed
- X'43' Invalid SAP\_VALUE or value already in use
- X'45' Membership requested in non-existent group SAP
- X'46' Requested resources not available
- X'49' Group SAP has reached maximum membership



## DLC.OPEN.STATION

Hex 19

### DLC.OPEN.STATION

**Command Description:** Allocate resources to support a logical link connection.

**Command Specifics:** These resources may also be allocated when a SABME is received against an open SAP and the appropriate station is not already open.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'19', DLC.OPEN.STATION
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	SAP ID (X'nn00')
6	TIMER_T1	1	T1 value, response timer
7	TIMER_T2	1	T2 value, acknowledgment timer
8	TIMER_Ti	1	Ti value, inactivity timer
9	MAXOUT	1	Max transmits without a receive acknowledgment
10	MAXIN	1	Max receives without a transmit acknowledgment
11	MAXOUT_INCR	1	Dynamic window increment value
12	MAX_RETRY_COUNT	1	N2 value
13	RSAP_VALUE	1	The remote SAP value
14	MAX_I_FIELD	2	Maximum received information field length
16	STATION_OPTIONS	1	Option byte, see below
17		1	Reserved
18	REMOTE_ADDRESS	6	Ring address of the remote station

### STATION\_ID

**Explanation:** The application program must specify the SAP STATION\_ID (X'nn00') and the adapter will set the link ID (ss portion of X'nsss') to be used in future commands referencing this station.

### TIMER\_T1 through MAX\_RETRY\_COUNT

**Explanation:** See the same parameters for the DLC.OPEN.SAP command on page 7-48.

### RSAP\_VALUE

**Explanation:** The value that will be used as the destination SAP in transmitted frames and recognized as the source SAP in received frames. The low-order bit of this field must be zero, indicating an individual SAP. A value of X'00' (the null SAP) will be rejected.

---

**MAX\_I\_FIELD**

**Explanation:** This parameter defines the maximum length of a received I frame. If this field is zero, the value from the SAP control block will be used.

---

**STATION\_OPTIONS**

**Explanation:**

<b>Bits 7-5</b>	Priority	The transmission priority for this link station. If the requested priority exceeds the limit authorized for the adapter, the command will be rejected. If a zero is provided, an access priority of zero is used.
<b>Bits 4-0</b>	Reserved	Should be zero. Not checked.

---

**REMOTE\_ADDRESS**

**Explanation:** The 6-byte NODE\_ADDRESS of the remote station. The high-order bit of the high-order byte of this field must be zero, indicating a specific address.

**SRB Response**

When the adapter completes the open command, the return code is placed in the RETCODE field. The adapter then interrupts the PC system by setting ISRP-odd bit 5.

This command should be followed by a DLC.CONNECT.STATION command, which should include the routing information if the remote station is on a different ring.

**Valid Return Codes:**

- X'01'** Invalid command code
- X'04'** Adapter closed, should be open
- X'05'** Required parameter(s) not provided
- X'08'** Unauthorized access priority
- X'40'** Invalid STATION\_ID
- X'42'** Parameter exceeded maximum allowed
- X'43'** Invalid SAP\_VALUE or value already in use
- X'46'** Requested resources not available
- X'4F'** Invalid remote address

## DLC.ALLOCATE

## DLC.REALLOCATE

Hex 17

### DLC.REALLOCATE

**Command Description:** This command removes a given number of link station control blocks from a SAP and returns them to the adapter pool, or removes a given number of link station control blocks from the adapter pool and adds them to a SAP.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'17', DLC.REALLOCATE
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	Link station ID affected
6	OPTION_BYTE	1	Add/subtract option
7	STATION_COUNT	1	Number of link station control blocks to move
8	ADAPTER_COUNT	1	Number of link station control blocks for the adapter. Set by adapter on return.
9	SAP.COUNT	1	Number of link station control blocks for the SAP. Set by adapter on return.

### OPTION\_BYTE

**Explanation:**

- If bit 0 is 0, then take link station control blocks from the adapter and add to the SAP.
- If bit 0 is 1, then take link station control blocks from the SAP and add to the adapter.
- Bits 1 through 7 are reserved

### STATION\_COUNT

**Explanation:** The number of link station control blocks to be moved as indicated by the option byte. If more link station control blocks are requested than are available on the adapter or SAP, all those available will be moved.

### ADAPTER\_COUNT

**Explanation:** The number of link station control blocks available for the adapter (not allocated to a SAP), after the command has completed. This field is only valid if the return code is X'00' or X'40'.

### SAP\_COUNT

**Explanation:** The number of link station control blocks available for the SAP specified in the station ID field (Not in use for an open station) after the command has completed. This field is only valid if the return code is X'00'.

### SRB Response

When the adapter completes the command, it sets the return code and the PC system is interrupted with ISRP bit 2 set.

**Valid Return Codes:**

- X'00'** Operation completed successfully
- X'01'** Invalid command code
- X'04'** Adapter closed, should be open
- X'40'** Invalid STATION\_ID

## DLC.RESET

Hex 14

### DLC.RESET

**Command Description:** Reset either one SAP and all associated link stations, or all SAPs and all associated link stations.

**Command Specifics:** After the command is completed the affected SAPs and link stations will be closed. No commands or communication directed to them will be accepted. The reset command will not complete until all related resources can be freed. This means that transmissions already queued to the ring hardware and commands from the adapter to the PC system must be complete before this command will be completed. Frames received for the affected SAPs and link stations but not yet passed to the PC system will be discarded by the adapter. The same is true for frames received while the reset is in progress. Requests queued to SAPs and link stations that have not started completion will not be completed. A beaconing ring can cause this command to hang if transmits are queued to the hardware. The command will complete when beaconing clears.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'14', DLC.RESET
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the SAP(s)/station(s) to be reset <b>X'0000'</b> all SAPs and all stations <b>X'nn00'</b> SAP nn and all its stations

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

**Valid Return Codes:**

- X'00'** Operation completed successfully
- X'01'** Invalid command code
- X'04'** Adapter closed, should be open
- X'40'** Invalid STATION\_ID

## DLC.STATISTICS

Hex 1E

## DLC.STATISTICS

**Command Description:** Read statistics for a specific link station.

**Command Specifics:** The error counters (first five station statistics) may be reset if requested. If a counter overflows (high-order bit of the field changes from zero to 1), a DLC status adapter request block (ARB) will be presented to the PC system, indicating that this command should be issued.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'1E', DLC.STATISTICS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	The link station to obtain statistics from
6	COUNTERS_ADDR	2	Offset to the address of the statistics *
8	HEADER_ADDR	2	Offset to the address of the LAN header *
10	HEADER_LENGTH	1	Length of the LAN header *
10	RESET_OPTION	1	Option byte, see below
* Indicates a value set by the adapter.			

## COUNTERS\_ADDR

**Explanation:** An address within the SRB where a copy of the counter contents is located. The PC system application program should move this information into PC system memory before reusing the SRB.

## HEADER\_ADDR

**Explanation:** The offset within shared RAM of the LAN header consisting of the access control (AC) field, the frame control (FC) field, the destination address, the source address, and the routing information. If no routing information is present, the header length will be 14 bytes. The source address field will not be set until the first frame is transmitted for the link station, except that the high-order bit of the high-order byte is set on if routing information is present.

## RESET\_OPTION

**Explanation:**

Bit 7	If this bit is zero, the adapter will not alter the contents of the error counters. If this bit is 1, the adapter will reset the contents of the error counters.
Bits 6-0	Reserved

### The Link Station Statistics

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	I_FRAME_XMIT_COUNT	2	The number of I frames transmitted
2	I_FRAME_RCV_COUNT	2	The number of I frames received
4	I_FRAME_XMIT_ERR	1	The number of I frame transmit errors
5	I_FRAME_RCV_ERR	1	The number of I frame receive errors
6	T1_EXPIRED	2	The number of times T1 expired
8	STATION_RCVD_CMD	1	The last command or response received
9	STATION_SENT_CMD	1	The last command or response sent
10	STATION_PRMY_STATE	1	The link primary state, see below
11	STATION_SCDY_STATE	1	The link secondary state, see below
12	STATION_VS	1	The send state variable
13	STATION_VR	1	The receive state variable
14	STATION_VA	1	The last received NR
* All values are set by the adapter.			

#### STATION\_PRMY\_STATE

**Explanation:** This field indicates the link station's primary state as maintained in the control block at the time the DLC.STATISTICS command is completed. It consists of eight mutually exclusive bit flags, as follows:

<b>Bit 7</b>	Link Closed
<b>Bit 6</b>	Disconnected
<b>Bit 5</b>	Disconnecting
<b>Bit 4</b>	Link Opening
<b>Bit 3</b>	Resetting
<b>Bit 2</b>	FRMR Sent
<b>Bit 1</b>	FRMR Received
<b>Bit 0</b>	Link Opened

#### STATION\_SCDY\_STATE

**Explanation:** This field indicates the link station's secondary state as maintained in the control block at the time the DLC.STATISTICS command is completed. It consists of seven non-exclusive bit flags, as follows:

<b>Bit 7</b>	Checkpointing
<b>Bit 6</b>	Local Busy (user set)
<b>Bit 5</b>	Local Busy (buffer set)
<b>Bit 4</b>	Remote Busy
<b>Bit 3</b>	Rejection
<b>Bit 2</b>	Clearing
<b>Bit 1</b>	Dynamic Window Algorithm Running
<b>Bit 0</b>	Reserved (may appear as 0 or 1)

When the adapter completes the operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

## DLC.STATISTICS

### Valid Return Codes:

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open
- X'40' Invalid STATION\_ID

## Transmit Commands

There is only one basic transmit command with several subcommands indicating the type of data to be transmitted. All the commands have the same format, the only difference being the actual command code.

When a transmit command is presented to the adapter, it indicates a request to send data. The actual data is not given to the adapter until the adapter issues the TRANSMIT.DATA.REQUEST command to the PC system.

The processing sequence for transmit commands is:

1. The PC system application program issues a Transmit command to the adapter.
2. The adapter sets a command correlator and in-process return code in the SRB.
3. The adapter issues a TRANSMIT.DATA.REQUEST command (X'82') to the PC system using the Adapter Request Block (ARB). This command supplies the command correlator, the STATION\_ID, and the DHB address in shared RAM where the PC system should start to transfer the data.
4. The PC system moves the data into the DHB.
5. The PC system responds using the adapter status block (ASB) providing the original transmit command used in the SRB, the command correlator, the STATION\_ID, and the transmit data length information.
6. The adapter transmits the frame.
7. The adapter sets completion information in the system status block (SSB) on completion of the transmission for the direct and SAP interfaces, or on receipt of acknowledgment, or determination that acknowledgment will not be received for the link station interface. The adapter then interrupts the PC system.

The transmit commands are:

COMMAND NAME	CODE
TRANSMIT.DIR.frame	X'0A'
TRANSMIT.I.frame	X'0B'
TRANSMIT.UI.frame	X'0D'
TRANSMIT.XID.CMD	X'0E'
TRANSMIT.XID.RESP.FINAL	X'0F'
TRANSMIT.XID.RESP.NOT.FINAL	X'10'
TRANSMIT.TEST.CMD	X'11'

## Transmit Summary

- Direct Station

May use only the TRANSMIT.DIR.frame command. No retry is provided.

- SAP Station

May use all commands except the TRANSMIT.DIR.frame and TRANSMIT.I.frame commands. No retry is provided.

The TRANSMIT.XID.RESP commands should only be issued to a SAP that has the XID handling option selected to pass XID frames to the application program.

- Link Station

May use only the TRANSMIT.I.frame command. All transmission retry is handled by the adapter.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'xx', TRANSMIT.xxx
1	CMD.CORRELATE	1	Set by the adapter upon return
2	RETCODE	1	Set by the adapter upon return
3		1	Reserved
4	STATION_ID	2	ID of the station sending the data

If there is no immediate error, the adapter sets the RETCODE field to X'FF' and sets the command correlator field. The adapter will interrupt the PC system by setting ISRP-odd bit 5 if an SRB Free Request interrupt is received by the adapter.

If there is an immediate error, the adapter sets the RETCODE field with the appropriate code and interrupts the PC system by setting ISRP-odd bit 5.

### Valid Return Codes:

- X'FF' Command in process
- X'01' Invalid command code
- X'04' Adapter closed, should be open
- X'25' Maximum commands exceeded
- X'40' Invalid STATION\_ID
- X'41' Protocol error, link in invalid state for command
- X'4A' Sequence error, command in process

When the adapter completes the transmit command it prepares the system status block (SSB) and interrupts the PC system by setting ISRP-odd bit 2. If more than one TRANSMIT.I.frame command is being reported, the command correlate field will contain the correlator for the last command completed.



## ARB Commands

The SSB will contain:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'xx', The transmit command from the SRB
1	CMD.CORRELATE	1	PC/Adapter command correlator
2	RETCODE	1	Completion code
3		1	Reserved
4	STATION_ID	2	ID of the station providing status
6	TRANSMIT_ERROR	1	If RETCODE=X'22' the returned FS byte

### SSB Return Codes

#### Valid Return Codes:

- X'00' Operation completed successfully
- X'08' Unauthorized access priority
- X'22' Error on frame transmission, check TRANSMIT\_FS data
- X'23' Error in frame transmit or strip process
- X'24' Unauthorized MAC frame
- X'27' Link not transmitting l frames, state changed from link opened
- X'28' Invalid transmit frame length command

---

## Adapter-to-PC system Commands

The commands listed below, in alphabetical order, may be issued to the PC system by the adapter.

The commands have the following in common:

- The adapter prepares the command in the ARB and interrupts the PC system by setting ISRP-odd bit 3.
- The PC system reads the command information and interrupts the adapter by setting ISRA-odd bit 1 to acknowledge receipt of the command and indicate that the adapter may reuse the ARB.
- If a response is required, the PC system will put the response information in the ASB and interrupt the adapter by setting ISRA-odd bit 4.
- After reading the ASB response the adapter will either:
  - Set the return code to X'FF' and interrupt the PC system by setting ISRP-odd bit 4 if the ASB Free Request interrupt bit is set, or
  - Set a return code indicating that an error has been detected and interrupt the PC system by setting ISRP-odd bit 4 regardless of the state of the ASB Free Request interrupt bit.

The Adapter to PC system Commands are:

COMMAND NAME	CODE
DLC.STATUS	X'83'
RECEIVED.DATA	X'81'
RING.STATUS.CHANGE	X'84'
TRANSMIT.DATA.REQUEST	X'82'

## DLC.STATUS

Hex 83

### DLC.STATUS

**Command Description:** The adapter is indicating a change in DLC status to the PC system.

**Command Specifics:** When the PC system has read the command information from the ARB, it will interrupt the adapter by setting ISRA bit 1 to acknowledge receipt of the command and indicate that the adapter may reuse the ARB. No response is required for this command. However, refer to "Suggested Actions in Response to DLC Status" on page B-26.

The ARB content is:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'83', DLC.STATUS
1		3	Reserved
4	STATION_ID	2	ID of the SAP(s)/station(s) presenting status
6	STATUS	2	DLC status indicator, see below
8	FRMR_DATA	5	Data sent/received with FRMR response
13	ACCESS_PRIORITY	1	New access priority for SAP/Station
14	REMOTE_ADDRESS	6	The physical ring address of the remote station
20	RSAP_VALUE	1	Remote station's SAP_VALUE

## STATUS

**Explanation:** More than one bit may be set in the status word if the adapter had to wait for the ARB to become available. The bit meanings are listed below. For a list of responses to these conditions, see "Suggested Actions in Response to DLC Status" on page B-26.

- Bit 15** Link lost
- Bit 14** DM or DISC received, or DISC acknowledged
- Bit 13** FRMR received
- Bit 12** FRMR sent
- Bit 11** SABME received for an open link station
- Bit 10** SABME received, link station opened
- Bit 9** Remote station has entered Local Busy state
- Bit 8** Remote station has left Local Busy state
- Bit 7** Ti Timer has expired
- Bit 6** DLC counter overflow
- Bit 5** Access priority reduced

## RECEIVED.DATA

Bit 4-0 Reserved

## RECEIVED.DATA

Hex 81

### RECEIVED.DATA

**Command Description:** This command informs the PC system that data for a particular STATION\_ID has been received. The data must be moved from the receive buffers in shared RAM and placed into buffers in PC system memory.

**Command Specifics:** When the PC system has completed processing the RECEIVED.DATA command, it will provide a return code in the ASB and interrupt the adapter by setting ISRA-odd bit 4. If the return code is X'20', and the frame was an I frame destined for a link station, the adapter will set Local Busy state (buffer set) for the affected link station. It is the PC system application program's responsibility to determine whether or not receipt of a partial I frame is acceptable, and also to reset the local-busy state when buffers become available.

The ARB content is:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'81', RECEIVED.DATA
1		3	Reserved
4	STATION_ID	2	ID of the receiving station, see below
6	RECEIVE_BUFFER	2	Offset to the first receive buffer in shared RAM
8	LAN_HDR_LENGTH	1	The length of the LAN header field
9	DLC_HDR_LENGTH	1	The length of the LLC header field
10	FRAME_LENGTH	2	Length of the entire frame
12	NCB_TYPE	1	Category of the message received, see below

### STATION\_ID

**Explanation:** This field will indicate the link station, the SAP, or (if X'0000') the direct station that the data is destined for.

### DLC\_HDR\_LENGTH

**Explanation:** This is the actual LLC header length if the message is a non-MAC frame and the destination is either a SAP or a link station.

It is equal to X'00' if the message is either a MAC frame or a non-MAC frame and the destination is the direct station.

### NCB\_TYPE

**Explanation:** Below are the different categories of messages received.

*Hex*

Value	Type
02	MAC frame
04	I frame

06	UI frame
08	XID command poll
0A	XID command not-poll
0C	XID response final
0E	XID response not-final
10	TEST response final
12	TEST response not-final
14	Other or unidentified

### The ASB Response from the PC system

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'81', RECEIVED.DATA
1		1	Reserved
2	RETCODE	1	Return (completion) provided by the PC system program
3		1	Reserved
4	STATION_ID	2	ID of the station receiving data
6	RECEIVE_BUFFER	2	Offset to the address of the first receive buffer in shared RAM

### Return Code to the Adapter

- X'00' Operation completed successfully
- X'20' Lost data on receive, no buffers available

### Return Code to the PC system

- X'FF' Response valid, ASB available
- X'01' Unrecognized command code
- X'26' Unrecognized command correlator – see below
- X'40' Invalid STATION\_ID

**Note:** For this response only, "Unrecognized command correlator" means that the receive buffer address is not that which is expected by the adapter.

### The Received Data

Received data is held in the adapter shared RAM in one or more receive buffers, depending on the length of the frame. The address of the first, or only, receive buffer will be provided to the PC system in the ARB with the RECEIVED.DATA command. In the last, or only buffer containing the frame, bytes 2 and 3 will contain X'0000', otherwise they will contain the address of the next buffer plus 2 bytes.

### The Receive Buffer Format

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0		2	Reserved
2	BUFFER_POINTER	2	Offset to the address of the next buffer plus 2, or zero if this is the last buffer
4		1	Reserved
5	RECEIVE_FS	1	FS/Address match (last buffer only)
6	BUFFER_LENGTH	2	Length of the data in this buffer
8	FRAME_DATA	n	Frame data

## TRANSMIT.DATA.REQUEST

---

### RECEIVE\_FS

#### Explanation:

**Bit 7** Address recognized indicator  
**Bit 6** Frame copied indicator  
**Bit 5** Reserved  
**Bit 4** Reserved  
**Bit 3** Address recognized indicator  
**Bit 2** Frame copied indicator  
**Bit 1-0** Reserved

## RING.STATUS.CHANGE

Hex 84

### RING.STATUS.CHANGE

**Command Specifics:** The adapter is indicating a change in the network status to the PC system.

The status provided with this command is the current network status and may possibly equal the last status if the adapter has had to wait for the ARB to become available.

When the PC system has read the command information from the ARB, it will interrupt the adapter by setting ISRA-odd bit 1 to acknowledge receipt of the command and indicate that the adapter may reuse the ARB. No response is required for this command.

The ARB content is:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'84', RING.STATUS.CHANGE
1		5	Reserved
6	NETW_STATUS	2	Current network status, See "Network Status" on page B-44

## TRANSMIT.DATA.REQUEST

Hex 82

### TRANSMIT.DATA.REQUEST

**Command Specifics:** This command informs the PC system that data for a transmit command previously issued by the PC system is needed.

When the PC system has read the command information from the ARB, it will interrupt the adapter by setting ISRA-odd bit 1 to acknowledge receipt of the command and indicate that the adapter may reuse the ARB.

When the PC system has completed processing the TRANSMIT.DATA.REQUEST command, it will provide a return code in the ASB and interrupt the adapter by setting ISRA-odd bit 4. Only a successful return code is expected by the adapter in response to this request. The PC system program should make sure that the transmit request is valid before issuing the original command to the adapter.

The ARB content is:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'82', TRANSMIT.DATA.REQUEST
1	CMD_CORRELATE	1	PC/Adapter command correlator
2		2	Reserved
4	STATION_ID	2	ID of the sending station
6	DHB_ADDRESS	2	The address of the DHB to put the data in

### The DHB Contents after the Data Move

#### 1. TRANSMIT.I.frame

This is the data field of the frame to be transmitted. The adapter provides the LAN and DLC headers.

#### 2. TRANSMIT.DIR.frame

This is the entire message, including the LAN header and any additional headers with space reserved for the LAN source address to be inserted by the adapter. If the LAN header contains routing information, the PC system must set the high-order bit of the high-order byte of the source address field on.

#### 3. All other commands

These include the LAN header with space reserved for the LAN source address to be inserted by the adapter, followed by 3 bytes reserved for the adapter to insert the LLC header, followed by the data. The adapter will determine whether or not the LAN header includes routing information by checking the length field in the ASB accompanying the DHB.

### The ASB Response from the PC system

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	The transmit command as provided in the original SRB command
1	CMD_CORRELATE	1	PC/Adapter command correlator
2	RETCODE	1	Return (completion) provided by the PC system program
3		1	Reserved
4	STATION_ID	2	ID of the station sending data
6	FRAME_LENGTH	2	Length of the entire frame
8	HEADER_LENGTH	1	Length of the LAN header, required only for a SAP
9	RSAP_VALUE	1	Remote SAP, the DSAP in the transmitted frame, required only for SAP station IDs

### Return Code to the Adapter

**X'00'** Operation completed successfully

### Return Code to the PC system

**X'FF'** Response valid, ASB available

**X'01'** Unrecognized command code

**X'26'** Unrecognized command correlator

**X'40'** Invalid STATION\_ID

---

### Additional Adapter Functions

Since the original Token-Ring Network PC Adapter was introduced, many new Token-Ring Network adapters have been developed. This section discusses some of these additional functions provided by these newer adapters. The additional functions not discussed here are discussed earlier in this chapter.

- Additional shared RAM
- Bridge functions
- Shared RAM paging
- Adapter data rate selection
- Early Token Release (ETR)

The additional functions are described in the following sections.

#### Adapter's Shared RAM

- The original Token-Ring Network PC Adapter has 8K of shared RAM.
- The Token-Ring Network PC Adapter II has 16K of shared RAM.
- The Token-Ring Network Adapter/A has 16K of shared RAM.
- The Token-Ring Network 16/4 Adapter/A has 64K of shared RAM.
- The Token-Ring Network 16/4 Adapter has 64K of shared RAM.

#### The Bridge Functions

By using two Token-Ring Network adapters (neither being the original Token-Ring Network PC Adapter) in the same PC system, each connected to a separate ring, a bridge application program can forward frames from either ring to the other. This is called a bridge between the two rings.

The bridge function is provided by:

- Two additional SRB commands: X'09' and X'0C'
- An additional ARB command: X'85'
- Two additional areas in shared RAM
  - Bridge transmit control area
  - Bridge transmit buffers
- Two additional interrupt status register bits, ISRA-odd bit 6 and ISRP-odd bit 1.

A DIR.CONFIG.BRIDGE.RAM command must be issued prior to the DIR.OPEN.ADAPTER command. This ensures that the shared RAM will be prepared with the bridge transmit areas allocated when the open is performed.

After the adapter has been opened, a DIR.SET.BRIDGE.PARM command must be issued to enable frames to be received for forwarding.

An adapter that is opened for bridge functions interrogates all frames passing on the ring. Any received frame that does not have any other address match for the adapter and has a routing information (RI) field is to be forwarded. See the *IBM Token-Ring Network Architecture Reference* for more about routing frames.

When the adapter receives a frame from the ring for forwarding, the adapter issues an ARB command (the RECEIVE.BRIDGE.DATA command) to the PC system.

The application program must move the frame data from the receive buffers of the receiving adapter in shared RAM to the transmit buffers in the shared RAM of the adapter connected to the other ring. Then the

application program must inform the receiving adapter that the frame has been accepted by responding to the ARB with an ASB.

The application program must set ISRA-odd bit 6 to initiate transmitting the frame now in the bridge transmit buffer in shared RAM of the transmit adapter. When the adapter has completed transmitting the frame, it sets ISRP-odd bit 1 to inform the application program.

The bridge transmit control area is used during the transmission to monitor buffer use and availability.

**Shared RAM Layout for Bridge Use**

The Adapter II or Adapter/A will assign locations in shared RAM when the adapter is opened for bridge use in a format such as this:

**Beginning of shared RAM (as seen by the adapter)**

PC read-only address space	Adapter private variables and work areas Length: 1496 bytes
	System status block (SSB) Length: 20 bytes
	Adapter request block (ARB) Length: 28 bytes
	Receive buffers Length: space remaining after all SAPs/Stations are defined
	SAP and link station control blocks Length: as defined by maximum number of SAPs/Stations
PC read/write address space	Data holding buffer (DHB) Length: as specified at open adapter time. There may be one or more DHBs.
	System request block (SRB) Length: 28 bytes
	Adapter status block (ASB) Length: 12 bytes
	Bridge transmit control area Length: 16 bytes
	Bridge transmit buffers Length: Defined by the DIR.CONFIG.BRIDGE.RAM command

**End of shared RAM (as seen by the adapter)**

**Note:** On 64K adapters, the 512 bytes at the end is reserved.

The bridge transmit control area and the bridge transmit buffers are the additional fields defined in shared RAM for bridge functions, if the bridge functions are activated by a DIR.CONFIG.BRIDGE.RAM command.

**The Bridge Commands**

The following two commands are provided to allow the use of the bridge functions of the Token-Ring Network adapters (excluding the Token-Ring Network PC Adapter).

COMMAND NAME	CODE
DIR.CONFIG.BRIDGE.RAM	X'0C'
DIR.SET.BRIDGE.PARMS	X'09'



**DIR.CONFIG.BRIDGE.RAM**

Hex 0C

**DIR.CONFIG.BRIDGE.RAM**

**Command Description:** This command tells the adapter how much shared RAM to allocate for bridge transmit control area and buffers.

**Command Specifics:** The adapter must have been initialized and must not be open for this command to be accepted. When subsequent commands are issued, the conditions enabled by this command are incorporated.

In addition to the allocation of shared RAM, this command:

- Forces bit 8 to be set by a subsequent DIR.OPEN.ADAPTER or DIR.SET.FUNCTIONAL.ADDRESS command independent of the value provided by the command.

Once set, this functional address can be reset only by either closing and reinitializing the adapter, or issuing a DIR.CONFIG.BRIDGE.RAM command with a shared RAM size of zero.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'0C', DIR.CONFIG.BRIDGE.RAM
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		5	Reserved
8	XMIT_RAM_SIZE	2	The amount of shared RAM for bridge transmit space

**XMIT\_RAM\_SIZE**

**Explanation:** The number of 8-byte blocks of shared RAM to dedicate for bridge transmit buffers and the associated bridge transmit control area. The transmit buffers will be formatted identically to the receive buffers when the adapter is opened. The minimum value for this field is 3 (24 bytes).

**SRB Response**

When the adapter completes the command, it sets return values in SRB bytes 8 through 11 and the return code is placed in the RETCODE field. The adapter then interrupts the PC system by setting ISRP-odd bit 5. The SRB content will then be as follows.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'0C', DIR.CONFIG.BRIDGE.RAM
1		1	Reserved
2	RETCODE	1	Return code, see below
3		5	Reserved
8	BRIDGE_XMIT	2	Offset to the address of bridge transmit control area
10	SRB_ADDRESS	2	Offset to the address of the SRB

**Valid Return Codes:**

X'00' Operation completed successfully

X'01' Invalid command code

X'03' Adapter open, should be closed

**DIR.SET.BRIDGE.PARMS**

Hex 09

**DIR.SET.BRIDGE.PARMS**

**Command Description:** This command provides values and conditions for the adapter to use when copying frames for forwarding.

**Command Specifics:** A DIR.CONFIG.BRIDGE.RAM command must have previously been completed successfully and the adapter must be open for this command to be accepted. A return code of X'05' (Required parameter[s] not provided) is returned if the DIR.CONFIG.BRIDGE.RAM command was not previously completed successfully. The adapter does not check for parameters missing from this command.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'09', DIR.SET.BRIDGE.PARMS
1		1	Reserved
2	RETCODE	1	Set by the adapter upon return
3		3	Reserved
6	SOURCE_RING	2	Source ring number
8	TARGET_RING	2	Target ring number
10	BRIDGE_NUMBER	2	Individual bridge number
12	PARTITION_BITS	1	Number of partition bits
13	SROUTE_BROADCAST	1	Single-route broadcast path indicator
14	TOKEN_PRIORITY	1	Access priority for forwarding frames

**SOURCE\_RING**

**Explanation:** The adapter compares the value in this field with the routing information source ring field in frames received from the ring when determining if the frame is to be forwarded. This value must be the number of the ring to which this adapter is connected. For instance, the valid range of values is X'X001' to X'XFFF' if the PARTITION\_BITS parameter value is 4. The SOURCE\_RING value must be different from the TARGET\_RING value.

**Note:** All bridges connected to a specific ring must refer to the ring with the same ring number value.

**TARGET\_RING**

**Explanation:** The adapter compares the value in this field with the routing information target ring field in frames received from the ring when determining if the frame is to be forwarded. This value must be the number of the ring to which the other adapter in this PC system is connected. For instance, the valid range of values is X'X001' to X'XFFF' if the PARTITION\_BITS parameter value is 4. The TARGET\_RING value must be different from the SOURCE\_RING value. See the note above.

## DIR.SET.BRIDGE.PARMS

---

### BRIDGE\_NUMBER

**Explanation:** The adapter compares the value in this field with the routing information bridge number field in frames received from the ring when determining if the frame is to be forwarded.

---

### PARTITION\_BITS

**Explanation:** The value in this field is used to determine what portion of each 2-byte segment in the routing information field contains the bridge number. A value of 4 indicates that the low-order 4 bits of the segment is the bridge number. The remaining 12 bits contain the ring number. There is no default value for this field. The application program is responsible for maintaining a validity check on the value used. All bridges in the network must use the same value for this field or its equivalent. See "Routing Control Field" in the *IBM Token-Ring Network Architecture Reference*.

---

### SROUTE\_BROADCAST

**Explanation:** The value in this field is used to determine the handling of single-route broadcast frames that are received. If the value is zero, single-route broadcast frames will not be copied by the adapter. If the value is not zero, single-route broadcast frames will be copied by the adapter.

---

### TOKEN\_PRIORITY

**Explanation:** This value indicates the priority token that can be captured or requested for bridge forward frame use. The maximum value allowed is 4. A value greater than 4 will cause a return code of X'08' (Unauthorized access priority).

This parameter does not affect the priority of frames sent by the application program using the standard transmit buffer path. Refer to the transmit command section from page 3-88 to page 3-94.

### SRB Response

When the adapter completes the DIR.SET.BRIDGE.PARMS operation, it sets the return code in the SRB and interrupts the PC system by setting ISRP-odd bit 5.

### Valid Return Codes:

- X'00' Operation completed successfully
- X'01' Invalid command code
- X'04' Adapter closed, should be open
- X'05' Required parameter(s) not provided
- X'08' Unauthorized access priority

### The Additional Adapter-to-PC System Command

An additional ARB command is provided for notification of frames received for bridge forwarding.

## RECEIVED.BRIDGE.DATA

Hex 85

### RECEIVED.BRIDGE.DATA

**Command Description:** This command informs the PC system that the adapter has received a frame that does not have any other address match for the adapter (such as specific, group, or functional address match) and has an RI field. This command is only valid after a DIR.SET.BRIDGE.PARMS command has been received by the adapter.

**Command Specifics:** All frames received by the adapter as bridge frames will be given to the PC system via the direct interface (station X'0000'). Therefore, until the PC system issues an ASB response to the RECEIVE.BRIDGE.DATA ARB command, no other ARB interrupts for data received on the direct interface can be issued to the PC system. This includes both MAC and additional frames to be forwarded.

The application program must set ISRA-odd bit 1 to indicate that the command has been read from the ARB. After the application program has completed processing the command and written the ASB, ISRA-odd bit 4 must be set to indicate completion to the adapter.

Frames that have a destination address match are passed to the PC system via the normal RECEIVE.DATA ARB (X'81').

The ARB contains:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'85', RECEIVE.BRIDGE.DATA
1		3	Reserved
4	STATION_ID	2	ID of the receiving station, always X'0000'
6	RECEIVE_BUFFER	2	Offset to the address of the first receive buffer in shared RAM
8	LAN_HDR_LENGTH	1	The length of the LAN header field
9		1	Reserved
10	FRAME_LENGTH	2	Length of the entire frame (including CRC)
12	NCB_TYPE	1	Category of the message received, always X'14' (other)

**Note:** The last 4 bytes of data in the receive buffer for a frame received via a RECEIVE.BRIDGE.DATA (X'85') ARB are the received CRC.

### The ASB Response from the PC system

The application program should respond to the RECEIVE.BRIDGE.DATA (X'85') ARB with a RECEIVED.DATA (X'81') ASB as shown here.

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	COMMAND	1	X'81', RECEIVED.DATA
1		1	Reserved
2	RETCODE	1	Return (completion) provided by the PC system program
3		1	Reserved
4	STATION_ID	2	ID of the station receiving data, always X'0000' (direct interface)

## RECEIVED.BRIDGE.DATA

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
6	RECEIVE_BUFFER	2	Offset to the address of the first receive buffer in shared RAM

### Return Code to the Adapter

**X'00'** Operation completed successfully

### Return Code to the PC System

**X'FF'** Response valid, ASB available

**X'01'** Unrecognized command code

**X'26'** Unrecognized command correlator

**X'40'** Invalid STATION\_ID

**Note:** For this response only, "Unrecognized command correlator" means that the receive buffer address is not that which is expected by the adapter.

### The Received Data

See the RECEIVED.DATA ARB command description for details about the receive buffers.

### Transmitting Bridge Forwarded Frames

Transmitting frames forwarded by a bridge application program using the Adapter II is performed differently than normal transmitting. The frame data can be moved directly from the receive buffers in the shared RAM of the receiving adapter to the transmit buffers in the shared RAM of the transmit adapter.

The bridge transmit control area allows the adapter and the application program to jointly manage the transmit buffer pool. The locations of the bridge transmit control area and the bridge transmit buffers in shared RAM are available at the completion of the DIR.CONFIG.BRIDGE.RAM SRB command.

### The Bridge Transmit Control Area

The format of the 16-byte bridge transmit control area is:

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0		2	Reserved
2	INPUT_COUNT	1	Count of the buffers in use by the PC system
3	OUTPUT_COUNT	1	Count of the buffers transmitted by the adapter
4	RETURN_COUNT	1	Count of the buffers returned to the PC system by the adapter after transmission
5		1	Reserved for PC system use
6	MAX_BUFFERS	2	The total number of bridge transmit buffers
8	NEXT_BUFFER	2	The address of the next available buffer
10	OLD_BUFFER	2	The address of the buffer containing the next data to transmit
12		4	Reserved for adapter work area

---

**INPUT\_COUNT**

**Explanation:** This field is incremented by the application program to indicate the number of bridge transmit buffers filled. The adapter may only read this field.

---

**OUTPUT\_COUNT**

**Explanation:** This field is incremented by the adapter to indicate the number of buffers that have been transmitted (successfully or unsuccessfully). The PC system may only read this field.

---

**RETURN\_COUNT**

**Explanation:** This field is set by the application program to indicate the number of buffers that have been returned after transmission by the adapter. The adapter does not use this field.

---

**MAX\_BUFFERS**

**Explanation:** This field contains the total number of bridge transmit buffers formatted when the adapter is opened with the bridge functions active. The PC system may only read this field.

---

**NEXT\_BUFFER**

**Explanation:** When the adapter is opened it sets this field with the address in shared RAM of the first bridge transmit buffer. Thereafter, the adapter will neither read nor write this field.

---

**OLD\_BUFFER**

**Explanation:** When the adapter is opened it sets this field with the address in shared RAM of the first bridge transmit buffer (the same as the NEXT\_BUFFER field). Thereafter, the adapter will neither read nor write this field.

---

**Initiating Transmission**

The application program must follow these steps to transmit frames using the bridge transmit control area.

1. Determine the number of transmit buffers that are currently available by the following calculation using 8-bit unsigned arithmetic.

$$\text{Number of buffers available} = \text{MAX\_BUFFERS} - \text{INPUT\_COUNT} + \text{RETURN\_COUNT}$$

2. If buffers are available, the application program then fills the data area of the buffer(s), sets the BUFFER\_LENGTH field to the length of data in the buffer, and sets the XMIT\_CONTROL field in the buffer appropriately (see "The Bridge Transmit Buffer Layout" on page 7-72).

If an insufficient number of buffers are available to hold the entire frame to be transmitted, the application program may fill the available buffers and wait until additional buffers become available. The application program must not update the INPUT\_COUNT field until the entire frame is copied into the bridge transmit buffers and the "last-buffer-indicator" bit in the XMIT\_CONTROL field has been set.

3. After the application program has placed an entire frame into the bridge transmit buffers, it must update the bridge transmit control area as follows:
  - It must update the NEXT\_BUFFER field to point to the next available buffer (that is, the contents of the BUFFER\_POINTER field of the last buffer used are stored in the NEXT\_BUFFER field of the bridge transmit control area).
  - It must increment the INPUT\_COUNT field by the number of bridge transmit buffers used by the frame.

## RECEIVED.BRIDGE.DATA

The application program must ensure that the "last-buffer-indicator" bit in the XMIT\_CONTROL field has been set in the last buffer before updating the INPUT\_COUNT field.

**Note:** Failure to do this can result in an adapter check with a reason code of X'0001' (program-detected error).

4. The application program should then set the frame forward request (ISRA-odd bit 6) to indicate to the adapter that a bridge frame is ready for forwarding.

After the adapter has transmitted the frame (successfully or unsuccessfully) it will update the OUTPUT\_COUNT field of the bridge transmit control area and set the frame forward complete (ISRP-odd bit 2) bit to interrupt the PC system.

5. The application program must then update its fields in the bridge transmit control area so that joint buffer management may be maintained. For example, it would set the RETURN\_COUNT field equal to the OUTPUT\_COUNT field value.

**Note:** The BUFFER\_POINTER field of the last buffer of a frame always points to the first buffer of the next frame to be transmitted, because the bridge transmit buffers are linked in a circular queue.

### The Bridge Transmit Buffer Layout

Bridge frames are transmitted out of special buffers dedicated to bridge traffic. These buffers are formatted when the adapter is opened with bridge functions selected and are the same length as the receive buffers in that adapter. If both adapters are opened with the same parameters, the logic required to copy frames from the receive buffers of one adapter to the transmit buffers of the other is minimal.

There are two formats for the bridge transmit buffers: one for buffers filled by the application program with the frame to forward and another for the buffers that are returned to the PC system after the adapter has transmitted the frame. The formats for the buffers are:

#### The Bridge Transmit Buffers (before transmission)

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	FRAME_LENGTH	2	The length of the entire frame (including CRC), In first buffer only
2	BUFFER_POINTER	2	Offset to the address of the next buffer plus 2
4	XMIT_CONTROL	1	Control bits
5		1	Reserved
6	BUFFER_LENGTH	2	Length of the data in this buffer
8	FRAME_DATA	n	Frame data

---

#### FRAME\_LENGTH

**Explanation:** This field must be set by the application program to indicate the entire length of the frame to be transmitted. This field is valid in only the first buffer of the frame. The field is reserved in the remainder of buffers for the frame.

---

#### BUFFER\_POINTER

**Explanation:** This field points to the BUFFER\_POINTER field of the next available bridge transmit buffer. This field must not be altered by the application program. It must only be interrogated.

---

**XMIT\_CONTROL**

**Explanation:** This field is set by the application program to control the CRC generation and to flag the last buffer of a frame. The bit meanings are:

**Bits 7-5** Reserved

**Bit 4** CRC Generation (required in first buffer only)

0 = CRC is to be calculated by the adapter and inserted after the buffer data.

1 = The last 4 bytes of data in the last buffer for the frame are the CRC to be sent with the frame.

**Bits 3-1** Reserved

**Bit 0** Last buffer indicator

0 = There are additional buffer(s) for the frame.

1 = This is the last buffer for the frame.

---

**BUFFER\_LENGTH**

**Explanation:** This field contains the total number of bytes to be transmitted from this buffer.

**The Bridge Transmit Buffers** (after transmission)

OFF-SET	PARAMETER NAME	BYTE LEN	DESCRIPTION
0	LAST_BUFFER	2	The address of the last buffer in the frame, plus 2, first buffer only
2	BUFFER_POINTER	2	Offset to the address of the next buffer plus 2
4	XMIT_STATUS	1	Transmit completion status, last buffer only
5	STRIP_FS	1	The FS byte as removed from the frame, last buffer only
6	FRAME_LENGTH	2	Length of the entire frame (including CRC), last buffer only
8	NUMBER_BUFFERS	1	The number of buffers in the frame, last buffer only
9		n	Data area

---

**LAST\_BUFFER**

**Explanation:** This field contains the address of the BUFFER\_POINTER field of the last buffer in the transmitted frame. This field is valid in only the first buffer of the frame. The field is reserved in the remainder of buffers for the frame.

---

**BUFFER\_POINTER**

**Explanation:** This field points to the BUFFER\_POINTER field of the next available bridge transmit buffer.

---

**XMIT\_STATUS**

**Explanation:** This field is set by the adapter to indicate the transmit completion status of the frame. The bit meanings are:



**Bit 7** Purge indicator

0 = The frame has not been purged.

1 = The frame was not transmitted but was purged from the transmit queue by the adapter. The adapter may purge frames from the transmit queue under three conditions:

1. The ring the adapter is connected to is beaconing.
2. The source routing indicator bit in the source address field is not set.

**Note:** This bit is not checked by the Token-Ring Network Adapter/A, Token-Ring Network 16/4 Adapter/A, and Token-Ring Network 16/4 Adapters.

3. The frame is a MAC frame with a source class and destination class of "ring station."

When this bit is set, bits 6 through 1 are not set.

**Bit 6** Strip frame error detect (SFED)

When this bit is set, the adapter detected a transmission error when removing the frame from the ring.

**Bit 5** Strip error detect indicator (SEDI)

This bit is a representation of the error-detected bit found in the ending delimiter (ED) byte of the frame after transmission.

**Bits 4-1** Transmit completion code 4-1

These bits represent a transmit completion code which is placed into the last transmit buffer of a frame. The field definitions are:

Bits 4-3 – Parallel completion

- 00 = Good completion
- 01 = DMA parity error
- 10 = DMA underrun
- 11 = Next buffer available

Bits 2-1 – Serial completion

- 00 = Good completion
- 01 = PTT timeout
- 10 = Corrupted token
- 11 = Either an implicit or explicit abort was stripped

**Bit 0** Last buffer indicator

This bit is always set to indicate the last buffer of a transmitted frame.

---

**STRIP\_FS**

**Explanation:** This field contains the frame status (FS) byte of the frame after transmission. This field is valid for only the last buffer of a frame. It is valid only when the purge bit in the XMIT\_STATUS field is zero.

---

**FRAME\_LENGTH**

**Explanation:** This field contains the value of the FRAME\_LENGTH field before the frame was transmitted. This field is valid for only the last buffer of a frame.

---

## NUMBER\_BUFFERS

**Explanation:** This field contains the number of bridge transmit buffers used for the frame. This field is valid for only the last buffer of a frame.

**Note:** This field overwrites the first byte of data in the last buffer of the frame.

## POS Registers in Token-Ring Network Adapters with Micro Channel

The program option selection (POS) registers in Token-Ring Network adapters with Micro Channel are identified as follows:

For Token-Ring Network Adapter/A

**POS register 0**      ID low byte = 00  
**POS register 1**      ID high byte = E0

For Token-Ring Network 16/4 Adapter/A

**POS register 0**      ID low byte = 01  
**POS register 1**      ID high byte = E0

For all Token-Ring Network adapters with Micro Channel

### POS register 2

**Bit 7**      RAM Address Compare Bit (A19)  
**Bit 6**      RAM Address Compare Bit (A18)  
**Bit 5**      RAM Address Compare Bit (A17)  
**Bit 4**      RAM Address Compare Bit (A16)  
**Bit 3**      RAM Address Compare Bit (A15)  
**Bit 2**      RAM Address Compare Bit (A14)  
**Bit 1**      RAM Address Compare Bit (A13)  
**Bit 0**      Card Enable Bit (0 = Disabled, 1 = Enabled)

For Token-Ring Network Adapter/A

### POS register 3

**Bit 7**      Encoded Interrupt Level LSB <sup>2</sup>  
**Bit 6**      Reserved  
**Bit 5**      Reserved  
**Bit 4**      Reserved  
**Bit 3**      Shared RAM Size MSB (16K Fixed)  
**Bit 2**      Shared RAM Size LSB (16K Fixed)  
**Bit 1**      Reserved  
**Bit 0**      Primary/Alternate Adapter Bit (0 = Primary, 1 = Alternate)

---

<sup>2</sup> Bit 7 of POS register 3 and bit 0 of POS register 4 together indicate the Encoded Interrupt Level where 00 = 2, 01 = 3, 10 = 10, and 11 = 11.

<sup>3</sup> Bits 3 and 2 = (00 = 8K, 01 = 16K, 10 = 32K, and 11 = 64K)

For Token-Ring Network 16/4 Adapter/A

**POS register 3**

- Bit 7** Encoded Interrupt Level LSB <sup>2</sup>
- Bit 6** Reserved
- Bit 5** Reserved
- Bit 4** Reserved
- Bit 3** Shared RAM Size MSB <sup>3</sup>
- Bit 2** Shared RAM Size LSB <sup>3</sup>
- Bit 1** Adapter Data Rate (0=4 Mbps, 1=16 Mbps)
- Bit 0** Primary/Alternate Adapter Bit (0=Primary, 1=Alternate)

For all Token-Ring Network adapters with Micro Channel

**POS register 4**

- Bit 7** BIOS/MMIO Address Compare Bit (A19)
- Bit 6** BIOS/MMIO Address Compare Bit (A18)
- Bit 5** BIOS/MMIO Address Compare Bit (A17)
- Bit 4** BIOS/MMIO Address Compare Bit (A16)
- Bit 3** BIOS/MMIO Address Compare Bit (A15)
- Bit 2** BIOS/MMIO Address Compare Bit (A14)
- Bit 1** BIOS/MMIO Address Compare Bit (A13)
- Bit 0** Encoded Interrupt Level MSB <sup>2</sup>

For all Token-Ring Network adapters with Micro Channel

**POS register 5**

- Bit 7** Channel Check (0=Active; 1=Inactive)
- Bit 6** Channel Check Status (only if active) 0=Present, 1=Not Present
- Bit 5** Reserved
- Bit 4** Reserved
- Bit 3** Reserved
- Bit 2** Reserved
- Bit 1** Reserved
- Bit 0** Reserved

For all Token-Ring Network adapters with Micro Channel

POS registers 6,7 are reserved

## Early Token Release

Early Token Release (ETR) is a method of reducing the delay or latency that can occur within a token ring due to the signal propagation time on a 16 Mbps Token-Ring Network as intermediate nodes repeat the signal. As the physical ring length (number of active stations) and/or the data rate are increased, the number of bytes required to “fill” the ring increases. The maximum efficiency of the standard token protocol may be affected if a high percentage of the frames are shorter than the latency of the ring, resulting in an overall average frame size that is less than the ring latency. This is due to the idle characters that must be inserted by a transmitting station until it has recognized its source address in the header of the returning frame and subsequent releasing of a new token. An early token release protocol allows a transmitting station to release a new token as soon as it has completed frame transmission, whether or not the frame header has returned to that station.

ETR can optimize the use of the available ring bandwidth when the average frame size is less than the ring length by decreasing the delay that subsequent stations would see before receiving a token. Frames from two (or more) different stations can be on the ring simultaneously with this enhancement to the protocol.

This mode of operation (ETR) alters the behavior of the priority protocol in that the station may not have received the header of the transmitted frame prior to releasing the token. When this occurs, the priority of the next token is the same as the token that was captured by the station prior to transmitting its frame. Any priority reservations that may appear in the header of the returning frame are ignored. However, application programs that offer a mixture of both long and short frames will see little or no negative performance impact due to this mode of operation.

The maximum ring efficiency is defined as that portion of the bandwidth that is available for frame information transfer. The maximum ring efficiency that can be achieved on a ring is simply the ratio of the average number of bytes per frame to the total number of bytes expended on the network to transfer the frame. If the average frame contains 100 bytes of information, but requires 25 idle bytes per frame, then a maximum ring efficiency of 80 % can be achieved. This is expressed as:

$$\text{Maximum Ring Eff} = \frac{\text{Number of Frame bytes}}{\text{Frame bytes} + \text{Idle bytes}}$$

$$\text{Maximum Ring Eff} = \frac{100}{100 + 25} = 0.80$$

The number of idle bytes will vary, depending upon the total frame length. The standard token protocol requires the sender to transmit idle characters onto the ring until it recognizes its own address in the returning header of the frame. The length of the idle field is thus a function of the frame size relative to the total latency of the ring.

The typical latency of a 4 Mbps ring is approximately 50 to 100 bytes. However, 16 Mbps rings are more likely to be used in large backbone rings where the ring latency could exceed 400 bytes. The benefits of the ETR protocol will be greater where a 16 Mbps ring spans a large area. With ETR, no idle characters are required to fill such a ring.<sup>4</sup>

---

<sup>4</sup> Hardware implementation may require a small inter-frame gap between the end of one frame and the beginning of the next frame.

## **Additional Factors to Consider**

Small frames are normally used for frame acknowledgements and command/control data. High link utilization is more likely to be associated with simultaneous file transfers by several stations, thus having a higher percentage of "long" frames. Also, since the short frames could be back-to-back, a sequence of 40 frames at 50 bytes each will cause about the same delay to a priority station as one continuous 2000 byte frame, assuming the priority station had to simply wait its turn for the token. Fairness of service insures that all stations continue to receive their fair share of the ring bandwidth during periods of heavy load.

It is interesting to note that priority access provides the most benefit to a user only when the link utilization is extremely high. Utilizations at these high levels should only occur during bursts of activity rather than for extended periods. During periods of low utilization, the priority reservation scheme should be invoked less frequently, thus offering negligible transfer time improvement.

## **ETR Conclusion**

ETR can provide a substantial increase in available data bandwidth when the average frame size is shorter than the latency of the ring.

The priority reservation protocol may be affected by the early token release mode of operation due to the loss of some reservations. However, based on simulation results, there appears to be negligible delay at utilizations below 80%, and the delays above 80% utilization should be limited for most rings exhibiting a mixture of long and short frames. The elimination of ring latency as a limiting factor in maximum ring efficiency will be a major benefit of the ETR protocol.

---

## Chapter 8. Token-Ring Network Adapters

Token-Ring Network Adapters with PC I/O Bus	8-2
Power	8-2
Adapter I/O Pins	8-2
IBM PC I/O Channel Pin Definitions	8-4
Token-Ring Network Adapters with Micro Channel	8-6
Power	8-6
Adapter I/O Pins	8-6
Token-Ring Network Adapter Cable	8-10

## Token-Ring Network Adapters with PC I/O Bus

### Power

The Adapter and Adapter IIs use one voltage from the PC system, +5 volts DC.

### Adapter I/O Pins

The pins at the PC system I/O connector are used as described in the following chart.

Pin	Dir	Card Driver Type	Signal Name
A01	O	OC	- I/O CH CHK
A02	B	TS	+ D7
A03	B	TS	+ D6
A04	B	TS	+ D5
A05	B	TS	+ D4
A06	B	TS	+ D3
A07	B	TS	+ D2
A08	B	TS	+ D1
A09	B	TS	+ D0
A10	O	OC	+ I/O CH RDY
A11	I		+ AEN
A12	I		+ A19
A13	I		+ A18
A14	I		+ A17
A15	I		+ A16
A16	I		+ A15
A17	I		+ A.4
A18	I		+ A13
A19	I		+ A12
A20	I		+ A11
A21	I		+ A10
A22	I		+ A9
A23	I		+ A8
A24	I		+ A7
A25	I		+ A6
A26	I		+ A5
A27	I		+ A4
A28	I		+ A3
A29	I		+ A2
Legend			
<b>Dir</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration
B	Bidirectional line to/from the adapter.	TS	Tri-state
I	Input to the adapter.		
O	Output from the adapter.		
PWR	Power or Ground input to the adapter.		

Pin	Dir	Card Driver Type	Signal Name
A30	I		+ A1
A31	I		+ A0
B01	PWR		Ground
B02	I		- Reset DRV
B03	PWR		+ 5 VDC
B04	B	OC	+ IRQ2
B05	---		- 5 VDC
B06	---		+ DRQ2
B07	---		- 12 VDC
B08	---		
B09	---		+ 12 VDC
B10	PWR		Ground
B11	I		- MEMW
B12	I		- MEMR
B13	I		- IOW
B14	I		- IOR
B15	---		- DACK3
B16	---		+ DRQ3
B17	---		- DACK1
B18	---		+ DRQ1
B19	I		- MREF
B20	---		+ CLK
B21	B	OC	+ IRQ7
B22	B	OC	+ IRQ6
B23	---		+ IRQ5
B24	---		+ IRQ4
B25	B		+ IRQ3
B26	---		- DACK2
B27	---		+ T/C
B28	---		+ ALE
B29	PWR		+ 5 VDC
B30	---		+ OSC
B31	PWR		Ground
Legend			
<b>Dir</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration
B	Bidirectional line to/from the adapter.	TS	Tri-state
I	Input to the adapter.		
O	Output from the adapter.		
PWR	Power or Ground input to the adapter.		



### IBM PC I/O Channel Pin Definitions

The signal requirements are further explained in the *Technical Reference* of your Personal Computer.

#### Pin A01 - I/O CH CK

I/O channel check line is used to generate a non-maskable interrupt to the IBM PC if an uncorrectable adapter or system interface error is detected. The line is driven active (low) to indicate an error and it must remain low for at least two IBM PC system clock cycles to be detected. This line must be driven with an open collector configuration.<sup>1</sup>

#### Pins A02 through A09 + D7 to D0

Data bus bits D0 to D7 are for the processor, memory, and the adapter. D0 is the least significant bit and D7 is the most significant bit. Data is valid on these lines after the issuance of a read or write command and must remain valid for the duration of the command. The lines are active when high.

#### Pin A10 + I/O CH RDY

The I/O channel ready line indicates ready when active (high). This line is pulled low by the memory or I/O adapter to lengthen a memory or I/O cycle. I/O CH RDY is activated as soon as the decision is made in the adapter to service the IBM PC request. The early latching of this signal ensures that the adapter will always incur an extra wait. The maximum time that this line can be pulled low (inactive) by the adapter is 10 clock cycles.

#### Pin A11 + AEN

The address enable line is used to de-gate the processor and other adapters from the I/O bus to allow DMA transfers to take place. The line is active (high) during all DMA operations and inactive (low) during I/O operations. When this line is active, the DMA controller has control of the bus, data bus, and the memory and I/O read and write command lines. This line must be used as part of the adapter select decode for I/O operations to prevent false selects during DMA operations.

#### Pins A12 through A31 + A19 to A0

Address bits 19 through 0 are used to address memory and I/O adapters within the IBM PC system. A0 is the least significant bit and A19 is the most significant bit. These 20 address lines allow access of up to 1 megabyte of memory. The lower 16 bits are used for I/O operations and all 16 must be decoded by the adapter. These lines are active when high. During an I/O operation the unused bits A16 through A19 will be low. The addresses on this bus are generated by either the IBM PC processor or the DMA controller and gated by the line AEN.

#### Pin B02 + Reset Drv

The reset drive line is used to reset or initialize adapter logic. The line is active when high and will be active for at least 100 milliseconds at power on or during a low line voltage condition. This line is synchronized to the falling edge of the IBM PC system clock.

#### Pins B21, B22, B25 and B04 + IRQ 7, 6, 3, 2

The IRQ request lines 2 to 7 are used by the adapter to signal the IBM PC for attention. IRQ 2 is the highest priority and IRQ 7 is the lowest. These lines are normally high and a request is created by pulsing the line low then high. The interrupt is recognized on the low-to-high transition on the IRQ line. The IRQ lines must be capable of being enabled and disabled by the IBM PC application program. The lines should be driven with an open collector configuration to allow sharing.

---

<sup>1</sup> The term "open collector" refers to either an open collector device or a tri-state device with the input grounded and the enable controlling the output.

### **Pin B11 - MEMW**

The memory write command line, when active (low), instructs the memory at the present address to store the data present on the data bus. The data on the data bus will be valid after the MEMW is active and must be held on the bus until the MEMW goes inactive. The address on the address bus must be valid prior to MEMW going active and must remain valid until the MEMW goes inactive.

### **Pin B12 - MEMR**

The memory read command line, when active (low), instructs the memory at the present address to drive its data onto the data bus. The address on the address bus must be valid prior to the MEMR line going active and must remain valid during the read operation. The memory data must remain valid on the data bus until the MEMR line goes inactive.

### **Pin B13 - IOW**

The input/output write command line, when active (low), instructs the I/O port at the present address to store the data present on the data bus. The address on the address bus must be valid prior to the IOW line going active and must remain valid during the write operation. The data on the data bus will be valid after the IOW line goes active and must remain valid until the IOW line goes inactive.

### **Pin B14 - IOR**

The input/output read command line, when active (low), instructs the I/O port at the present address to drive its data onto the data bus. The address on the address bus must be valid prior to the IOR line going active and must remain valid during the read operation. The data on the data bus must be valid after the IOR line goes active and must remain valid until the IOR line goes inactive.

### **Pin B19 - MREF**

The memory refresh line, when active (low) is used by the IBM PC system logic to indicate to the adapter that a memory refresh cycle is in progress.

## Token-Ring Network Adapters with Micro Channel

### Power

The PC system with Micro Channel uses one voltage from the IBM Personal System/2, +5 volts DC.

### Adapter I/O Pins

The signal requirements are further explained in the *Technical Reference* provided with your computer.

All circuit loads are no more than the equivalent of two LS TTL loads per pin.

The pins at the PC system I/O connector are used as described in the following chart.

Pin	Dir	Card Driver Type	Signal Name																								
A01	I		- CD Setup																								
A02	I		+ MADE 24																								
A03	PWR		GND																								
A04	I		+ A 11																								
A05	I		+ A 10																								
A06	I		+ A 09																								
A07	PWR		+ 5 Power																								
A08	I		+ A 08																								
A09	I		+ A 07																								
A10	I		+ A 06																								
A11	PWR		+ 5 Power																								
A12	I		+ A 05																								
A13	I		+ A 04																								
A14	I		+ A 03																								
A15	PWR		+ 5 Power																								
A16	I		+ A 02																								
A17	I		+ A 01																								
A18	I		+ A 00																								
A19	---		+ 12 Power																								
A20	I		- ADL																								
A21	---		- PREEMPT																								
A22	---		- BURST																								
A23	---		-12 Power																								
A24	---		+ ARB 00																								
A25	---		+ ARB 01																								
A26	---		+ ARB 02																								
<p>Legend</p> <table> <tr> <td><b>D/R</b></td> <td><b>Meaning</b></td> <td><b>Type</b></td> <td><b>Meaning</b></td> </tr> <tr> <td>---</td> <td>The pin is not used by the adapter.</td> <td>OC</td> <td>Open Collector or tri-state in open collector configuration</td> </tr> <tr> <td>B</td> <td>Bidirectional line to/from the adapter.</td> <td>TP</td> <td>Totem Pole</td> </tr> <tr> <td>I</td> <td>Input to the adapter.</td> <td>TS</td> <td>Tri-state</td> </tr> <tr> <td>O</td> <td>Output from the adapter.</td> <td></td> <td></td> </tr> <tr> <td>PWR</td> <td>Power or Ground input to the adapter.</td> <td></td> <td></td> </tr> </table>				<b>D/R</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>	---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration	B	Bidirectional line to/from the adapter.	TP	Totem Pole	I	Input to the adapter.	TS	Tri-state	O	Output from the adapter.			PWR	Power or Ground input to the adapter.		
<b>D/R</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>																								
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration																								
B	Bidirectional line to/from the adapter.	TP	Totem Pole																								
I	Input to the adapter.	TS	Tri-state																								
O	Output from the adapter.																										
PWR	Power or Ground input to the adapter.																										

Pin	Dir	Card Driver Type	Signal Name																								
A27	---		- 12 Power																								
A28	---		+ ARB 03																								
A29	---		+ ARB/-GNT																								
A30	---		- TC																								
A31	PWR		+ 5 Power																								
A32	I		- S0																								
A33	I		- S1																								
A34	I		+ M/-IO																								
A35	---		+ 12 Power																								
A36	O	TP	+ CD CHRDY																								
A37	B	TS	+ D 00																								
A38	B	TS	+ D 02																								
A39	PWR		+ 5 Power																								
A40	B	TS	+ D 05																								
A41	B	TS	+ D 06																								
A42	B	TS	+ D 07																								
A43	PWR		GND																								
A44	---		- DS 16 RTN																								
A45	I		- Refresh																								
A46			Key																								
A47			Key																								
A48	PWR		+ 5 Power																								
A49	B	TS	+ D 10																								
A50	B	TS	+ D 11																								
A51	B	TS	+ D 13																								
A52	---		+ 12 Power																								
A53	---		Reserved																								
A54	I		- SBHE																								
A55	O	TP	- CD DS 16																								
A56	PWR		+ 5 Power																								
A57	---		- IRQ 14																								
A58	---		- IRQ 15																								
B01	---		Audio GND																								
B02	---		Audio																								
B03	PWR		GND																								
B04	---		OSC																								
B05	PWR		GND																								
B06	I		+ A 23																								
B07	I		+ A 22																								
<p>Legend</p> <table> <tbody> <tr> <td><b>D/R</b></td> <td><b>Meaning</b></td> <td><b>Type</b></td> <td><b>Meaning</b></td> </tr> <tr> <td>---</td> <td>The pin is not used by the adapter.</td> <td>OC</td> <td>Open Collector or tri-state in open collector configuration</td> </tr> <tr> <td>B</td> <td>Bidirectional line to/from the adapter.</td> <td>TP</td> <td>Totem Pole</td> </tr> <tr> <td>I</td> <td>Input to the adapter.</td> <td>TS</td> <td>Tri-state</td> </tr> <tr> <td>O</td> <td>Output from the adapter.</td> <td></td> <td></td> </tr> <tr> <td>PWR</td> <td>Power or Ground input to the adapter.</td> <td></td> <td></td> </tr> </tbody> </table>				<b>D/R</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>	---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration	B	Bidirectional line to/from the adapter.	TP	Totem Pole	I	Input to the adapter.	TS	Tri-state	O	Output from the adapter.			PWR	Power or Ground input to the adapter.		
<b>D/R</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>																								
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration																								
B	Bidirectional line to/from the adapter.	TP	Totem Pole																								
I	Input to the adapter.	TS	Tri-state																								
O	Output from the adapter.																										
PWR	Power or Ground input to the adapter.																										

## Adapters with Micro Channel

Pin	Dir	Card Driver Type	Signal Name																								
B08	I		+ A 21																								
B09	PWR		GND																								
B10	I		+ A 20																								
B11	I		+ A 19																								
B12	I		+ A 18																								
B13	PWR		GND																								
B14	I		+ A 17																								
B15	I		+ A 16																								
B16	I		+ A 15																								
B17	PWR		GND																								
B18	I		+ A 14																								
B19	I		+ A 13																								
B20	I		+ A 12																								
B21	PWR		GND																								
B22	O	OC	- IRQ 02 (09)																								
B23	O	OC	- IRQ 03																								
B24	---		- IRQ 04																								
B25	PWR		GND																								
B26	---		- IRQ 05																								
B27	---		- IRQ 06																								
B28	---		- IRQ 07																								
B29	PWR		GND																								
B30	---		Reserved																								
B31	---		Reserved																								
B32	O	OC	- CHCK (not used by Adapter/A)																								
B33	PWR		GND																								
B34	I		- CMD																								
B35	---		+ CHRDYRTN																								
B36	O	TP	- CD SFDBK																								
B37	PWR		GND																								
B38	B	TS	+ D 01																								
B39	B	TS	+ D 03																								
B40	B	TS	+ D 04																								
B41	PWR		GND																								
B42	I		+ CHRESET																								
B43	---		Reserved																								
B44	---		Reserved																								
B45	PWR		GND																								
B46			Key																								
<p>Legend</p> <table> <thead> <tr> <th>D/R</th> <th>Meaning</th> <th>Type</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>---</td> <td>The pin is not used by the adapter.</td> <td>OC</td> <td>Open Collector or tri-state in open collector configuration</td> </tr> <tr> <td>B</td> <td>Bidirectional line to/from the adapter.</td> <td>TP</td> <td>Totem Pole</td> </tr> <tr> <td>I</td> <td>Input to the adapter.</td> <td>TS</td> <td>Tri-state</td> </tr> <tr> <td>O</td> <td>Output from the adapter.</td> <td></td> <td></td> </tr> <tr> <td>PWR</td> <td>Power or Ground input to the adapter.</td> <td></td> <td></td> </tr> </tbody> </table>				D/R	Meaning	Type	Meaning	---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration	B	Bidirectional line to/from the adapter.	TP	Totem Pole	I	Input to the adapter.	TS	Tri-state	O	Output from the adapter.			PWR	Power or Ground input to the adapter.		
D/R	Meaning	Type	Meaning																								
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration																								
B	Bidirectional line to/from the adapter.	TP	Totem Pole																								
I	Input to the adapter.	TS	Tri-state																								
O	Output from the adapter.																										
PWR	Power or Ground input to the adapter.																										

Pin	Dir	Card Driver Type	Signal Name
B47			Key
B48	B	TS	+ D 08
B49	B	TS	+ D 09
B50	PWR		GND
B51	B	TS	+ D 12
B52	B	TS	+ D 14
B53	B	TS	+ D 15
B54	PWR		GND
B55	O	OC	- IRQ 10
B56	O	OC	- IRQ 11
B57	---		- IRQ 12
B58	PWR		GND
Legend			
<b>D/R</b>	<b>Meaning</b>	<b>Type</b>	<b>Meaning</b>
---	The pin is not used by the adapter.	OC	Open Collector or tri-state in open collector configuration
B	Bidirectional line to/from the adapter.	TP	Totem Pole
I	Input to the adapter.	TS	Tri-state
O	Output from the adapter.		
PWR	Power or Ground input to the adapter.		

## Token-Ring Network Adapter Cable

### Token-Ring Network Adapter Cable

A cable is required to connect the Token-Ring Network adapter to the cabling system used by the IBM Token-Ring Network. IBM P/N 6339098 is available for use with the IBM Cabling System. A cable with built-in filters housed in a nine-pin D-connector at the system unit end and a six-pin modular plug at the cabling end is available for type 3 media use. This cable is not available from IBM. Consult your IBM representative or local branch office for a list of suppliers.

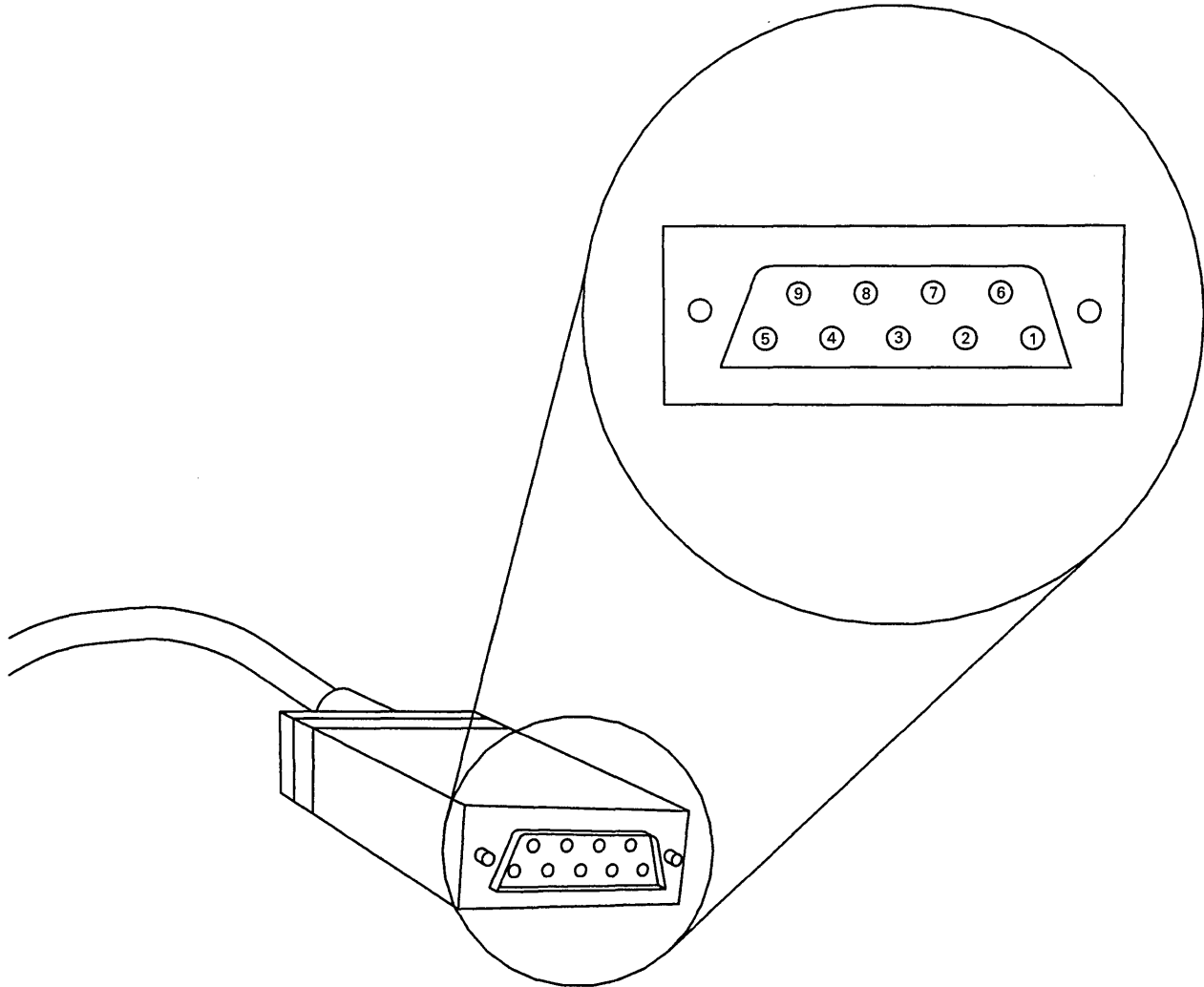


Figure 8-1. Adapter Cable D Connector

Table 8-1. P/N 6339098 Cable Schematic				
Adapter D Connector Pin	Wire Number	Wire Color	Data Connector Pin	Usage
Shield (ground)	1	Shield	Shield (ground)	Ground
1	4	Red	Red	Receive
2				Reserved
3				Reserved
4				Reserved
5	3	Black	Black	Transmit
6	5	Green	Green	Receive
7				Reserved
8				Reserved
9	2	Orange	Orange	Transmit

Table 8-2. Type 3 Media Filter Schematic				
Adapter D connector Pin	Wire Color	6-PIN Connector Pin		Usage
1	Orange	3		Receive
2				Reserved
3				Reserved
4				Reserved
5	Black	2		Transmit
6	Blue	4		Receive
7				Reserved
8				Reserved
9	Yellow	5		Transmit





## Appendix A. Valid Commands

The following charts show the valid commands found in control blocks. An **X** indicates what interface(s) the command may be directed to. Some commands sent to NETBIOS can request that the adapter complete the command before allowing the application program to continue. An additional bit (high order) is included in the hexadecimal command code to request this wait for completion.

### Notes:

1. Command codes not listed are reserved.
2. The commands used to operate the adapter directly, without the adapter support software, are not listed here. See Chapter 7, "The Token-Ring Network Adapter Interface."

## DLC and Direct Interface Commands

PSEUDO COMMAND NAME	Page	HEX Code	CCB1	CCB2	CCB3
BUFFER.FREE	3-3	27	x	x	x
BUFFER.GET	3-4	26	x	x	x
DIR.CLOSE.ADAPTER	3-25	04	x	x	x
DIR.CLOSE.DIRECT	3-27	34		x	x
DIR.DEFINE.MIF.ENVIRONMENT	3-27	2B	x		
DIR.INITIALIZE	3-29	20	x	x	x
DIR.INTERRUPT	3-33	00	x	x	x
DIR.MODIFY.OPEN.PARMS	3-33	01	x		
DIR.OPEN.ADAPTER	3-34	03	x	x	x
DIR.OPEN.DIRECT	3-46	35		x	x
DIR.READ.LOG	3-47	08	x	x	x
DIR.RESTORE.OPEN.PARMS	3-50	02	x		
DIR.SET.EXCEPTION.FLAGS	3-50	2D		x	x
DIR.SET.FUNCTIONAL.ADDRESS	3-52	07	x	x	x
DIR.SET.GROUP.ADDRESS	3-52	06	x	x	x
DIR.SET.USER.APPENDAGE	3-53	2D	x		
DIR.STATUS	3-54	21	x	x	x
DIR.TIMER.CANCEL	3-57	23	x	x	x
DIR.TIMER.CANCEL.GROUP	3-58	2C	x	x	x
DIR.TIMER.SET	3-59	22	x	x	x
DLC.CLOSE.SAP	3-5	16	x	x	x
DLC.CLOSE.STATION	3-6	1A	x	x	x
DLC.CONNECT.STATION	3-7	1B	x	x	x
DLC.FLOW.CONTROL	3-8	1D	x	x	x
DLC.MODIFY	3-9	1C	x	x	x
DLC.OPEN.SAP	3-11	15	x	x	x
DLC.OPEN.STATION	3-16	19	x	x	x
DLC.REALLOCATE	3-19	17	x	x	x
DLC.RESET	3-20	14	x	x	x

PSEUDO COMMAND NAME	Page	HEX Code	CCB1	CCB2	CCB3
DLC.SET.THRESHOLD	3-22	33		x	
DLC.STATISTICS	3-23	1E	x	x	x
PDT.TRACE.OFF	3-67	25	x		
PDT.TRACE.ON	3-59	24	x		
PURGE.RESOURCE	3-68	36			x
READ	3-69	31		x	x
READ.CANCEL	3-77	32		x	x
RECEIVE	3-77	28	x	x	x
RECEIVE.CANCEL	3-83	29	x	x	x
RECEIVE.MODIFY	3-84	2A	x		x
TRANSMIT.DIR.FRAME	3-88	0A	x	x	x
TRANSMIT.I.FRAME	3-88	0B	x	x	x
TRANSMIT.TEST.CMD	3-89	11	x	x	x
TRANSMIT.UI.FRAME	3-90	0D	x	x	x
TRANSMIT.XID.CMD	3-90	0E	x	x	x
TRANSMIT.XID.RESP.FINAL	3-90	0F	x	x	x
TRANSMIT.XID.RESP.NOT.FINAL	3-91	10	x	x	x

## NETBIOS Commands

PSEUDO COMMAND NAME	Page	Hex Code	
		No Wait	Wait
NCB.ADD.GROUP.NAME	4-14	B6	36
NCB.ADD.NAME	4-14	B0	30
NCB.CALL	4-15	90	10
NCB.CANCEL	4-16		35
NCB.CHAIN.SEND	4-17	97	17
NCB.CHAIN.SEND.NO.ACK	4-18	F2	72
NCB.DELETE.NAME	4-19	B1	31
NCB.FIND.NAME	4-20	F8	78
NCB.HANG.UP	4-21	92	12
NCB.LAN.STATUS.ALERT	4-22		F3
NCB.LISTEN	4-22	91	11
NCB.RECEIVE	4-23	95	15
NCB.RECEIVE.ANY	4-24	96	16
NCB.RECEIVE.BROADCAST.DATAGRAM	4-25	A3	23
NCB.RECEIVE.DATAGRAM	4-26	A1	21
NCB.RESET	4-26		32
NCB.SEND	4-30	94	14
NCB.SEND.BROADCAST.DATAGRAM	4-31	A2	22
NCB.SEND.DATAGRAM	4-32	A0	20
NCB.SEND.NO.ACK	4-32	F1	71
NCB.SESSION.STATUS	4-33	B4	34
NCB.STATUS	4-35	B3	33
NCB.TRACE (For Local Area Network Support Program only)	4-37	F9	79
NCB.UNLINK	4-41		70



---

## Appendix B. Return Codes

CCB Return Codes Listed by Interface	B-2
CCB Return Codes Listed by Command	B-4
Return Codes for CCB1 Commands	B-4
Return Codes for CCB2 Commands	B-6
Return Codes for CCB3 Commands	B-9
CCB Return Codes Cause and Action	B-12
DLC Status Codes	B-25
DLC Status Table	B-25
DLC Status Codes	B-26
Suggested Actions in Response to DLC Status	B-26
NCB Return Codes	B-27
NCB Return Codes Listed by Command	B-29
NCB Return Codes Cause and Action	B-30
Adapter Status Parameter Table	B-38
Frame Status	B-39
Exception Indications	B-40
Adapter Check	B-40
Adapter Check for CCB1	B-40
Adapter Check for CCB2	B-40
Adapter Check for CCB3	B-42
Token-Ring Network Adapter Check Reason Codes for All CCBs	B-43
PC Network Adapter Check Reason Codes for All CCBs	B-44
Network Status	B-44
Network Status for CCB1	B-44
Network Status for CCB2	B-44
Network Status for CCB3	B-46
Token-Ring Network Status Codes for All CCBs	B-47
PC Network Status Codes for All CCBs	B-48
Bring-Up Errors for All CCBs	B-48
Bring-up Error Codes for All CCBs	B-48
Token-Ring Network Adapter Open Errors for All CCBs	B-49
Open Error Codes for All CCBs	B-49
Phases	B-49
Errors	B-50
Suggested Actions in Response to Open Errors	B-50
Recommended Actions Table	B-52
PC Network Adapter Open Errors for All CCBs	B-53
PC System Detected Errors	B-53
PC System Detected Errors for CCB1	B-53
PC System Detected Errors for CCB2	B-54
PC System Detected Errors for CCB3	B-56
System Action Exceptions for OS/2	B-58
System Action Exceptions for CCB2	B-58
System Action Exceptions for CCB3	B-60

## CCB Return Codes

This chapter includes:

- DLC and Direct Interface return codes
- NETBIOS return codes
- Other reason and status codes
- Exception indications
- Formats of special returned tables.

### CCB Return Codes Listed by Interface

Code	Meaning	CCB1	CCB2	CCB3
00	Operation completed successfully	X	X	X
01	Invalid command code	X	X	X
02	Duplicate command, one already outstanding	X	X	X
03	Adapter open, should be closed	X	X	X
04	Adapter closed, should be open	X	X	
05	Required parameter(s) not provided	X	X	X
06	Option(s) invalid or incompatible	X	X	X
07	Command canceled, unrecoverable failure	X	X	X
08	Unauthorized access priority	X	X	X
09	Adapter not initialized, should be initialized	X		
0A	Command canceled by user request	X	X	X
0B	Command canceled, adapter closed while command in progress	X	X	X
0C	Command completed successfully, adapter not open	X		
10	Adapter open, NETBIOS not operational	X		
11	DIR.TIMER.SET or DIR.TIMER.CANCEL error	X	X	X
12	Available work area exceeded	X		
13	Invalid LOG.ID	X	X	X
14	Invalid shared RAM segment or size	X		
15	Lost log data, inadequate buffer space, log reset	X	X	X
16	Requested buffer size exceeds pool length	X	X	X
17	Command invalid, NETBIOS operational	X		
18	Invalid buffer length	X	X	X
19	Inadequate buffers available for request	X	X	X
1A	User length too large for buffer length	X	X	X
1B	The CCB_PARAM_TAB pointer is invalid	X	X	X
1C	A pointer in the CCB parm table is invalid	X	X	X
1D	Invalid CCB_ADAPTER value	X	X	X
20	Lost data on receive, no buffers available	X	X	X
21	Lost data on receive, inadequate buffer space	X	X	X
22	Error on frame transmission, check TRANSMIT.FS data	X	X	X
23	Error in frame transmit or strip process	X	X	X
24	Unauthorized MAC frame	X	X	X
25	Maximum commands exceeded	X	X	X
26	Unrecognized command correlator	X		
27	Link not transmitting I frames, state changed from link opened	X	X	X

## CCB Return Codes

Code	Meaning	CCB1	CCB2	CCB3
28	Invalid transmit frame length	X	X	X
30	Inadequate receive buffers for adapter to open	X	X	X
32	Invalid NODE_ADDRESS	X	X	X
33	Invalid adapter receive buffer length defined	X	X	X
34	Invalid adapter transmit buffer length defined	X	X	X
40	Invalid STATION_ID	X	X	X
41	Protocol error, link in invalid state for command	X	X	X
42	Parameter exceeded maximum allowed	X	X	X
43	Invalid SAP_VALUE or value already in use	X	X	X
44	Invalid routing information field	X	X	X
45	Requested group membership in non-existent group SAP	X	X	X
46	Resources not available	X	X	X
47	SAP cannot close unless all link stations are closed	X	X	X
48	Group SAP cannot close, individual SAPs not closed	X	X	X
49	Group SAP has reached maximum membership	X	X	X
4A	Sequence error: incompatible command in progress	X	X	X
4B	Station closed without remote acknowledgment	X	X	X
4C	Sequence error, cannot close, DLC commands outstanding	X	X	X
4D	Unsuccessful link station connection attempted	X	X	X
4E	Member SAP not found in group SAP list	X	X	X
4F	Invalid remote address, may not be a group address	X	X	X
50	Invalid pointer in CCB_POINTER field		X	
52	Invalid application program ID		X	X
53	Invalid application program key code		X	X
54	Invalid System Key code		X	X
55	Buffer is smaller than buffer size given on DLC.OPEN.SAP		X	X
56	Adapter's system process is not installed		X	
57	Inadequate stations available		X	X
58	Invalid CCB_PARAMETER_1 parameter		X	X
59	Inadequate queue elements to satisfy request		X	X
5A	Initialization failure, cannot open adapter		X	X
5B	Error detected in chained READ command		X	
5C	Direct stations not assigned to application program		X	X
5D	Device Driver Interface not installed		X	
5E	Requested adapter is not installed		X	X
5F	Chained CCBs must all be for the same adapter		X	
60	Adapter initializing, command not accepted		X	X
61	Number of allowed application programs has been exceeded		X	X
62	Command cancelled by system action		X	X
63	Direct stations not available		X	X
64	Invalid DDNAME parameter		X	X
65	Inadequate GDT selectors to satisfy request		X	X
67	Command cancelled, CCB resources purged			X
68	Application ID not valid for interface		X	X
69	Segment associated with request cannot be locked		X	



# CCB Return Codes

Code	Meaning	CCB1	CCB2	CCB3
FF	Command in progress	X	X	X

## CCB Return Codes Listed by Command

### Return Codes for CCB1 Commands

CCB1 Commands	Return Codes in Hex																													
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	20	21	
BUFFER.FREE	x	x			x					x																x	x	x		
BUFFER.GET	x	x			x					x													x			x		x		
DLC.CLOSE.SAP	x	x			x			x		x		x																x		
DLC.CLOSE.STATION	x	x			x			x		x		x																	x	
DLC.CONNECT.STATION	x	x	x		x			x		x		x														x	x	x		
DLC.FLOW.CONTROL	x	x			x			x		x		x																x		
DLC.MODIFY	x	x			x			x		x		x														x		x		
DLC.OPEN.SAP	x	x			x			x		x		x									x		x			x	x	x		
DLC.OPEN.STATION	x	x			x		x		x		x		x													x	x	x		
DLC.REALLOCATE	x	x			x			x		x																		x		
DLC.RESET	x	x			x			x		x		x														x		x		
DLC.STATISTICS	x	x			x			x		x		x								x						x		x		
RECEIVE	x	x	x		x			x		x		x													x	x	x	x	x	x
RECEIVE.CANCEL	x	x			x					x																		x		
RECEIVE.MODIFY	x	x	x		x			x		x		x													x	x	x	x	x	x
TRANSMIT.DIR.FRAME	x	x			x			x		x		x														x		x		
TRANSMIT.I.FRAME	x	x			x			x		x		x														x		x		
TRANSMIT.TEST.CMD	x	x			x			x		x		x														x		x		
TRANSMIT.UI.FRAME	x	x			x			x		x		x														x		x		
TRANSMIT.XID.CMD	x	x			x			x		x		x														x		x		
TRANSMIT.XID.RESP.FINAL	x	x			x			x		x		x														x		x		
TRANSMIT.XID.RESP.NOT.FINAL	x	x			x			x		x		x														x		x		
DIR.CLOSE.ADAPTER	x	x			x		x		x																			x		
DIR.DEFINE.MIF.ENVIRONMENT	x	x																								x		x		
DIR.INITIALIZE	x	x	x					x											x							x		x	x	
DIR.INTERRUPT	x	x						x		x																		x		
DIR.MODIFY.OPEN.PARMS	x	x	x		x			x		x										x		x				x		x	x	
DIR.OPEN.ADAPTER	x	x	x	x		x		x		x				x		x				x		x				x		x	x	
DIR.READ.LOG	x	x			x			x		x		x								x		x				x		x		
DIR.RESTORE.OPEN.PARMS	x	x			x			x		x		x																x		
DIR.SET.FUNCTIONAL.ADDRESS	x	x			x			x		x		x																x		
DIR.SET.GROUP.ADDRESS	x	x			x			x		x		x																	x	
DIR.SET.USER.APPENDAGE	x	x			x			x		x																x		x		
DIR.STATUS	x	x								x			x													x		x		
DIR.TIMER.CANCEL	x	x								x																		x		
DIR.TIMER.CANCEL.GROUP	x	x								x																		x		
DIR.TIMER.SET	x	x						x		x		x																x		
PDT.TRACE.ON	x		x					x				x															x			
PDT.TRACE.OFF	x																													
CCB1 Commands	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	20	21	

# CCB Return Codes

CCB1 Commands	Return Codes in Hex																													
	22	23	24	25	26	27	28	30	31	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	FF	
BUFFER.FREE													x																	
BUFFER.GET													x																	
DLC.CLOSE.SAP													x							x	x					x				
DLC.CLOSE.STATION													x							x					x	x				
DLC.CONNECT.STATION													x	x			x							x				x		
DLC.FLOW.CONTROL													x																	
DLC.MODIFY													x	x			x						x					x		
DLC.OPEN.SAP															x	x		x	x				x							
DLC.OPEN.STATION													x	x	x					x									x	
DLC.REALLOCATE													x																	
DLC.RESET													x																	
DLC.STATISTICS													x																	
RECEIVE													x																	
RECEIVE.CANCEL													x																	
RECEIVE.MODIFY													x																	
TRANSMIT.DIR.FRAME	x	x	x	x			x	x					x	x			x							x						
TRANSMIT.I.FRAME	x	x	x	x			x	x					x	x			x							x			:			
TRANSMIT.TEST.CMD	x	x	x	x			x	x					x	x			x							x						
TRANSMIT.UI.FRAME	x	x	x	x			x	x					x	x			x							x						
TRANSMIT.XID.CMD	x	x	x	x			x	x					x	x			x							x						
TRANSMIT.XID.RESP.FINAL	x	x	x	x			x	x					x	x			x							x						
TRANSMIT.XID.RESP.NOT.FINAL	x	x	x	x			x	x					x	x			x							x						
DIR.CLOSE.ADAPTER																														
DIR.DEFINE.MIF.ENVIRONMENT																														
DIR.INITIALIZE																														
DIR.INTERRUPT																														
DIR.MODIFY.OPEN.PARMS																														
DIR.OPEN.ADAPTER													x				x	x	x											
DIR.READ.LOG																														
DIR.RESTORE.OPEN.PARMS																														
DIR.SET.FUNCTIONAL.ADDRESS																														
DIR.SET.GROUP.ADDRESS																														
DIR.SET.USER.APPENDAGE																														
DIR.STATUS																														
DIR.TIMER.CANCEL																														
DIR.TIMER.CANCEL.GROUP																														
DIR.TIMER.SET																														
PDT.TRACE.ON																														
PDT.TRACE.OFF																														
CCB Commands	22	23	24	25	26	27	28	30	31	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	FF	
	Return Codes in Hex																													

# CCB Return Codes

## Return Codes for CCB2 Commands

CCB2 Commands	Return Codes in Hex																												
	00	01	02	03	04	05	06	07	08	0A	0B	0C	11	13	15	16	18	19	1A	1B	1C	1D	20	21	22	23	24	25	27
BUFFER.FREE	x	x			x			x												x	x	x							
BUFFER.GET	x	x			x			x										x		x		x							
DLC.CLOSE.SAP	x	x			x			x			x											x							
DLC.CLOSE.STATION	x	x	x		x			x			x											x							
DLC.CONNECT.STATION	x	x	x		x	x		x			x										x	x	x						
DLC.FLOW.CONTROL	x	x			x			x			x											x							
DLC.MODIFY	x	x			x	x		x	x		x										x	x	x						
DLC.OPEN.SAP	x	x			x	x	x	x	x		x					x	x				x	x	x						
DLC.OPEN.STATION	x	x			x	x		x	x		x										x	x	x						
DLC.REALLOCATE	x	x			x	x		x			x										x	x	x						
DLC.RESET	x	x			x			x			x											x							
DLC.SET.THRESHOLD	x	x			x	x		x													x		x						
DLC.STATISTICS	x	x			x	x		x			x					x					x	x	x						
READ	x	x			x	x	x	x		x	x										x		x						
READ.CANCEL	x	x			x			x														x							
RECEIVE	x	x	x		x	x	x	x		x	x									x	x		x	x	x				
RECEIVE.CANCEL	x	x			x			x														x							
TRANSMIT.DIR.FRAME	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.I.FRAME	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.TEST.CMD	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.UI.FRAME	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.XID.CMD	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.XID.RESP.FINAL	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
TRANSMIT.XID.RESP.NOT.FINAL	x	x			x	x	x	x	x		x										x	x	x			x	x	x	x
DIR.CLOSE.ADAPTER	x	x			x			x														x							
DIR.CLOSE.DIRECT	x	x			x			x			x												x						
DIR.INITIALIZE	x	x				x		x													x		x						
DIR.INTERRUPT	x	x			x			x			x												x						
DIR.OPEN.ADAPTER	x	x		x		x	x	x													x	x	x						
DIR.OPEN.DIRECT	x	x			x	x	x	x	x		x					x	x				x	x	x						
DIR.READ.LOG	x	x			x	x		x			x				x	x					x	x	x						
DIR.SET.EXCEPTION.FLAGS	x	x			x	x		x													x		x						
DIR.SET.FUNCTIONAL.ADDRESS	x	x			x			x			x												x						
DIR.SET.GROUP.ADDRESS	x	x			x			x			x												x						
DIR.STATUS	x	x			x	x		x				x									x		x						
DIR.TIMER.CANCEL	x	x			x			x					x										x						
DIR.TIMER.CANCEL.GROUP	x	x			x			x					x										x						
DIR.TIMER.SET	x	x			x			x		x	x		x										x						
<b>CCB2 Commands</b>	<b>00</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>	<b>07</b>	<b>08</b>	<b>0A</b>	<b>0B</b>	<b>0C</b>	<b>11</b>	<b>13</b>	<b>15</b>	<b>16</b>	<b>18</b>	<b>19</b>	<b>1A</b>	<b>1B</b>	<b>1C</b>	<b>1D</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>27</b>

CCB2 Commands	Return Codes in Hex																														
	28	30	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	52	53	54	55	56	57	58		
BUFFER.FREE						x																x	x	x		x	x				
BUFFER.GET						x																x	x	x				x			
DLC.CLOSE.SAP						x							x	x				x				x	x	x				x			
DLC.CLOSE.STATION						x											x	x				x	x	x				x			
DLC.CONNECT.STATION						x	x			x						x				x			x	x	x				x		
DLC.FLOW.CONTROL						x																	x	x	x				x		
DLC.MODIFY						x		x			x					x					x			x	x	x				x	
DLC.OPEN.SAP								x	x		x	x				x							x	x	x				x		
DLC.OPEN.STATION						x		x	x			x										x	x	x	x				x		
DLC.REALLOCATE						x																	x	x	x			x	x		
DLC.RESET						x																	x	x	x	x			x		
DLC.SET.THRESHOLD						x																	x	x	x				x		
DLC.STATISTICS						x																	x	x	x				x		
READ						x																	x	x	x				x		
READ.CANCEL																							x	x	x				x	x	
RECEIVE						x																	x	x	x				x		
RECEIVE.CANCEL						x																	x	x	x				x		
TRANSMIT.DIR.FRAME	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.I.FRAME	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.TEST.CMD	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.UI.FRAME	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.XID.CMD	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.XID.RESP.FINAL	x					x	x			x						x							x	x	x			x	x		
TRANSMIT.XID.RESP.NOT.FINAL	x					x	x			x						x							x	x	x			x	x		
DIR.CLOSE.ADAPTER																							x	x	x	x			x		
DIR.CLOSE.DIRECT																							x	x	x	x			x		
DIR.INITIALIZE																							x				x		x		
DIR.INTERRUPT																							x	x	x				x		
DIR.OPEN.ADAPTER		x	x	x	x																		x						x		
DIR.OPEN.DIRECT									x														x	x	x				x		
DIR.READ.LOG																							x	x	x	x			x		
DIR.SET.EXCEPTION.FLAGS																							x	x	x				x		
DIR.SET.FUNCTIONAL.ADDRESS																							x	x	x	x			x		
DIR.SET.GROUP.ADDRESS																							x	x	x	x			x		
DIR.STATUS																							x	x	x				x		
DIR.TIMER.CANCEL																							x	x	x				x		
DIR.TIMER.CANCEL.GROUP																							x	x	x				x		
DIR.TIMER.SET																							x	x	x				x		
CCB2 Commands	28	30	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	52	53	54	55	56	57	58		
	Return Codes in Hex																														

# CCB Return Codes

CCB2 Commands	Return Codes in Hex															
	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	68	69	FF
BUFFER.FREE	x		x		x	x	x						x	x	x	
BUFFER.GET	x		x		x	x	x						x	x	x	
DLC.CLOSE.SAP	x		x		x	x	x			x			x	x	x	
DLC.CLOSE.STATION	x		x		x	x	x			x			x	x	x	
DLC.CONNECT.STATION	x		x		x	x	x			x			x	x	x	
DLC.FLOW.CONTROL	x		x		x	x	x			x			x	x	x	
DLC.MODIFY	x		x		x	x	x			x			x	x	x	
DLC.OPEN.SAP	x		x		x	x	x			x			x	x	x	
DLC.OPEN.STATION	x		x		x	x	x			x			x	x	x	
DLC.REALLOCATE	x		x		x	x	x			x			x			
DLC.RESET	x		x		x	x	x			x			x	x	x	
DLC.SET.THRESHOLD	x		x		x	x	x						x	x	x	
DLC.STATISTICS	x		x		x	x	x			x			x	x	x	
READ	x				x	x	x			x			x	x	x	
READ.CANCEL	x				x	x	x						x	x	x	
RECEIVE	x		x		x	x	x			x			x	x	x	
RECEIVE.CANCEL	x		x		x	x	x						x	x	x	
TRANSMIT.DIR.FRAME	x		x		x	x	x			x			x	x	x	
TRANSMIT.I.FRAME	x		x		x	x	x			x			x	x	x	
TRANSMIT.TEST.CMD	x		x		x	x	x			x			x	x	x	
TRANSMIT.UI.FRAME	x		x		x	x	x			x			x	x	x	
TRANSMIT.XID.CMD	x		x		x	x	x			x			x	x	x	
TRANSMIT.XID.RESP.FINAL	x		x		x	x	x			x			x	x	x	
TRANSMIT.XID.RESP.NOT.FINAL	x		x		x	x	x			x			x	x	x	
DIR.CLOSE.ADAPTER	x		x		x	x	x						x	x	x	
DIR.CLOSE.DIRECT	x		x	x	x	x	x			x			x	x	x	
DIR.INITIALIZE	x				x	x	x						x		x	
DIR.INTERRUPT	x		x		x	x	x			x			x	x	x	
DIR.OPEN.ADAPTER	x	x			x	x	x	x	x				x		x	
DIR.OPEN.DIRECT	x		x		x	x	x			x	x		x	x	x	
DIR.READ.LOG	x		x	x	x	x	x			x			x	x	x	
DIR.SET.EXCEPTION.FLAGS	x		x		x	x	x						x	x	x	
DIR.SET.FUNCTIONAL.ADDRESS	x		x		x	x	x			x			x	x	x	
DIR.SET.GROUP.ADDRESS	x		x		x	x	x			x			x	x	x	
DIR.STATUS	x		x		x	x	x						x	x	x	
DIR.TIMER.CANCEL	x		x		x	x	x						x	x	x	
DIR.TIMER.CANCEL.GROUP	x		x		x	x	x						x	x	x	
DIR.TIMER.SET	x		x		x	x	x			x			x	x	x	
CCB2 Commands	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	68	69	FF
	Return Codes in Hex															

Return Codes for CCB3 Commands

CCB3 Commands	Return Codes in Hex																												
	00	01	02	03	05	06	07	08	0A	0B	0C	11	13	15	16	18	19	1A	1B	1C	1D	20	21	22	23	24	25	27	28
BUFFER.FREE	x	x					x													x	x	x							
BUFFER.GET	x	x			x		x										x			x		x							
DLC.CLOSE.SAP	x	x					x			x																			
DLC.CLOSE.STATION	x	x	x				x			x																			
DLC.CONNECT.STATION	x	x	x		x		x			x										x	x	x							
DLC.FLOW.CONTROL	x	x					x			x																			
DLC.MODIFY	x	x			x		x	x		x										x	x	x							
DLC.OPEN.SAP	x	x			x	x	x	x		x					x	x				x	x	x							
DLC.OPEN.STATION	x	x			x		x	x		x										x	x	x							
DLC.REALLOCATE	x	x			x		x			x										x	x	x							
DLC.RESET	x	x					x			x																			
DLC.STATISTICS	x	x			x		x			x				x						x	x	x							
PURGE.RESOURCE	x	x					x																						
RECEIVE	x	x	x		x	x	x			x	x									x	x		x	x	x				
RECEIVE.CANCEL	x	x					x																						
RECEIVE.MODIFY	x	x	x		x		x			x	x									x	x		x	x	x				
TRANSMIT.DIR.FRAME	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.I.FRAME	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.TEST.CMD	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.UI.FRAME	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.XID.CMD	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.XID.RESP.FINAL	x	x			x	x	x	x		x										x	x	x				x	x	x	x
TRANSMIT.XID.RESP.NOT.FINAL	x	x			x	x	x	x		x										x	x	x				x	x	x	x
DIR.CLOSE.ADAPTER	x	x					x																						
DIR.CLOSE.DIRECT	x	x					x			x																			
DIR.INITIALIZE	x	x			x		x													x									
DIR.INTERRUPT	x	x					x			x																			
DIR.OPEN.ADAPTER	x	x			x	x	x													x	x	x							
DIR.OPEN.DIRECT	x	x			x	x	x	x		x					x	x				x	x	x							
DIR.READ.LOG	x	x			x		x			x			x	x						x	x	x							
DIR.SET.EXCEPTION.FLAGS	x	x			x		x													x									
DIR.SET.FUNCTIONAL.ADDRESS	x	x					x			x																			
DIR.SET.GROUP.ADDRESS	x	x					x			x																			
DIR.STATUS	x	x			x		x				x									x									
DIR.TIMER.CANCEL	x	x					x					x																	
DIR.TIMER.CANCEL.GROUP	x	x					x					x																	
DIR.TIMER.SET	x	x					x			x	x		x																
CCB3 Commands	00	01	02	03	05	06	07	08	0A	0B	0C	11	13	15	16	18	19	1A	1B	1C	1D	20	21	22	23	24	25	27	28
	Return Codes in Hex																												

# CCB Return Codes

CCB3 Commands	Return Codes in Hex																													
	30	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	52	53	54	55	57	58	59	5A	5C	
BUFFER.FREE					x																x	x		x			x			
BUFFER.GET					x																	x	x					x		
DLC.CLOSE.SAP					x						x	x					x					x	x					x		
DLC.CLOSE.STATION					x											x	x					x	x					x		
DLC.CONNECT.STATION					x	x			x							x					x							x		
DLC.FLOW.CONTROL					x																							x		
DLC.MODIFY					x		x				x									x								x		
DLC.OPEN.SAP							x	x		x	x																	x		
DLC.OPEN.STATION					x		x	x			x																	x		
DLC.REALLOCATE					x																							x		
DLC.RESET					x																								x	
DLC.STATISTICS					x																								x	
PURGE.RESOURCE																													x	
RECEIVE					x																								x	
RECEIVE.CANCEL					x																								x	
RECEIVE.MODIFY					x																								x	
TRANSMIT.DIR.FRAME					x	x			x							x													x	
TRANSMIT.I.FRAME					x	x			x							x													x	
TRANSMIT.TEST.CMD					x	x			x							x													x	
TRANSMIT.UI.FRAME					x	x			x							x													x	
TRANSMIT.XID.CMD					x	x			x							x													x	
TRANSMIT.XID.RESP.FINAL					x	x			x							x													x	
TRANSMIT.XID.RESP.NOT.FINAL					x	x			x							x													x	
DIR.CLOSE.ADAPTER																													x	
DIR.CLOSE.DIRECT																													x	x
DIR.INITIALIZE																													x	
DIR.INTERRUPT																													x	
DIR.OPEN.ADAPTER	x	x	x	x																									x	x
DIR.OPEN.DIRECT										x																			x	
DIR.READ.LOG																													x	x
DIR.SET.EXCEPTION.FLAGS																													x	
DIR.SET.FUNCTIONAL.ADDRESS																													x	
DIR.SET.GROUP.ADDRESS																													x	
DIR.STATUS																													x	
DIR.TIMER.CANCEL																													x	
DIR.TIMER.CANCEL.GROUP																													x	
DIR.TIMER.SET																													x	
CCB3 Commands	30	32	33	34	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	52	53	54	55	57	58	59	5A	5C	
	Return Codes in Hex																													

CCB3 Commands	Return Codes in Hex															
	5E	60	61	62	63	64	65	67	68	FF						
BUFFER.FREE	x								x							
BUFFER.GET	x								x							
DLC.CLOSE.SAP	x			x				x	x							
DLC.CLOSE.STATION	x			x				x	x							
DLC.CONNECT.STATION	x			x				x	x							
DLC.FLOW.CONTROL	x			x				x	x							
DLC.MODIFY	x			x				x	x							
DLC.OPEN.SAP	x			x				x	x							
DLC.OPEN.STATION	x			x				x	x							
DLC.REALLOCATE	x			x				x	x							
DLC.RESET	x			x				x	x							
DLC.STATISTICS	x			x				x	x							
PURGE.RESOURCES	x								x							
RECEIVE	x			x				x	x							
RECEIVE.CANCEL	x								x							
RECEIVE.MODIFY	x			x				x	x	x						
TRANSMIT.DIR.FRAME	x			x				x	x							
TRANSMIT.I.FRAME	x			x				x	x							
TRANSMIT.TEST.CMD	x			x				x	x							
TRANSMIT.UI.FRAME	x			x				x	x							
TRANSMIT.XID.CMD	x			x				x	x							
TRANSMIT.XID.RESP.FINAL	x			x				x	x							
TRANSMIT.XID.RESP.NOT.FINAL	x			x				x	x							
DIR.CLOSE.ADAPTER	x								x							
DIR.CLOSE.DIRECT	x			x				x	x							
DIR.INITIALIZE	x															
DIR.INTERRUPT	x			x				x	x							
DIR.OPEN.ADAPTER	x	x	x				x									
DIR.OPEN.DIRECT	x			x	x			x	x							
DIR.READ.LOG	x			x				x	x							
DIR.SET.EXCEPTION.FLAGS	x								x							
DIR.SET.FUNCTIONAL.ADDRESS	x			x				x	x							
DIR.SET.GROUP.ADDRESS	x			x				x	x							
DIR.STATUS	x								x							
DIR.TIMER.CANCEL	x								x							
DIR.TIMER.CANCEL.GROUP	x								x							
DIR.TIMER.SET	x			x				x	x							
CCB3 Commands	5E	60	61	62	63	64	65	67	68	FF						
	Return Codes in Hex															



### CCB Return Codes Cause and Action

---

#### Hex 00

**Explanation:** Operation completed successfully.

---

#### Hex 01

**Explanation:** Invalid command code.

**Cause:** CCB\_COMMAND did not contain a recognized command code.

**Action:** Try again, using a valid command.

---

#### Hex 02

**Explanation:** Duplicate command—one already outstanding.

**Cause:** Only one command of this type may be outstanding at one time.

**Action:** Wait for the previously issued command to complete.

---

#### Hex 03

**Explanation:** Adapter open—should be closed.

**Cause:** This command may only be issued when the adapter is not open.

**Action:** Close the adapter or issue the correct command.

**Note:** For an exception to the above, see the DIR.OPEN.ADAPTER command on page 3-34.

---

#### Hex 04

**Explanation:** Adapter closed—should be open.

**Cause:** This command may only be issued when the adapter is open.

**Action:** Open the adapter.

---

#### Hex 05

**Explanation:** Required parameter(s) not provided.

**Cause:** At least one required parameter for which no default is available is coded as zero.

**Action:** Correct the value and try again.

---

#### Hex 06

**Explanation:** Option(s) invalid, or incompatible.

**Cause:** The options selected are not a valid combination. For example, this return code can occur if an attempt is made to open a SAP if that SAP has an XID command handling option different from that of the GSAP it is associated with. In that case the command will have completed up to the point where the failing item in the GSAP list was encountered. Otherwise, no action will have been taken for the command.

**Action:** Correct the options and try again or issue a DLC modify for the remaining GSAP list members.

---

**Hex 07**

**Explanation:** Command canceled—unrecoverable failure.

**Cause:** The adapter has been closed because of an error condition.

**Action:** Analyze the error indications. If the error is not permanent, issue DIR.INITIALIZE (for DOS) and DIR.OPEN.ADAPTER (for DOS and OS/2).

---

**Hex 08**

**Explanation:** Unauthorized access priority.

**Cause:** The requested access priority has not been authorized.

**Action:** Lower the priority specified with either the OPTIONS\_PRIORITY or the ACCESS\_PRIORITY value and reissue the command.

---

**Hex 09**

**Explanation:** Adapter not initialized—should be initialized.

**Cause:** This command may be completed only if the adapter is initialized.

**Action:** Issue the DIR.INITIALIZE command.

---

**Hex 0A**

**Explanation:** Command canceled by user request.

**Cause:** This is the expected response when a command is canceled by an application program command.

**Action:** None.

---

**Hex 0B**

**Explanation:** Command canceled—adapter closed while command in progress.

**Cause:** A DIR.CLOSE.ADAPTER command was issued while this command was in process.

**Action:** As appropriate for the application program.

---

**Hex 0C**

**Explanation:** Command completed successfully—adapter not open.

**Cause:** Information only. The command may execute even though the adapter is not open.

**Action:** None.

---

**Hex 10**

**Explanation:** Adapter open—NETBIOS not operational.

**Cause:** One of the following:

- The DIR.OPEN.ADAPTER command has been passed NETBIOS parameters and NETBIOS code is not loaded.
- One or more NETBIOS parameters in the DIR.OPEN.ADAPTER command was incorrect.

## CCB Return Codes

**Note:** This can occur if the NCB\_MAX\_NAMES or NCB\_MAX\_SESSIONS values are not less than 255, or if there is insufficient work space available to satisfy the values of NCB\_STATIONS, NCB\_MAX\_NAMES, NCB\_MAX, and NCB\_MAX\_SESSIONS.

**Action:**

- To continue without NETBIOS, do nothing. The adapter is open.
  - To use NETBIOS, close the adapter, make appropriate changes, and reissue DIR.OPEN.ADAPTER.
- 

### Hex 11

**Explanation:** DIR.TIMER.SET or DIR.TIMER.CANCEL error.

DIR.TIMER.SET:

**Cause:** The TIMER\_VALUE is not in the 0 to 13107 range.

**Action:** Set a valid value and try again.

DIR.TIMER.CANCEL:

**Cause:** The DIR.TIMER.SET command to be canceled was not found.

**Action:** None

---

### Hex 12

**Explanation:** Available work area exceeded.

**Cause:** Requested parameters exceeded allotted memory. Either the adapter support software work area or the work area provided by the application program is not adequate.

**Action:** Reduce MAX.STATION and/or MAX.SAP values or increase memory to the value returned in parameter DLC.WORK.LEN.ACT.

---

### Hex 13

**Explanation:** Invalid LOG.ID.

**Cause:** The requested LOG.ID is not defined.

**Action:** Adjust the value accordingly.

---

### Hex 14

**Explanation:** Invalid shared RAM segment or size.

**Cause:** The value is not an allowable value.

**Action:** Adjust the value accordingly.

---

### Hex 15

**Explanation:** Lost log data, inadequate buffer space—log reset.

**Cause:** The buffer pointed to by DIR.READ.LOG or DLC.STATISTICS was too short to continue the entire log contents. The information that could not be placed in the buffer was *lost* if the command indicated "reset."

**Action:** The next time the command is issued, increase the size of the buffer.

---

---

**Hex 16**

**Explanation:** Requested buffer size exceeds pool length.

**Cause:** The buffer pool is not large enough to hold one buffer.

**Action:** Issue the command with either smaller buffers or larger pool.

---

**Hex 17**

**Explanation:** Command invalid—NETBIOS operational.

**Cause:** The command being issued would cause a change to NETBIOS parameters that are currently operational.

**Action:** To issue the command, the adapter must be closed and then re-opened, either without NETBIOS, or with NETBIOS parameters that avoid the conflict.

---

**Hex 18**

**Explanation:** Invalid SAP buffer length.

**Cause:** The specified buffer size must be at least 80 bytes and a multiple of 16.

**Action:** Specify the buffer size accordingly.

---

**Hex 19**

**Explanation:** Inadequate buffers available for request.

**Cause:** A request was made for more buffers than were available.

**Action:** Either issue the command requesting fewer buffers, or wait until more buffers become available and try again.

---

**Hex 1A**

**Explanation:** USER LENGTH too large for buffer length.

**Cause:** The user requested area is too large.

**Action:** Reduce the user space specified by the USER\_LENGTH field.

---

**Hex 1B**

**Explanation:** The CCB\_PARM\_TAB pointer is invalid.

**Cause:** The CCB\_PARM\_TAB field value is either pointing into the PC system interrupt vector area or the offset is too near the end of the segment and wrap-around will occur on some of the fields.

**Action:** Re-issue the command with the CCB\_PARM\_TAB field corrected.

---

**Hex 1C**

**Explanation:** A pointer in the CCB Parm table is invalid.

**Cause:** A pointer value in the CCBs parameter table is either pointing into the PC system interrupt vector area or the offset is too near the end of the segment and wrap-around will occur on some of the fields.

**Action:** Re-issue the command with the CCB\_PARM\_TAB field corrected.

---

## CCB Return Codes

---

### Hex 1D

**Explanation:** Invalid CCB\_ADAPTER value.

**Cause:** The value is outside the prescribed range.

**Action:** Set an acceptable value.

---

### Hex 20

**Explanation:** Lost data on receive, no buffers available.

**Cause:** There were no available buffers in the SAP's buffer pool. The frame was lost. This return code will not occur if the frame was an I frame.

**Action:** Free some buffers via BUFFER.FREE, then reissue the receive command. The frame was lost if it was for a link station.

---

### Hex 21

**Explanation:** Lost data on receive—inadequate buffer space.

**Cause:** There was inadequate buffer space in the SAP's buffer pool to contain the entire frame. As much of the frame as possible was placed into receive buffers. The remainder of the message was lost. This return code will not occur if the frame was an I frame.

**Action:** Free some buffers via BUFFER.FREE and reissue the receive command.

---

### Hex 22

**Explanation:** Error on frame transmission—check TRANSMIT\_FS data.

**Cause:** The frame may or may not have been received by the destination adapter, as indicated by the FS byte.

**Action:** As appropriate for the application program.

---

### Hex 23

**Explanation:** Error in frame transmit or read back checking

**Cause:** An error was detected either during the frame transmission or when the frame was read back and checked.

**Action:** As appropriate for the application program.

---

### Hex 24

**Explanation:** Unauthorized MAC frame.

**Cause:** Possible causes:

- The adapter is not authorized to send a MAC frame with the specified source class.
- The source class was zero.
- An attempt has been made to transmit a MAC frame via a SAP.
- An attempt has been made to transmit a MAC frame on the PC Network.

**Action:** Adjust the value and try again.

---

---

**Hex 25**

**Explanation:** Maximum commands exceeded.

**Cause:** The maximum number of transmit commands that may be outstanding for a given station at any time (128) has been exceeded.

**Action:** Issue the transmit command at some later time.

---

**Hex 26**

**Explanation:** Unrecognized command correlator.

**Cause:** The command correlation sent to the adapter during ASB communications is invalid.

**Action:** The application program will never see this return code, since the adapter support software will assume a PC Hard Error state.

---

**Hex 27**

**Explanation:** Link not transmitting I frames—state changed from link opened.

**Cause:** This return code will be set in a transmit CCB whenever the link station leaves link-opened state because of a received frame (for instance, DISC), or because of a timeout. It will *not* be set if the link leaves link-opened state because of receipt of a CCB (for instance, DLC.CLOSE.STATION).

**Action:** The LINK STATION may be closed via DLC.CLOSE.STATION, or an attempt may be made to re-establish the connection via DLC.CONNECT.STATION. If the remote station is on a different ring, a different route may be required in order to re-establish the link.

---

**Hex 28**

**Explanation:** Invalid transmit frame length.

**Cause:** The frame length, as specified, is either too short to contain sufficient header information, or too long for the adapter's transmit buffer. If the transmit was for a link station, it has entered the disconnected state.

**Action:** Transmit frames must be no longer than the maximum transmit length, as defined by DIR.OPEN.ADAPTER.

---

**Hex 30**

**Explanation:** Inadequate receive buffers for adapter to open.

**Cause:** The requested DIR.OPEN.ADAPTER parameters have not allowed adequate receive buffer space in the adapter's shared RAM.

**Action:** Reduce the RAM requirements by reconfiguring with either the configuration aid (OS/2) or the DIR.OPEN.ADAPTER command (DOS). Resources that may be reduced first to free up larger amounts of Shared RAM space are Data Hold buffers (if more than one is specified), and the number of queue elements. In addition, if the number of receive buffers can be reduced without affecting the expected performance level, the number of receive buffers may also be reduced.

## CCB Return Codes

---

### Hex 32

**Explanation:** Invalid NODE\_ADDRESS.

**Cause:** The defined node address is invalid.

**Action:** Adjust the value accordingly. Refer to the *IBM Token-Ring Network Architecture Reference* for node address restrictions.

---

### Hex 33

**Explanation:** Invalid adapter receive buffer length defined.

**Cause:** The value is either greater than the allowable maximum, less than the allowable minimum, or not a multiple of 8.

**Action:** Adjust the value accordingly.

---

### Hex 34

**Explanation:** Invalid adapter transmit buffer length defined.

**Cause:** The value is either greater than the allowable maximum, less than the allowable minimum, or not a multiple of 8.

**Action:** Adjust the value accordingly.

---

### Hex 40

**Explanation:** Invalid STATION\_ID.

**Cause:** Either the requested station ID does not exist or the command code is invalid for the station type.

**Action:** Make the appropriate changes and reissue the command.

---

### Hex 41

**Explanation:** Protocol error—link in invalid state for command.

**Cause:** The requested command cannot be accepted because of the existing primary link state of the link station. A DLC.CONNECT.STATION command will not be accepted if the link is in the disconnected or closed state. A transmit command will not be accepted if the link is in any state other than opened.

**Action:** According to the situation.

---

### Hex 42

**Explanation:** Parameter exceeded maximum allowed.

**Cause:** One of the parameter values is greater than acceptable.

**Action:** Use an acceptable value.

---

---

**Hex 43**

**Explanation:** Invalid SAP\_VALUE or value already in use.

**Cause:** For a DLC.OPEN.SAP command, this return code indicates that the SAP\_VALUE has already been used or the specified SAP is the Null or Global SAP.

For a DLC.OPEN.STATION command, this return code indicates that this SAP already has a link to the specified RSAP\_VALUE and DESTINATION\_ADDR combination, or that the remote SAP specified was the Null SAP, Global SAP, or a group SAP.

**Action:** Use an acceptable value. (Do not use X'00'.)

---

**Hex 44**

**Explanation:** Invalid routing field length.

**Cause:** The indicated routing field is either too short, greater than 18 bytes long, or is an odd number of bytes long.

**Action:** Set the length field to a correct value.

---

**Hex 45**

**Explanation:** Requested group membership in non-existent group SAP.

**Cause:** Membership has been requested in a group SAP that is not open.

**Note:** The command has been completed up to the point at which the adapter encountered the error. Other parameters have been changed if the command was DLC.MODIFY. However, the SAP has not been added as a member to a group SAP.

**Action:** Either change the group SAP value to a group SAP that has been opened or open the group SAP prior to requesting its membership.

---

**Hex 46**

**Explanation:** Inadequate link stations.

**Cause:**

- DLC.OPEN.SAP: There are inadequate link stations or SAPs available to satisfy the open.
- DLC.OPEN.STATION: All link stations assigned to this SAP are in use.

**Action:**

- DLC.OPEN.SAP: Close other SAPs, reduce the number of link stations being requested for the SAP, or wait for these resources to be freed.
- DLC.OPEN.STATION: Close other link stations for the SAP, reallocate some link stations using the DLC.REALLOCATE command, close the SAP and reserve additional link stations, or wait for these resources to be freed.



## CCB Return Codes

---

### Hex 47

**Explanation:** SAP cannot close unless all link stations are closed.

**Cause:** At least one link station is open for this SAP.

**Action:** Close all link stations and try again.

**Note:** If a 47 error code results when a DLC.CLOSE.SAP command closely follows a DLC.CLOSE.STATION command for the last open station for that SAP, reissue the DLC.CLOSE.SAP command.

---

### Hex 48

**Explanation:** Group SAP cannot close—all member SAPs not closed

**Cause:** At least one individual member SAP of this group SAP is open.

**Action:** Delete all SAPs in the group using the DLC.MODIFY command and try again.

---

### Hex 49

**Explanation:** Group SAP has reached maximum membership

**Cause:** As stated.

**Note:** The command has completed up to the point at which the adapter encountered the error. Other parameters have been changed if the command was DLC.MODIFY.

**Action:** According to the application program.

---

### Hex 4A

**Explanation:** Sequence error—incompatible command in progress.

**Cause:** The station is in the process of closing or establishing a connection.

**Action:** Await completion or issue a DLC.RESET command.

---

### Hex 4B

**Explanation:** Station closed without remote acknowledgment.

**Cause:** The adapter issued a DISC command to the remote station as a result of receiving a DLC.CLOSE.STATION SRB. No acknowledgment has been received from the remote adapter and the link station has been closed.

**Action:** According to the application program.

---

### Hex 4C

**Explanation:** Sequence error—cannot close while commands are outstanding.

**Cause:** Commands are in process. This prevents closing the SAP or link station.

**Action:** Wait until all outstanding commands are complete, or issue a reset.

---

---

**Hex 4D**

**Explanation:** Unsuccessful link station connection attempt.

**Cause:** The DLC.CONNECT.STATION command could not establish a requested connection.

**Action:** Determine the cause for the failure (for example, verify RSAP values, routing information, MAC address, and connection between the two work stations) and try again when resolved.

---

**Hex 4E**

**Explanation:** Member SAP not found in group SAP list.

**Note:** The command has completed up to the point at which the adapter encountered the error. Other parameters have been changed if the command was DLC.MODIFY.

**Cause:** A request was issued to delete an individual member SAP from a group SAP. The SAP was not found to be assigned to the group.

**Action:** Verify the SAP value.

---

**Hex 4F**

**Explanation:** Invalid remote address—may not be a group address.

**Cause:** The remote address parameter has the high bit of the high byte set to 1, which indicates a group address. A group address is not allowed to be specified for this command.

**Action:** Correct the remote address and reissue the command.

---

**Hex 50**

**Explanation:** Invalid pointer in the CCB\_POINTER field.

**Cause:** As stated.

**Action:** Check to ensure that all 4 byte pointers in the CCB's pointer field are accessible to the Operating System/2 process issuing the command.

---

**Hex 52**

**Explanation:** Invalid application program ID.

**Cause:** As stated.

**Action:** Use the CCB\_APPL\_ID field returned on the DIR.OPEN.ADAPTER command.

---

**Hex 53**

**Explanation:** Invalid application program key code.

**Cause:** As stated.

**Action:** Use the CCB\_APPL\_KEY field provided on DIR.OPEN.ADAPTER command.

---

**Hex 54**

**Explanation:** Invalid System Key code.

**Cause:** As stated.

**Action:** Use the System Key code as defined by the configuration parameters.

---

## CCB Return Codes

---

### Hex 55

**Explanation:** Buffer is smaller than buffer size given on the DIR.OPEN.SAP command.

**Cause:** As Stated.

**Action:** Increase the size of the buffer being used to at least that of the buffers in the SAP buffer pool.

---

### Hex 56

**Explanation:** The adapter's system process is not installed.

**Cause:** As stated.

**Action:** Load the adapter support software's system process. Insert the DETACH command into STARTUP.BAT file specifying the system process load module name.

---

### Hex 57

**Explanation:** Inadequate stations are available.

**Cause:** Command has completed, however only a portion of the stations requested have been reserved. No more stations are available for reservation.

**Action:** None

---

### Hex 58

**Explanation:** Invalid CCB\_PARAMETER\_1.

**Cause:** As stated.

**Action:** Adjust the parameter defined in the CCB\_PARAMETER\_1 field and reissue the command.

---

### Hex 59

**Explanation:** Inadequate queue elements to satisfy request.

**Cause:** As stated open.

**Action:** On a short term basis wait for other requests to complete. Otherwise, increase queue elements using the configuration aid.

---

### Hex 5A

**Explanation:** Initialization failure, cannot open the adapter.

**Cause:** As stated.

**Action:** Check Bring-up error code for details of initialization failure.

---

### Hex 5B

**Explanation:** Error detected in chained READ command.

**Cause:** Bad return code given on a chained READ command.

**Action:** Correct READ command problem and reissue the command.

---

---

**Hex 5C**

**Explanation:** Direct stations are not assigned to this application program.

**Cause:** The invoking application program must request ownership of direct stations before issuing any requests involving the direct stations.

**Action:** Issue a DIR.OPEN.DIRECT command to gain ownership of the direct stations.

---

**Hex 5D**

**Explanation:** The Device Driver Interface is not installed.

**Cause:** As stated.

**Action:** Insert the DEVICE command into CONFIG.SYS file specifying the device driver load module name.

---

**Hex 5E**

**Explanation:** The requested adapter is not installed.

**Cause:** As stated.

**Action:** Install the adapter into the system. Check that the adapter is configured correctly for the primary or alternate adapter.

---

**Hex 5F**

**Explanation:** Chained commands must all be for the same adapter.

**Cause:** As stated.

**Action:** Only chain multiple command requests that are for the same adapter. Do not mix primary and alternate adapter requests.

---

**Hex 60**

**Explanation:** Adapter initializing, command not accepted.

**Cause:** As stated.

**Action:** Reissue the DIR.OPEN.ADAPTER command until command completes successfully.

---

**Hex 61**

**Explanation:** The number of allowed application programs has been exceeded.

**Cause:** As stated.

**Action:** On a short term basis, terminate one of the applications using the adapter support software. Otherwise, adjust the configuration parameter: MAX\_USERS.

---

**Hex 62**

**Explanation:** Command cancelled, system action.

**Cause:** The system administrator has issued a command using the System Key.

**Action:** Re-establish session with the adapter support software if necessary.

---

## CCB Return Codes

---

### Hex 63

**Explanation:** Direct stations are not available.

**Cause:** Direct stations have already been assigned.

**Action:** The application program owning the direct stations must issue the DIR.OPEN.DIRECT command to relinquish ownership of direct stations in order for the command to complete successfully.

---

### Hex 64

**Explanation:** Invalid DDNAME parameter.

**Cause:** Either the device driver name given was invalid or the device driver did not provide the proper inter-device driver communication information in its device driver header.

**Action:** Make sure the device driver is loaded and has a valid header.

---

### Hex 65

**Explanation:** Inadequate GDT selectors to satisfy request.

**Cause:** The number of GDT selectors defined during configuration is not adequate to support the number of control blocks and buffers passed to the adapter support software.

**Action:** Increase the number of GDT selectors when configuring the adapter support software using the configuration aid.

---

### Hex 67

**Explanation:** Command cancelled, CCB resources purged

**Cause:** The command has been cancelled as a result of the PURGE.RESOURCE command.

**Action:** None

---

### Hex 68

**Explanation:** The application program ID is not valid.

**Cause:** The application program ID provided was not obtained from the interface used for the command request.

**Action:** The application program ID can only be used at the interface where it was obtained, use the other interface (For example, Dynamic Link Routine Interface or Device Driver Interface).

---

### Hex 69

**Explanation:** The segment associated with the request cannot be locked.

**Cause:** Too many processes are running concurrently and the system has run out of the resources. The segment cannot be locked.

**Action:** Reduce the number of Operating System/2 processes (the overall number of different segments controlled by the adapter support software), or wait until the resources are available. Memory references that are passed to the application program that contain control blocks that will be accessed when the adapter support software processes an adapter interrupt (for example, CCBs and buffers in the SAP buffer pool) are locked so they will not be moved or swapped by OS/2. By canceling processes using the application program or reducing the amount of control blocks passed to the application program, more memory will be available for other requests that require their control blocks locked. Adding memory to your system may also alleviate this problem.

---

**Hex FF**

**Explanation:** Command in process.

**Cause:** As stated.

**Action:** None.

---

## DLC Status Codes

**For CCB1:** Certain conditions that arise in the DLC function of the adapter (for Token-Ring Network) or the DLC function of the adapter support software in the PC system (for PC Network) are reported to a DLC Status Appendage. DLC status codes are presented to the appendage in the AX register. The CX register contains the adapter number (0 or 1). Register SI contains a user value as defined in the USER\_STAT\_VALUE parameter of the DLC.OPEN.SAP command. Registers ES and BX point to the DLC status table that contains additional data for certain status codes. See "Appendages" on page 2-4 for more about providing an appendage to use these codes.

**For CCB2:** Certain conditions that occur in the DLC function are returned in the READ command for DLC status change events. The format of the DLC Status table in the READ command's CCB is shown below. This table starts at offset 10 in the READ command's parameter table.

**For CCB3:** Certain conditions that arise in the DLC function of the adapter (for Token-Ring Network) or the DLC function of the adapter support software in the PC system (for PC Network) are reported to a DLC Status Appendage. DLC status codes are presented to the appendage in the AX register. The CX register contains the adapter number (0 or 1). Register SI contains a user value as defined in the USER\_STAT\_VALUE parameter of the DLC.OPEN.SAP command. Registers ES and BX point to the DLC status table that contains additional data for certain status codes. See "Appendages" on page 2-4 for more about providing an appendage to use these codes.

The DI register contains the offset of the DLC appendage. The DS register contains the application program device driver protect mode data segment. An invocation code of X'0001' has been pushed onto the stack. Before returning the the adapter support software, the application program must remove the invocation code from the stack.

## DLC Status Table

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	STATION_ID	2	DW	STATION.ID
2	DLC_STATUS_CODE	2	DW	DLC status Code
4	FRMR_DATA	5	DB	5 bytes of reason code that are applicable when an FRMR is either transmitted or received.
9	ACCESS_PRIORITY	1	DB	The new access priority that is applicable when status bit 5 is on. The format is: B'nnn00000' where "nnn" is the access priority. This byte is not used and set to zero when used on the PC Network.
10	REMOTE_NODE	6	DB	The 6-byte remote node value of a newly opened link station. Applicable when status bit 10 is on.
16	REMOTE_SAP	1	DB	The 1-byte remote SAP value of a newly opened link station. Applicable when status bit 10 is on.
<b>For CCB2 and CCB3:</b>				
17		1	DB	Reserved
18	USER_STAT_VALUE	2	DW	User value as defined in the DLC.OPEN.SAP command

## DLC Status Codes

BIT	FUNCTION	ADDITIONAL DATA
15	Link lost	
14	DM or DISC received, or DISC acknowledged	
13	FRMR received	Five bytes of reason code data are contained in the area pointed to by the address in the ES:BX register (FRMR_DATA).
12	FRMR sent	Five bytes of reason code data are contained in the area pointed to by the address in the ES:BX register (FRMR_DATA).
11	SABME received for an open link station	
10	SABME received, link station opened	A new link station has been opened by the adapter. A DLC.OPEN.STATION command should NOT be issued for that link station. A DLC.CONNECT.STATION command must be issued to accept the SABME or a DLC.CLOSE.STATION command to reject it. The local STATION_ID, remote node address (DESTINATION_ADDR), and remote SAP value (RSAP_VALUE) are provided in the status table pointed to by the address in the ES and BX registers.
9	Remote station has entered a "local busy" condition	
8	Remote station has left a "local busy" condition	
7	Ti has expired	
6	DLC counter overflow	One or more of the DLC LINK STATION's DLC log counters has reached one half the maximum value. A DLC.STATISTICS command should be issued.
5	Access Priority lowered	The new access priority (ACCESS_PRIORITY) is in the area pointed to by the address in the ES:BX register. This bit is not set on the PC Network.
4-1	Reserved	
0	Local station has entered a "local busy" condition	This code is reported only when the state has changed because of a buffer pool (out-of-buffers) condition when the adapter support software cannot accept an I-frame. It is not reported because of a DLC.FLOW.CONTROL command being issued by the network application program.  It is the responsibility of the application program to issue a "flow control on" command to reset the "local busy" condition.

## Suggested Actions in Response to DLC Status

- Link Lost**  
It appears that the connection to the remote partner has been lost, or that the remote station has been closed. A DLC.CLOSE.STATION command may be issued to free the control block, or a DLC.CONNECT.STATION command (possibly with different routing information) may be issued to attempt to reestablish the connection. Any outstanding transmit commands will be returned with a CCB\_RETCODE of X'27'.
- DM or DISC received**  
The remote partner is attempting to terminate the connection. A DLC.CLOSE.STATION command should be issued. Any outstanding transmit commands will be returned with a CCB\_RETCODE of X'27'.
- FRMR Received**  
The remote partner has detected a DLC protocol error in the frame received from this station. Either a DLC.CLOSE.STATION or DLC.CONNECT.STATION command should be issued. Any outstanding transmit commands will be returned with a CCB\_RETCODE of X'27'.
- FRMR Sent**

The local link station has detected a DLC protocol error in a frame received from the remote partner. However, if a Ti Timer-expired DLC Status interrupt is received after receipt of this interrupt, a DLC.CLOSE.STATION or DLC.CONNECT.STATION command should be issued to the local station. Any outstanding transmit commands will be returned with a CCB\_RETCODE of X'27'.

- **SABME Received for an Open Link Station**  
The remote station wishes to reset an existing connection. A DLC.CONNECT.STATION command may be issued to reestablish the connection, or a DLC.CLOSE.STATION command may be issued to terminate it. Any outstanding transmit commands will be returned with a CCB\_RETCODE of X'27'.
- **SABME Received, Link Station Opened**  
A control block has been allocated and a station has been opened, in disconnected state, in response to a SABME received from a remote station. The connection request may be accepted by issuing a DLC.CONNECT.STATION command, or rejected by issuing a DLC.CLOSE.STATION command.
- **Remote Station Has Entered Local Busy**  
The remote station has temporarily stopped receiving I frames, probably because of buffer congestion. The local station will stop sending I frames. The application program may choose to issue transmit commands for the affected station, up to the maximum number accepted by the adapter, but they will be queued until the remote station leaves the local busy state.
- **Remote Station Has Left Local Busy**  
The local station will resume I-frame transmission.
- **Ti Timer Expired**  
This status is not returned while the link is in link-opened state. In other states it is returned to indicate that there is no activity on the link, and that the PC system may therefore wish to close the link to free up the control block.
- **DLC Counter Overflow**  
One or more of the error counters maintained for the link station has reached half of its maximum value. The counter will wrap back to zero when it reaches its maximum value. The application program should issue a DLC.STATISTICS command to read and reset the counters.
- **Access Priority Reduced**  
The access priority requested for this SAP or link station was greater than that authorized for the adapter and it has been reduced. The new priority is in the Adapter Status Table or if the adapter is being operated without the adapter support software, in ARB byte 13. There is no PC system application program action required as this is for information only. However, a DLC.MODIFY command may be issued to change the access priority. Access priority is not set on the PC Network.

---

## NCB Return Codes

Code	Meaning
00	Good return
01	Illegal buffer length
03	Invalid command
05	Command timed out
06	Message incomplete
07	Data for one or more SEND type NO.ACK commands was not received
08	Illegal local session number
09	No resource available
0A	Session closed
0B	Command canceled



## NCB Return Codes

Code	Meaning
0D	Duplicate name in local name table
0E	Name table full
0F	Command completed—name has active session and is now de-registered
11	Local session table full
12	Session open rejected
13	Illegal name number
14	Cannot find name called
15	Name not found or cannot specify "*" or null
16	Name in use on remote NETBIOS
17	Name deleted
18	Session ended abnormally
19	Name conflict detected
21	Interface busy
22	Too many commands outstanding
23	Invalid number in NCB_LANA_NUM field
24	Command completed while cancel occurring
26	Command not valid to cancel
30	Name defined by another environment
34	Environment not defined, RESET must be issued
35	Required operating system resources exhausted, retry later.
36	Maximum applications exceeded
37	No SAPs available for NETBIOS.
38	Requested resource(s) not available.
39	Invalid NCB address or length does not fit in segment.
3A	RESET may not be issued from a NETBIOS adapter appendage.
3B	Invalid NCB_DD_ID value.
3C	NETBIOS attempted to lock user storage and the lock failed.
3F	NETBIOS Device Driver open error.
40	OS/2 error detected.
4E	Network status—one or more of bits 12, 14, 15 on for longer than 60 seconds
4F	Network status—one or more of bits 8-11 on
F6	Unexpected error on CCB completion
F7	Error on implicit DIR.INITIALIZE
F8	Error on implicit DIR.OPEN.ADAPTER
F9	adapter support software internal error
FA	Adapter check
FB	NETBIOS program not loaded in PC
FC	DIR.OPEN.ADAPTER or DLC.OPEN.SAP failed, check parameters
FD	Unexpected adapter close
FE	NETBIOS not operational and application program explicitly opened the adapter.

### NCB Return Codes Listed by Command

NCB Commands	Return Codes in Hex																													
	00	01	03	05	06	07	08	09	0A	0B	0D	0E	0F	11	12	13	14	15	16	17	18	19	21	22	23	24	26	30		
NCB.ADD.GROUP.NAME	x		x								x	x						x	x			x	x	x	x					
NCB.ADD.NAME	x		x								x	x						x	x			x	x	x	x			x		
NCB.CALL	x		x	x				x		x				x	x		x	x			x	x	x	x	x			x		
NCB.CANCEL	x	x	x																							x	x	x		
NCB.CHAIN.SEND	x	x	x	x			x	x		x	x										x			x	x	x		x		
NCB.CHAIN.SEND.NO.ACK	x	x	x				x	x		x	x										x			x	x	x		x		
NCB.DELETE.NAME	x		x										x					x					x	x	x					
NCB.FIND.NAME	x	x	x	x																			x	x	x	x				
NCB.HANG.UP	x		x	x				x		x	x										x			x	x	x				
NCB.LAN.STATUS.ALERT	x		x							x														x	x	x				
NCB.LISTEN	x		x						x		x			x				x		x	x	x	x	x	x			x		
NCB.RECEIVE	x	x	x	x	x	x	x		x	x											x			x	x	x				
NCB.RECEIVE.ANY	x	x	x			x	x		x	x						x					x	x	x	x	x	x				
NCB.RECEIVE.BROADCAST.DATAGRAM	x	x	x		x					x							x				x		x	x	x	x		x		
NCB.RECEIVE.DATAGRAM	x	x	x		x					x							x				x	x	x	x	x	x		x		
NCB.RESET	x	x	x																					x	x					
NCB.SEND	x	x	x	x			x	x		x	x											x			x	x	x			
NCB.SEND.BROADCAST.DATAGRAM	x	x	x														x							x	x	x	x	x		
NCB.SEND.DATAGRAM	x	x	x														x							x	x	x	x	x		
NCB.SEND.NO.ACK	x	x	x				x	x		x	x											x			x	x	x	x		
NCB.SESSION.STATUS	x	x	x		x														x				x	x	x	x		x		
NCB.STATUS	x	x	x	x	x					x														x	x	x	x		x	
NCB.TRACE	x	x	x									x															x			
NCB.UNLINK	x		x																							x		x		
NCB Commands	00	01	03	05	06	07	08	09	0A	0B	0D	0E	0F	11	12	13	14	15	16	17	18	19	21	22	23	24	26	30		
	Return Codes in Hex																													

## NCB Return Codes

NCB Commands	Return Codes in Hex															
	34	35	36	37	38	39	3A	3B	3C	40	4x	Fx				
NCB.ADD.GROUP.NAME	x	x				x		x	x	x	x	x				
NCB.ADD.NAME	x	x				x		x	x	x	x	x				
NCB.CALL	x	x				x		x	x	x	x	x				
NCB.CANCEL	x	x				x		x	x	x	x	x				
NCB.CHAIN.SEND	x	x				x		x	x	x	x	x				
NCB.CHAIN.SEND.NO.ACK	x	x				x		x	x	x	x	x				
NCB.DELETE.NAME	x	x				x		x	x	x	x	x				
NCB.FIND.NAME	x	x				x		x	x	x	x	x				
NCB.HANG.UP	x	x				x		x	x	x	x	x				
NCB.LAN.STATUS.ALERT	x	x				x		x	x	x	x	x				
NCB.LISTEN	x	x				x		x	x	x	x	x				
NCB.RECEIVE	x	x				x		x	x	x	x	x				
NCB.RECEIVE.ANY	x	x				x		x	x	x	x	x				
NCB.RECEIVE.BROADCAST.DATAGRAM	x	x				x		x	x	x	x	x				
NCB.RECEIVE.DATAGRAM	x	x				x		x	x	x	x	x				
NCB.RESET			x	x	x		x	x	x	x	x	x				
NCB.SEND	x	x				x		x	x	x	x	x				
NCB.SEND.BROADCAST.DATAGRAM	x	x				x		x	x	x	x	x				
NCB.SEND.DATAGRAM	x	x				x		x	x	x	x	x				
NCB.SEND.NO.ACK	x	x				x		x	x	x	x	x				
NCB.SESSION.STATUS	x	x				x		x	x	x	x	x				
NCB.STATUS	x	x				x		x	x	x	x	x				
NCB.TRACE													x			
NCB.UNLINK								x	x		x	x				
NCB Commands	34	35	36	37	38	39	3A	3B	3C	40	4x	Fx				
	Return Codes in Hex															

## NCB Return Codes Cause and Action

### Hex 00

**Explanation:** Operation completed successfully.

### Hex 01

**Explanation:** Illegal buffer length.

**Cause:** The requested buffer length (or invalid buffer selector if using NETBIOS 3.0) is illegal for the NCB.SEND.DATAGRAM, NCB.SEND.BROADCAST, NCB.STATUS, or NCB.SESSION.STATUS command.

**Action:** Specify the correct size for the buffer and retry.

### Hex 03

**Explanation:** Invalid command.

**Cause:** As stated.

**Action:** Issue the correct command.

---

**Hex 05**

**Explanation:** Command timed out.

**Cause:** As stated.

**Action:** Reissue the same command or another command. If a send timed out, there may not be a receive outstanding from the other name.

---

**Hex 06**

**Explanation:** Message incomplete.

**Cause:** The application program received part of a message because the specified buffer length is not large enough to receive the full message.

**Action:**

- NCB.RECEIVE and NCB.RECEIVE.ANY: Issue another receive to get the rest of the message before the remote side times out.
- NCB.STATUS, NCB.SESSION.STATUS, NCB.RECEIVE.DATAGRAM, and NCB.RECEIVE.BROADCAST.DATAGRAM: The remaining data is lost.

**Note:** If the command was a NCB.STATUS, this error code could occur because the remote side could not transmit the entire status update if the data was of greater length than the maximum length UI-frame that may be transmitted.

---

**Hex 07**

**Explanation:** Data for one or more SEND type NO.ACK commands was not received.

**Cause:** Data sent by a previous NCB.SEND.NO.ACK or NCB.CHAIN.SEND.NO.ACK command was either not received at all or only partially received by the remote application program.

**Action:** The application program will need to initiate any data recovery needed.

---

**Hex 08**

**Explanation:** Illegal local session number.

**Cause:** The session number specified is not one of the active sessions.

**Action:** Reissue the command with the correct active session number.

---

**Hex 09**

**Explanation:** No resource available.

**Cause:** Trying to establish a session with a remote application program that has no more room in the session table.

**Action:** Reissue the command at a later time.

---

**Hex 0A**

**Explanation:** Session closed.

**Cause:** The name from the transmitting side closed the session. The session has terminated normally.

**Action:** None.

---

## NCB Return Codes

---

### Hex 0B

**Explanation:** Command canceled.

**Cause:** As stated.

**Action:** None.

---

### Hex 0D

**Explanation:** Duplicate name in local name table.

**Cause:** Tried to specify a name that is currently in the name table.

**Action:** Reissue the command and specify another name.

---

### Hex 0E

**Explanation:** Name table full.

**Cause:** The number of names defined has exceeded the number defined at initialization (default = 17).

**Action:** Wait until a delete name is issued so an entry will become available.

---

### Hex 0F

**Explanation:** Command completed, name has active session and is now de-registered.

**Cause:** The name to be deleted is active in a session now, but is de-registered. When the name is marked de-registered and has active sessions, it still occupies a slot in the table. The name is unusable for any new sessions.

**Action:** Close all the sessions using this name.

---

### Hex 11

**Explanation:** Local session table full.

**Cause:** There are no available entries on the session table. (The number of sessions is user-specified in NCB.RESET).

**Action:** Wait until a session has closed so an entry will become available.

---

### Hex 12

**Explanation:** Session open rejected.

**Cause:** No LISTEN command is outstanding on the remote NETBIOS.

**Action:** Wait until a LISTEN is issued on the remote NETBIOS.

---

### Hex 13

**Explanation:** Illegal name number

**Cause:** The number of the name has been changed or was never specified.

**Action:** Must use the most recent number that was assigned to the name.

---

---

**Hex 14**

**Explanation:** Cannot find name called or no answer.

**Cause:** No response to the NCB.CALL command received.

**Action:** Try again later.

---

**Hex 15**

**Explanation:** Name not found or cannot specify "\*" or null.

**Cause:** The name specified is not in the table, or the first character of the name is either an ASCII asterisk or "00."

**Action:** Try again with another name that has been verified to be correct.

---

**Hex 16**

**Explanation:** Name in use on remote NETBIOS.

**Cause:** Name found in another table. Names used in the network are unique and can only be used in one place. The name is already defined on another node.

**Action:** Either specify another name or have the name changed at the remote end.

---

**Hex 17**

**Explanation:** Name deleted.

**Cause:** As stated.

**Action:** Add the name to the table and reissue the command.

---

**Hex 18**

**Explanation:** Session ended abnormally.

**Cause:** The most probable cause is that a send-type NCB timed-out because no receive command was available in the remote node.

**Action:**

- If a send timed-out, re-establish the session and ensure that the remote node has issued a receive.
  - If the session cannot be re-established, maintenance procedures should be initiated for the node in question.
- 

**Hex 19**

**Explanation:** Name conflict detected.

**Cause:** Network protocol has detected two or more identical names on the network.

**Action:** Identical names on the network should be removed.

## NCB Return Codes

---

### Hex 21

**Explanation:** Interface busy.

**Cause:** NETBIOS is either busy or out of local resources.

**Note:** This condition can also be caused by any of the network status bits 12, 14, or 15 being on.

**Action:** Try again later.

---

### Hex 22

**Explanation:** Too many commands outstanding.

**Cause:** As stated.

**Action:** Try again later.

---

### Hex 23

**Explanation:** Invalid number in NCB\_LANA\_NUM field.

**Cause:** Tried to specify a value other than "00" or "01," or the adapter is not present.

**Action:** Verify that the adapter is present, or correct the value and try the command again. Use "00" for the primary adapter and "01" for the alternate.

---

### Hex 24

**Explanation:** Command completed while cancel occurring.

**Cause:** Tried to cancel a command that had already been completed.

**Action:** None.

---

### Hex 26

**Explanation:** Command not valid to cancel.

**Cause:** Tried to cancel a command that is invalid to cancel.

**Action:** Verify the correctness of the cancel command.

---

### Hex 30

**Explanation:** Name defined by another environment.

**Cause:** Another environment has already defined the name.

**Action:** Choose another name.

---

### Hex 34

**Explanation:** Environment not defined, RESET must be issued.

**Cause:** As stated.

**Action:** Issue RESET.

---

---

**Hex 35**

**Explanation:** Required operating system resources exhausted, retry later.

**Cause:** As stated.

**Action:** Retry command later.

---

**Hex 36**

**Explanation:** Maximum applications exceeded.

**Cause:** The maximum number of applications defined at NETBIOS 3.0 load-time are executing.

**Action:** Wait until another application terminates.

---

**Hex 37**

**Explanation:** No SAPs available for NETBIOS.

**Cause:** The adapter has no SAPs available for NETBIOS.

**Action:** Wait until an application using SAPs relinquishes use of a SAP.

---

**Hex 38**

**Explanation:** Requested resource(s) not available.

**Cause:** As stated.

**Action:** Operate with the smaller number of resources or terminate.

---

**Hex 39**

**Explanation:** Invalid NCB address or length does not fit in segment.

**Cause:** As stated.

**Action:** Application error. Correct NCB address and selector length.

**Note:** In the case of this return code, since the NCB is in doubt, the value is returned only in register AL. No attempt is made to place the return code into the NCB.

---

**Hex 3A**

**Explanation:** RESET may not be issued from a NETBIOS adapter appendage.

**Cause:** The RESET command was issued when the NETBIOS 3.0 adapter was processing a hardware interrupt.

**Action:** Application error. Do not issue RESET in this situation.

**Note:** Return code applies only to the Device Driver (DD) interface.

---

**Hex 3B**

**Explanation:** Invalid NCB\_DD\_ID value.

**Cause:** The value in NCB\_DD\_ID is not identical to the value returned by NETBIOS 3.0 in the first RESET issued by the device driver application program. Note that NCB\_DD\_ID must be X'0000' in the first RESET issued for a given device driver application program.

**Action:** Application error. Correct NCB\_DD\_ID value.

---



## NCB Return Codes

**Note:** Return code applies only to the Device Driver (DD) interface.

---

### Hex 3C

**Explanation:** NETBIOS attempted to lock user storage and the lock failed.

**Cause:** As stated.

**Action:** Try the command at a later time.

---

### Hex 3F

**Explanation:** NETBIOS Device Driver open error.

**Cause:** Either the device driver had an actual problem in its open process or the NETBIOS device driver was not loaded.

**Action:** Load the appropriate code before executing NETBIOS applications.

---

### Hex 40

**Explanation:** OS/2 error detected.

**Cause:** During processing, an unexpected error was indicated by OS/2.

**Action:** Issue RESET and restart. If failure continues, notify IBM.

---

### Hex 4E

**Explanation:** Network status—one or more of bits 12, 14, or 15 are on for more than 60 seconds.

**Cause:** As stated.

**Action:** Check the extended status last network status code. The only NETBIOS command that may be issued is NCB.RESET.

**Note:** This return code is not reported at all if some status bits (8-11) are also on. This return code is reported to the application program only if the status bits 12, 14, or 15 remain on longer than 60 seconds.

---

### Hex 4F

**Explanation:** Network status—one or more of bits 8 - 11 on.

**Cause:** As stated.

**Action:** Check the extended status last network status code. The only NETBIOS command that may be issued is NCB.RESET.

---

### Hex F6

**Explanation:** Unexpected error on CCB completion.

**Cause:** This is a NETBIOS 2.X return code that indicates that a CCB has completed with an unexpected bad return code. NETBIOS 1.X returned a X'FA' in these situations.

**Action:** The only NETBIOS command that may be issued is NCB.RESET.

---

**Hex F7**

**Explanation:** Error on implicit DIR.INITIALIZE.

**Cause:** As stated.

**Action:** Check the extended status bring-up error code. The only NETBIOS command that may be issued is NCB.RESET.

---

**Hex F8**

**Explanation:** Error on implicit DIR.OPEN.ADAPTER.

**Cause:** As stated.

**Action:** Check the extended status bring-up error code. The only NETBIOS command that may be issued is NCB.RESET.

**Notes:**

1. There is a possibility that a DIR.OPEN.ADAPTER could fail because of a temporary timing condition. Because of this, before reporting this return code, the DIR.OPEN.ADAPTER is retried twice at thirty-second intervals.
  2. This error could be caused by an attempt to open on a Token-Ring Network with the adapter set to the wrong data rate. Check the data rate setting.
- 

**Hex F9**

**Explanation:** adapter support software internal error.

**Cause:** As stated.

**Action:** Check the PC System detected error code. The only NETBIOS command that may be issued is NCB.RESET.

---

**Hex FA**

**Explanation:** Adapter check.

**Cause:** As stated.

**Action:** Check the adapter check reason code. The only NETBIOS command that may be issued is NCB.RESET.

---

**Hex FB**

**Explanation:** NETBIOS code not loaded in the PC system.

**Cause:** NETBIOS is not loaded or is loaded and not operational due to an error at load time, but a control block has been received by the adapter support software with a value greater than X'03' in the first field.

**Action:** Load and start NETBIOS or correct the conditions that cause a load error and reissue the command or correct the control block.

## NCB Return Codes

---

### Hex FC

**Explanation:** DIR.OPEN.ADAPTER or DLC.OPEN.SAP failed—check parameters.

**Cause:** As stated.

**Action:** Correct the parameters in error and execute the DIR.OPEN.ADAPTER command again. Note that the DLC.OPEN.SAP command is executed on initial start and restart of NETBIOS. The parameters used are obtained from the DIR.OPEN.ADAPTER command (executed either explicitly or implicitly).

**Note:** There is a possibility that a DIR.OPEN.ADAPTER could fail because of a temporary timing condition. Because of this, before reporting this return code, the DIR.OPEN.ADAPTER is tried again twice at thirty-second intervals.

---

### Hex FD

**Explanation:** Unexpected adapter close.

**Cause:** The adapter was closed while NETBIOS was executing.

**Action:** Issue a NCB.RESET command.

---

### Hex FE

**Explanation:** NETBIOS not operational and application program explicitly opened the adapter.

**Cause:** The adapter has been explicitly opened by the application program and NETBIOS is not operational.

**Action:** Close the adapter and re-issue the NETBIOS command.

### Notes:

For the following codes 'F7' to 'FD':

1. The condition to be reported via NCB\_RETCODE is the last to have occurred.
  2. Extended status information, with the exception of adapter counters, is available in the NCB\_RESERVE field of the command block. In the case of the NCB.RESET command, it is the status prior to the NCB.RESET.
  3. Network status information:
    - Any network status bits 8 - 11 on, cause error code '4F'.
    - Any network status bits 12, 14, or 15 on, for longer than 60 seconds, cause error code '4E'. Code '4F' has priority over code '4E'.
    - Network status bits 6 and 7 do not cause errors. If bit 7 (counter overflow) is on, nothing is reported. If no network status appendage is defined, the local NETBIOS counters will be updated via the DIR.READ.LOG command. Bit 6 (single station) is ignored.
- 

## Adapter Status Parameter Table

This information is placed in PC system memory by the adapter support software in response to a DIR.STATUS command. The adapter support software places a pointer address in the ADAPTER\_PARAMS\_ADDR field of the DIR.STATUS command's parameter table.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	PHYS_ADDR	4	DB	Adapter physical address

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
4	UP_NODE_ADDR	6	DB	The upstream node address
10	UP_PHYS_ADDR	4	DB	The upstream physical address
14	POLL_ADDR	6	DB	Last poll address
20	AUTH_ENV	2	DB	Authorized environment
22	ACC_PRIORITY	2	DB	Transmit access priority
24	SOURCE_CLASS	2	DB	Source class authorization
26	ATT_CODE	2	DB	Last attention code
28	SOURCE_ADDR	6	DB	Last source address
34	BEACON_TYPE	2	DB	Last beacon type
36	MAJOR_VECTOR	2	DB	Last major vector
38	NETW_STATUS	2	DB	Network status
40	SOFT_ERROR	2	DB	Soft error timer value
42	FE_ERROR	2	DB	Front end error counter
44	LOCAL_RING	2	DB	Ring number
46	MON_ERROR	2	DB	Monitor error code
48	BEACON_TRANSMIT	2	DB	Beacon transmit type
50	BEACON_RECEIVE	2	DB	Beacon receive type
52	FRAME_CORREL	2	DB	Frame correlation save
54	BEACON_NAUN	6	DB	Beaconing station NAUN
60		4	DB	Reserved
64	BEACON_PHYS	4	DB	Beaconing station physical address

## Frame Status

The frame status (FS) byte is returned to the application program for some commands on the Token-Ring Network only.

**Note:** This FS byte is also referred to as the TRANSMIT\_PCFE field for transmit commands.

Some values and their meanings are:

**X'CC'** The frame was copied.

**X'00'** The address was not received and the frame was not copied (the destination adapter must not be on the ring).

**X'88'** The destination adapter recognized the frame, but did not copy it (possibly due to being overloaded).

See the *IBM Token-Ring Network Architecture Reference* for more about the FS byte.

---

### Exception Indications

The exception indications include:

- Adapter Check
- Network Status
- Bring-Up and Open Errors
- PC System Detected Errors
- System Action Exceptions for OS/2.

---

### Adapter Check

#### Adapter Check for CCB1

When an adapter check occurs, the adapter support software will close the adapter, all ring communication will cease. The adapter support software will assume the adapter has encountered an unrecoverable error. An adapter check appendage (ADAPTER\_CHECK\_EXIT), if defined by these commands(DIR.INITIALIZE, DIR.OPEN.ADAPTER, DIR.MODIFY.OPEN.PARMS, DIR.SET.USER.APPENDAGE), will be taken. On entry, the CX register will contain the adapter number, the AX will contain the adapter check reason code, and the ES and BX registers will point to the following table. While interrogating the information, the application program should either move the data to private memory or keep all interrupts masked off.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	NOTIFICATION_FLAG	4	DD	A pointer to the first of a queue of commands that were outstanding when the adapter closed
4	REASON_CODE	2	DW	Adapter check reason code, see "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43.
6	PARAMETER_0	2	DW	Parameter 0: Set per reason-code
8	PARAMETER_1	2	DW	Parameter 1: Set per reason-code
10	PARAMETER_2	2	DW	Parameter 2: Set per reason-code

#### Adapter Check for CCB2

In some instances, the adapter hardware or software is in such a state that operation is not possible. If this is the case, the following will occur.

1. If possible, the adapter will close and all network communications will cease.
2. All adapters will assume a closed state.
3. SAPs and link stations are closed due to the adapter closing and SAP and direct station buffer pools, outstanding receive frames, and CCBs may be returned to the application program if the ADAPTER\_CHECK\_FLAG is set.
4. If the ADAPTER\_CHECK\_FLAG is set the application program can be notified of this event. In order for an application to receive notification of an adapter check, a READ command must be issued before the event occurs requesting notification of Critical Exceptions. When the event occurs the READ command will be posted using a semaphore. The information listed in the following table will be copied into the READ command's CCB parameter table.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	NOTIFICATION_FLAG	4	DD	User exception flag
4	CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
6	EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed
10	BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
12	FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools
16	RCV_FRAME_COUNT	2	DW	Count of received frames chained to RCV_FRAME_ADDR
18	RCV_FRAME_ADDR	4	DD	Address of the first received frame of a possible chain of frames
22	REASON_CODE	2	DW	Adapter check reason code, see "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43.
<b>Note:</b> The next three fields are event error data.				
24	PARAMETER_0	2	DW	Parameter 0: set per REASON_CODE
26	PARAMETER_1	2	DW	Parameter 1: set per REASON_CODE
28	PARAMETER_2	2	DW	Parameter 2: set per REASON_CODE

---

### NOTIFICATION\_FLAG

**Explanation:** This user exception flag is ADAPTER\_CHECK\_FLAG as defined using the DIR.SET.EXCEPTION.FLAGS command.

---

### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in the SAP or Direct buffer pool.

Buffers provided with the DLC.OPEN.SAP or DIR.OPEN.DIRECT commands are returned to the application program when the adapter closes as a result of entering the closed state.

---

### RCV\_FRAME\_ADDR

**Explanation:** Address of the first received frame.

All received frames for this application program that were on the completion list at the time of the exception will be queued to this field when the adapter enters the closed state. The first buffer of each frame will point to the next frame.

---

### REASON\_CODE

**Explanation:** Reason code for the adapter check, see "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43.

**Note:** Only one reason code will be reported at a time.

---

### PARAMETERS - 0,1,2

**Explanation:** PARAMETER\_0, PARAMETER\_1 and PARAMETER\_2 provide additional information on a per reason-code basis. The information may be useful for maintenance purposes and is not intended for application program use.

## Adapter Check

### Adapter Check for CCB3

In some instances, the adapter hardware or software is in such a state that operation is not possible. If this is the case, the following will occur.

1. If possible, the adapter will close and all network communications will cease.
2. All adapters will assume a closed state.
3. If SAPs and link stations are closed due to the adapter closing and the adapter check appendage (ADAPTER\_CHECK\_APPNDG\_OFFSET) is defined, the SAP and direct station buffer pools and CCBs may be returned to the application program in the information table pointed to by registers ES and BX. When the adapter support software calls the application program's device driver, the appropriate event appendage offset is passed in register DI. See the DIR.SET.EXCEPTION.FLAGS command on page 3-50.
4. If the adapter check appendage offset is specified, the application program can be notified of this event. Once the adapter check occurs, the adapter support software notifies the application program of the event by calling the application program's device driver with the appropriate event appendage offset passed in register DI. The information listed in the following table will be provided in the table pointed to by registers ES and BX when the adapter support software calls the application program's device driver.

#### Application Calls

When the adapter support software calls the application program's device driver entry point, the following information is provided to the application program.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software the application program must remove the invocation code from the stack.
- Register DI contains the offset of the adapter check appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command
- Register DS contains the application program's device driver protect mode data segment.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of a 14 byte information table.
- Register AX contains the error code.

OFF-SET	PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
2	EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed
6	BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
8	FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools
12	REASON_CODE	2	DW	Adapter check reason code, see "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43.
<b>Note:</b> The next three fields are event error data.				
14	PARAMETER_0	2	DW	Parameter 0: set per reason-code
16	PARAMETER_1	2	DW	Parameter 1: set per reason-code
18	PARAMETER_2	2	DW	Parameter 2: set per reason-code

**FIRST\_BUFFER\_ADDR**

**Explanation:** Address of the first buffer in the SAP and direct station buffer pool.

Buffers provided with the DLC.OPEN.SAP or DIR.OPEN.DIRECT commands are returned to the application program when the adapter closes as a result of entering the closed state.

**REASON\_CODE**

**Explanation:** Reason code for the adapter check, see "Token-Ring Network Adapter Check Reason Codes for All CCBs" on page B-43.

**PARAMETERS - 0,1,2**

**Explanation:** PARAMETER\_0, PARAMETER\_1 and PARAMETER\_2 provide additional information on a per reason-code basis. The information may be useful for maintenance purposes only and is not intended for application program use.

**Token-Ring Network Adapter Check Reason Codes for All CCBs**

VALUE	FUNCTION	MEANINGS/PARAMETERS
8000	Adapter inoperative	See note.
<p><b>Note:</b> Adapter inoperative (8000)</p> <p>When a machine check occurs in the adapter processor, it is reported to the adapter support software via an "adapter check interrupt." The PC system may receive this interrupt before the adapter processor is able to set the adapter check bits. Therefore the adapter support software does the following:</p> <ol style="list-style-type: none"> <li>1. If a reason code is set, that code is passed to the application program.</li> <li>2. If a reason code is not set, the adapter support software goes into a tight loop for 250 milliseconds. The adapter support software then checks the reason code set by the adapter processor and does one of the following:               <ol style="list-style-type: none"> <li>a. If a code is set, the adapter support software passes that code to the application program.</li> <li>b. If no code is set, the adapter support software assumes that the adapter processor's machine check handler was not capable of executing because of the severity of the processor's problem. The adapter support software then sets a value of (8000) in the adapter check reason code and passes that code to the application program.</li> </ol> </li> </ol>		
4000	Reserved	
2000	Reserved	
1000	Illegal op code	The adapter detected an illegal op code (micro failure).
0800	Local bus parity error	The adapter local bus detected a parity error.
0400	External Parity error	
0200	Reserved	
0100	Internal parity error	
0080	Parity error, Ring transmit	The adapter local bus detected a parity error while transmitting on the ring.
0040	Parity error, Ring receive	The adapter local bus detected a parity error while receiving from the ring.
0020	Transmit underrun	
0010	Receive overrun	
0008	Unrecognized interrupt	
0004	Unrecognized error interrupt	
0003	Adapter detected no PC system service	
0002	Unrecognized supervisory request	
0001	Program detected error	



## Network Status

---

### PC Network Adapter Check Reason Codes for All CCBs

The values shown for the reason codes are as if the values were contained in the AX register.

Code	Meaning
X'0100'	Program Detected Error
X'0200'	Reserved
X'0300'	Invalid Supervisor Request Code
X'0400'	Invalid Task ID on Supervisor Call
X'0500'	Invalid Ready Task Request
X'0600'	Invalid Adapter Number on Supervisor Call
X'0700'	Invalid ES Value on Supervisor Call

---

## Network Status

### Network Status for CCB1

Whenever network status changes, the application program will be notified if the network status appendage has been defined in the `NETW_STATUS_EXIT` field of the CCB for a `DIR.INITIALIZE` or in the `NETW_STATUS_EXIT` field of the `DIRECT_PARMS` table of the `DIR.OPEN.ADAPTER` command. See "Token-Ring Network Status Codes for All CCBs" on page B-47 for the information returned.

The AX register contains the network status code and the CX register contains the adapter number. Registers ES and BX point to the chain of outstanding CCBs if the adapter was closed.

### Network Status for CCB2

In some instances the Network Status will indicate that the adapter closed due to an unrecoverable error, (Critical Network Status). When the adapter is closed, link stations are closed and the SAP and direct station buffer pools, outstanding receive frames, and CCBs may be returned to the application program if the `NETWORK_STATUS_FLAG` is set. See the `DIR.SET.EXCEPTION.FLAGS` command on page 3-50.

Whenever Network Status changes, the application program can be notified if the `NETWORK_STATUS_FLAG` is set. In order for an application program to receive notification of a Network Status (non-critical), a `READ` command must be issued. If a `READ` command is already outstanding, it will be posted immediately using a semaphore.

In order for an application program to receive notification of a Critical Network Status (the adapter closes), a `READ` command must be issued before the event occurs requesting notification of Critical Exceptions. When the event occurs, the `READ` command will be posted using a semaphore.

The information in the following table will be copied into the READ command's parameter table.

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
NOTIFICATION_FLAG	4	DD	User exception flag
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER *
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed *
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR *
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools *
RCV_FRAME_COUNT	2	DW	Count of received frames chained to RCV_FRAME_ADDR *
RCV_FRAME_ADDR	4	DD	Address of the first received frame of a possible chain of frames *
NETWORK_STATUS_CODE	2	DW	Network Status Code
* Indicates that this value is only returned when the adapter is closed as a result of encountering an unrecoverable error state.			

---

### NOTIFICATION\_FLAG

**Explanation:** User Notification Flag.

This user exception flag is NETWORK\_STATUS\_FLAG as defined using the DIR.SET.EXCEPTION.FLAGS command.

---

### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in the SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program if the adapter closes.

---

### RCV\_FRAME\_ADDR

**Explanation:** Address of the first received frame.

All received frames for this application program that were on the completion list at the time of the exception will be queued to this field if the adapter closes. The first buffer of each frame will point to the next frame.

---

### NETWORK\_STATUS\_CODE

**Explanation:** Network Status Code being reported.

See "Token-Ring Network Status Codes for All CCBs" on page B-47 when using a Token-Ring Network adapter, and "PC Network Status Codes for All CCBs" on page B-48 when using a PC Network adapter.

### Network Status for CCB3

In some instances the Network Status will indicate that the adapter closed due to an unrecoverable error, (Critical Network Status). When the adapter is closed, link stations are closed and the SAP and direct station buffer pools, and CCBs may be returned to the application program if the NETWORK\_STATUS\_APPNDG\_OFFSET is set. See the DIR.SET.EXCEPTION.FLAGS command on page 3-50.

Whenever the network status changes, the application program is notified by a NETWORK\_STATUS\_APPNDG\_OFFSET defined using the DIR.SET.EXCEPTION.FLAGS command. In order for an application program to receive notification of a network status exception (critical/non-critical), the DIR.SET.EXCEPTION.FLAGS command must have been executed and the adapter support software must have been supplied with an appendage offset for network status.

When the event occurs the adapter support software calls the application program's device driver entry point with the appropriate event appendage offset passed in register DI. The information listed in the following table will be supplied to the application program by the registers ES and BX. ES and BX point to the information table when the adapter support software calls to the application program's device driver.

**Application Calls:** When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the application program.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software the application program must remove the invocation code from the stack.
- Register DI contains the offset of the network status appendage as defined by the DIR.SET.EXCEPTIONS.FLAGS command.
- Register DS contains the application program device driver's protect mode data segment.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of the following 14 byte information table.
- Register AX contains the network status code.

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER *
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed *
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR *
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools *
NETWORK_STATUS_CODE	2	DW	Network Status Code
* Indicates that this value is only returned when the adapter is closed as a result of encountering an unrecoverable error.			

---

#### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in the SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program if the adapter closes.

**NETWORK\_STATUS\_CODE**

**Explanation:** Network Status Code being reported.

See “Token-Ring Network Status Codes for All CCBs” on page B-47 when using a Token-Ring Network adapter, and “PC Network Status Codes for All CCBs” on page B-48 when using a PC Network adapter.

**Token-Ring Network Status Codes for All CCBs**

BIT	FUNCTION	MEANING
15	Signal loss	Absence of any received signal detected.
14	Hard error	Beacon frames are being transmitted or received.
13	Soft error	This adapter has transmitted a soft error report MAC frame.
12	Transmit beacon	The adapter is transmitting beacon frames.
11	Lobe wire fault	An open or short circuit has been detected in the lobe data path. The adapter will be closed.
10	Auto-removal error 1	An adapter hardware error has been detected following the beacon auto-removal process. The adapter has been removed from the ring. The adapter will be closed.
9	Reserved	
8	Remove received	A remove MAC frame has been received. The adapter will be closed.
7	Counter overflow	One of the adapter error log counters has been incremented from 254 to 255. The DIR.READ.LOG command should be issued.
6	Single station	The adapter has opened and is the only station on the ring. The bit will be reset when another station is detected.
5	Ring Recovery	The adapter is transmitting or receiving Monitor Contention (claim token) MAC frames. This bit will be reset upon receipt of a Ring Purge MAC frame.
0 - 4	Reserved	

Multiple bits may be set when a network status change is posted.

For a beaconing condition the following network status events will be reported in the following order:

- For the station that will initially be transmitting the beacon frames
  - Ring Recovery (bit 5) set
  - Then after 1 second
    - Signal Loss (bit 15) set
    - Hard Error (bit 14) set
    - Transmit Beacon (bit 12) set.
- For the station that will initially be receiving the beacon frames
  - Ring Recovery (bit 5) set
  - Then after 1 second
    - Hard Error (bit 14) set.

The Ring Recovery bit will remain on for the entire time that the ring is beaconing. When using OS/2, if the ring is beaconing when an application program issues the DIR.OPEN.ADAPTER command and then issues the DIR.STATUS command the Ring Recovery bit will be set. However, bits 15, 14, and 12 may toggle depending on the immediate state of the ring seen by each adapter as a result of adapters doing diagnostic testing. When the ring stops beaconing all bits including the Ring Recovery bit will be zero.

## PC Network Status Codes for All CCBs

BIT	FUNCTION	MEANING
15	No Carrier	No carrier on this card during transmit.
14	Reserved	
13	Reserved	
12	Continuous Carrier	The adapter has detected Continuous Carrier on the network for more than 30 seconds.
11	Reserved	
10	Continuous Carrier	This adapter is generating Continuous Carrier. (The adapter will be closed.)
9	Reserved	
8	Remove Received	A network management REMOVE frame has been received. (The adapter will be closed.)
7	Counter Overflow	One of the Error Log Counters has incremented from 254 to 255.
0-6	Reserved	

## Bring-Up Errors for All CCBs

**For CCB1:** Bring-up testing is done when the DIR.INITIALIZE command is executed. If these tests are not completed successfully indicating an adapter failure, the bring-up error code will be returned in the BRING\_UP field of the DIR.INITIALIZE parameter table. The CCB\_RETCODE in the CCB will also contain '07' (command canceled: unrecoverable failure) when the command is terminated.

**For CCB2 and CCB3:** Bring-up testing is done during initialization when the adapter support software is loaded and when the DIR.INITIALIZE command is issued with the correct System Key. The results of the bring-up tests are returned to application programs when the DIR.OPEN.ADAPTER and DIR.INITIALIZE commands are executed.

The bring-up error code is included in the ADAPTER\_PARMS table of the DIR.OPEN.ADAPTER command.

If during system initialization bring-up testing is not successful, error messages are displayed and logged in the ACSLAN.LOG file.

The values shown for the bring-up codes are as if the values were contained in the AX register.

## Bring-up Error Codes for All CCBs

Code	8086 Type	Meaning
'0020'	DW	Diagnostics could not execute
'0022'	DW	ROM (ROS) diagnostics failed
'0024'	DW	Shared RAM diagnostics failed
'0026'	DW	Processor instruction test failed
'0028'	DW	Processor interrupt test failed
'002A'	DW	Shared RAM interface register diagnostics failed
'002C'	DW	Protocol-handler diagnostics failed
'0040'	DW	Adapter's programmable timer for the PC system failed (set by the PC system code)
'0042'	DW	Cannot write to shared RAM (set by the PC system code)

Code	8086 Type	Meaning
'0044'	DW	Reading from shared RAM read-only area caused an invalid error indication (interrupt) (set by the PC system code).
'0046'	DW	Writing into shared RAM read-only area did not cause an error indication (interrupt) (set by the PC system code).
'0048'	DW	Initialization timed out

## Token-Ring Network Adapter Open Errors for All CCBs

**For CCB1:** Adapter open testing is done when the DIR.OPEN.ADAPTER command is executed. If these tests do not complete successfully, indicating either an adapter failure or a ring problem, the open error codes will be returned in the OPEN\_ERROR\_CODE field of the DIR.OPEN.ADAPTER parameter table. The CCB\_RETCODE in the CCB will also contain '07' (command canceled—unrecoverable failure) when the command is terminated.

**For CCB2 and CCB3:** Adapter open testing is done at system initialization, or when a physical open is issued as a result of the DIR.OPEN.ADAPTER command being issued. If these tests do not execute successfully, any subsequent DIR.OPEN.ADAPTER commands will be terminated with a return code of X'07'.

The open error codes are passed back to the user in the DIR.OPEN.ADAPTER ADAPTER\_PARMS table.

## Open Error Codes for All CCBs

The open errors are returned in 2 bytes. The high-order byte is always zero and the low-order byte contains:

1. The phase of testing in which the error was encountered is in the high-order nibble (half-byte) of the low-order byte.
2. The error condition is in the low-order nibble of the low-order byte.

## Phases

VALUE	MEANING
'1n'	Lobe media test
'2n'	Physical insertion
'3n'	Address verification
'4n'	Roll call poll (neighbor notification)
'5n'	Request parameters

## Adapter Open Errors

### Errors

Value	Meaning
'n1'	Function failure
'n2'	Signal loss
'n3'	Reserved
'n4'	Reserved
'n5'	Timeout
'n6'	Ring failure
'n7'	Ring beaconing
'n8'	Duplicate node address
'n9'	Parameter request
'nA'	Remove received
'nB'	Reserved
'nC'	Reserved
'nD'	No monitor detected
'nE'	Monitor contention failed for RPL

### Suggested Actions in Response to Open Errors

When the following *Phase - Error* combination values are presented, they are the result of certain specific occurrences. Explanation of the occurrences follow with recommended actions listed. A list of recommended actions for both the application program and the PC system operator is provided following the explanations.

#### Explanations of Occurrences

**X'11' Lobe Media, Function Failure** *Failure Definition:* The testing of the lobe between the adapter and the access unit has been unsuccessful because the lobe has a bit-error rate that is too high or the adapter cannot receive successfully.

*Recommended Actions* 1, 3, and 5

**X'26' Physical Insertion, Ring Failure** *Failure Definition:* The adapter, acting as an active monitor, was unable to complete the ring purge function successfully indicating an error condition has occurred since the successful completion of monitor contention (claim token), when this adapter became the active monitor.

*Recommended Actions* 1 and 2a

**X'27' Physical Insertion, Ring Beaconing** *Failure Definition:* The adapter has detected one of the following conditions.

- The adapter tried to insert on a ring that was operating at a different data rate.
- A monitor contention (claim token) failure occurred.
- The adapter received a beacon MAC frame from the ring.

*Recommended Actions* 1, 2, and 2b

**X'2A' Physical Insertion, Timeout** *Failure Definition:* The adapter has received a remove ring station MAC frame indicating that a network management function has directed this adapter to get off the ring.

*Recommended Actions* 2a and 4

**X'2D' No Monitor Detected** *Failure Definition:* RPL station is the first station attempting to insert onto the ring.

*Recommended Actions 1 and 2a*

**X'2E' Monitor Contention Failed for RPL** *Failure Definition:* Physical insertion failure of RPL station.

*Recommended Action 2*

**X'32' Address Verification, Signal Loss** *Failure Definition:* The adapter has detected a 250-millisecond signal loss (receiver cannot recognize signal) indicating that an error condition has occurred since the adapter successfully completed the ring signal recognition phase of the open operation.

*Recommended Actions 1 and 2a*

**X'35' Address Verification, Timeout** *Failure Definition:* The insertion timer has expired before this function completed, indicating that the ring may be congested, experiencing a high bit-error rate, or losing an abnormally high number of tokens or frames, thus preventing successful Address Verification MAC frame transmissions.

*Recommended Actions 1 and 2a*

**X'36' Address Verification, Ring Failure** *Failure Definition:* The adapter, acting as an active monitor, was unable to complete the ring purge function successfully indicating an error condition has occurred since the successful completion of monitor contention (claim token), when this adapter became the active monitor.

*Recommended Actions 1 and 2a*

**X'37' Address Verification, Ring Beaconing** *Failure Definition:* The adapter has either detected a monitor contention (claim token) failure or received a beacon MAC frame from the ring.

*Recommended Actions 1 and 2b*

**X'38' Address Verification, Duplicate Node Address** *Failure Definition:* The adapter has detected that another station on the ring has an adapter address equal to the adapter address being tested.

*Recommended Action 4*

**X'3A' Address Verification, Remove Received** *Failure Definition:* The adapter has received a remove ring station MAC frame indicating that a network management function has directed this specific address to get off the ring.

*Recommended Actions 2a and 4*

**X'42' Ring Poll, Signal Loss** *Failure Definition:* The adapter has detected a 250-millisecond signal loss (receiver cannot recognize signal) indicating that an error condition has occurred since the adapter successfully completed the ring signal recognition phase of the open operation.

*Recommended Actions 1 and 2a*

**X'45' Ring Poll, Timeout** *Failure Definition:* The insertion timer has expired before this function completed, indicating that the ring may be congested, experiencing a high bit error rate, or losing an abnormally high number of tokens or frames. This prevents the adapter's successful reception of either the ring poll request or response MAC frame, or transmission of the required ring poll response MAC frame.

*Recommended Actions 1 and 2a*

**X'46' Ring Poll, Ring Failure** *Failure Definition:* The adapter, acting as an active monitor, was unable to complete the ring purge function successfully indicating an error condition has occurred since the successful completion of monitor contention (claim token), when this adapter became the active monitor.



## Adapter Open Errors

*Recommended Actions 1 and 2a*

**X'47' Ring Poll, Ring Beaconing Failure Definition:** The adapter has either detected a monitor contention (claim token) failure or received a beacon MAC frame from the ring.

*Recommended Actions 1 and 2b*

**X'4A' Ring Poll, Remove Received Failure Definition:** The adapter has received a remove ring station MAC frame, indicating that a network management function has directed this adapter to get off the ring.

*Recommended Actions 2a and 4*

**X'55' Request Parameters, Timeout Failure Definition:** The insertion timer has expired before this function completed, indicating that the ring may be congested, experiencing a high bit-error rate, or losing an abnormally high number of tokens or frames. This prevents successful transmission of the request parameter MAC frame or reception of either the set parameters 1 or set parameters 2 MAC frame (required response to the adapter's request).

*Recommended Actions 1 and 2a*

**X'56' Request Parameters, Ring Failure Failure Definition:** The adapter, acting as an active monitor, was unable to complete the ring purge function successfully, indicating an error condition has occurred since the successful completion of monitor contention (claim token), when this adapter became the active monitor.

*Recommended Actions 1 and 2a*

**X'57' Request Parameters, Ring Beaconing Failure Definition:** The adapter has received a beacon MAC frame from the ring.

*Recommended Actions 1 and 2b*

**X'59' Request Parameters, Parameter Request Failure Definition:** The adapter has detected that the ring parameter server is present on the ring, but that the required response (set parameters 1 or set parameter 2 MAC frame) has not been received in the allotted time. This indicates that the ring may be congested, experiencing a high bit-error rate, or losing an abnormally high number of tokens or frames.

*Recommended Actions 1 and 2a*

**X'5A' Request Parameters, Remove Received Failure Definition:** The adapter has received a remove ring station MAC frame, indicating that a network management function has directed this adapter to get off the ring.

*The Recommended Actions 2a and 4*

## Recommended Actions Table

NUM	Description
1	After delaying at least 30 seconds, retry the open two times, inserting the same delay between each retry.
2	After delaying at least 30 seconds, check the adapter configuration (especially the adapter data rate) and retry the open.
2a	If this error persists, direct the PC system operator to contact the network administrator for assistance and provide "Open Error" information.
2b	If this error persists, direct the PC system operator to contact the network administrator for assistance and provide information from the "Adapter Status Parameter Table" on page B-38.
3	Direct the PC system operator to contact the network administrator for assistance and provide "Open Error" information.
4	Direct the PC system operator to contact the network administrator for assistance and provide "Node Address" information and try attaching to the ring after 6 minutes.

NUM	Description
5	If this error persists, problem determination of the adapter or lobe is necessary. Contact your network administrator for problem determination assistance.

## PC Network Adapter Open Errors for All CCBs

The open error codes are returned in a DW, the high order byte is always zero:

VALUE	MEANING
X'0022'	No carrier
X'0023'	Continuous carrier
X'0033'	Unable to transmit
X'0038'	Duplicate node address
X'003A'	REMOVE frame received

## PC System Detected Errors

### PC System Detected Errors for CCB1

This appendage is defined by the DIR.OPEN.ADAPTER, DIR.SET.USER.APPENDAGE, DIR.INITIALIZE, and DIR.MODIFY.OPEN.PARMS commands.

**Note:** The PC System Detected error is not intended as a substitute for normal PC system error detection, for example, a divide check.

If the adapter support software detects an error condition in the PC system, the sequence of events will be determined by the type of error, which is passed to the appendage in register AL.

The following information is passed to the appendage:

1. Register AL contains the error code.
2. Register AH contains information on a per error code basis.  
If register AH is used, it is indicated in the status code explanation.
3. Register CX contains the adapter number.
4. Registers ES and BX:

If both registers are not X'0000', they are pointers to a queue of commands that were outstanding when the error occurred.

If both registers were X'0000', either the command executing could not be determined or it is not applicable.

The error code is passed to the appendage in register AX.

The PC System detected error codes are:

**Code**     **Meaning**

**X'0000'** Spurious interrupt detected.

**X'0001'** Access violation. An attempt to write into the read-only portion of shared RAM has occurred.

## PC System Detected Errors

- X'01xx'** An ARB command code error, where xx is the command code.
- X'02xx'** An ARB return code error, where xx is the return code from the adapter.
- X'03xx'** An SRB/SSB command code error, where xx is the CCB command code.
- X'04xx'** ARB transmit data request error, (the transmit CCB was not found) where xx is the command correlator from the adapter.

## PC System Detected Errors for CCB2

These types of errors are errors encountered by the adapter support software during operation.

**Note:** The PC System Detected error is not intended as a substitute for normal PC system error detection, for example, a divide check.

If the adapter support software detects an error condition in the PC system or OS/2 generates a return code that is not acceptable for the given situation, an error code is passed to the application program via the READ command.

In these cases the adapter will be closed, adapter support software will assume the adapter has encountered an unrecoverable error.

When the unrecoverable error is entered, link stations are closed and SAP and direct station buffer pools, outstanding receive frames, and CCBs may be returned to the application program if the PC\_ERROR\_FLAG is set. See DIR.SET.EXCEPTION.FLAGS on page 3-50.

Whenever a PC System Detected error occurs, the application program can be notified if the PC\_ERROR\_FLAG is set. In order for an application program to receive notification of a PC System Detected error a READ command must be issued before the event occurs requesting notification of critical exceptions. When the event occurs the READ command will be posted using a semaphore.

The information listed in the table below will be copied into the READ commands parameter table.

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
NOTIFICATION_FLAG	4	DD	User exception flag
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools
RCV_FRAME_COUNT	2	DW	Count of received frames chained to RCV_FRAME_ADDR
RCV_FRAME_ADDR	4	DD	Address of the first received frame of a possible chain of frames
ERROR_CODE	1	DB	Error Code
ERROR_DATA	1	DB	Error Data
FUNCTION_CODE	1	DB	Operating System/2 command code that failed
	1	DB	Reserved
	2	DW	Reserved
PROCESS_ID	2	DW	Process ID

---

**NOTIFICATION\_FLAG**

**Explanation:** User Notification Flag.

This user exception flag is the PC\_ERROR\_FLAG as defined using the DIR.SET.EXCEPTIONS.FLAG command.

---

**FIRST\_BUFFER\_ADDR**

**Explanation:** Address of the first buffer in SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program when the adapter closes as a result of an unrecoverable error.

---

**RCV\_FRAME\_ADDR**

**Explanation:** Address of the first received frame.

All received frames for this application that were on the completion list at the time of the exception will be queued to this field when the adapter encounters an unrecoverable error. The first buffer of each frame will point to the next frame.

---

**ERROR\_CODE/ERROR\_DATA**

**Explanation:** These two fields contain the error code and associated data for the PC System Detected error.

The list below indicates both the ERROR\_CODE and the ERROR\_DATA. The ERROR\_CODE is the first byte and the ERROR\_DATA is the second byte.

**X'0000'** A Token-Ring Network adapter interrupt has occurred and the interrupt was unexpected. For example, the adapter is not open.

**X'0001'** Access violation: an attempt was made to write to the read-only portion of shared RAM.

**X'01xx'** ARB command code error: an undefined ARB was returned from the adapter. The "xx" indicates the command code.

**X'02xx'** ASB return code error: there was an unexpected ASB interrupt from the adapter as a result of an ARB command. The "xx" will be set to the return code from the adapter.

**X'03xx'** SRB/SSB command code error. The "xx" will be set to the CCB command code.

**X'04xx'** ARB transmit data request error: the transmit CCB was not found. The "xx" is the command correlator from the adapter.

**X'05xx'** Unacceptable error conditions resulting from Operating System/2 return code values.

If while processing on the thread of an application program in the adapter support software device driver, an Operating System/2 return code is generated that is not acceptable for a given situation, the return code in the AX register will contain X'04'.

**X'06xx'** An adapter support software internal error. The "xx" will be set to an internal error code.

**Note:** Codes not shown (X'07' through X'7F') are reserved.

## PC System Detected Errors

---

### FUNCTION\_CODE

**Explanation:** The Operating System/2 command code that failed

This field is only used for an ERROR\_CODE of X'05'.

This field will contain the Function Code of a Device Help command that resulted in this PC System Detected error.

---

### PROCESS\_ID

**Explanation:** The Operating System/2 Process ID

This field is only used for the error codes X'05' and X'06', and will contain the Operating System/2 Process ID that was dispatched while the error occurred. A Process ID of X'0000' is used when an error occurs while processing an interrupt.

## PC System Detected Errors for CCB3

These types of errors are errors encountered by the adapter support software during operation.

**Note:** The PC System Detected error is not intended as a substitute for normal PC system error detection, for example, a divide check.

If the adapter support software detects an error condition in the PC system, or OS/2 generates a return code that is not acceptable for the given situation, an error code is passed to the application program when the adapter support software calls the application program's device driver with the appropriate event appendage offset passed in register DI. Upon entry into the application program's device driver, the AX register will contain the error code.

In these cases the adapter will be closed, the adapter support software will assume that the adapter has encountered an unrecoverable error.

When the unrecoverable error is entered, link stations are closed and SAP and direct station buffer pools and CCBs may be returned to the application program if the PC detected error appendage (PCERROR\_APPNDG\_OFFSET) has been defined using the DIR.SET.EXCEPTION.FLAGS command, see DIR.SET.EXCEPTION.FLAGS on page 3-50. If the appendage has been defined, the adapter support software will call the application program's device driver with the PC system detected appendage offset passed in register DI to notify the user of the error.

Whenever a PC System Detected error occurs, the application program can be notified if the PC detected error appendage has been defined using the DIR.SET.EXCEPTION.FLAGS command.

The information listed in the table below will be contained in the table pointed to by registers ES and BX when the adapter support software calls the application program's device driver with the appropriate event appendage offset passed in register DI.

**Application Calls:** When the adapter support software calls the application program at the address obtained by the ATTACHDD function, the following information is provided to the using application.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software the application must remove the invocation code from the stack.
- Register DI contains the offset of the adapter check appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program device driver's protect mode data segment.

- Register CX contains the adapter number.
- Registers ES and BX contain the address of the following 20 byte information table.
- Register AX contains the error code.

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP and direct station buffer pools
ERROR_CODE	1	DB	Error Code
ERROR_DATA	1	DB	Error Data
FUNCTION_CODE	1	DB	Operating System/2 command code that failed
	1	DB	Reserved
	2	DW	Reserved
	2	DW	Reserved

---

#### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program when the adapter closes as a result of an unrecoverable error.

---

#### ERROR CODE/ERROR DATA

**Explanation:** These two fields contain the error code and associated data for the PC System Detected error.

The list below indicates both the ERROR\_CODE and the ERROR\_DATA. The ERROR\_CODE is the first byte and the ERROR\_DATA is the second byte.

**X'0000'** A Token-Ring Network adapter interrupt has occurred and the interrupt was unexpected. For example, the adapter is not open.

**X'0001'** Access violation: an attempt was made to write to the read-only portion of shared RAM.

**X'01xx'** ARB command code error: an undefined ARB was returned from the adapter. The "xx" indicates the command code.

**X'02xx'** ASB return code error: there was an unexpected ASB interrupt from the adapter as a result of an ARB command. The "xx" will be set to the return code from the adapter.

**X'03xx'** SRB/SSB command code error. The "xx" will be set to the CCB command code.

**X'04xx'** ARB transmit data request error: the transmit CCB was not found. The "xx" is the command correlator from the adapter.

**X'05xx'** Unacceptable error conditions resulting from Operating System/2 return code values.

If while processing on the thread of an application program in the adapter support software device driver, an Operating System/2 return code is generated that is not acceptable for a given situation the immediate return code in the AX register will contain X'04'.

**X'06xx'** An adapter support software internal error. The "xx" will be set to an internal error code.

## System Action Exceptions

**Note:** Codes not shown (X'07' through X'7F') are reserved.

---

### FUNCTION\_CODE

**Explanation:** The Operating System/2 command code that failed.

This field is only used for an ERROR\_CODE of X'05'.

This field will contain the Function Code of a Device Help command that resulted in this PC System Detected error.

---

## System Action Exceptions for OS/2

### System Action Exceptions for CCB2

This exception is the result of a system administrator issuing commands using the SYSTEM.KEY defined by the configuration parameters. The following commands when issued with the SYSTEM.KEY will result in a System Action exception.

- DIR.CLOSE.ADAPTER
- DIR.CLOSE.DIRECT
- DIR.INITIALIZE
- DIR.READ.LOG
- DIR.SET.FUNCTIONAL.ADDRESS
- DIR.SET.GROUP.ADDRESS
- DLC.RESET

System Action exceptions result in an adapter closing, an adapter initializing, reading of either the adapter or direct interface logs, modification of the functional or group addresses, resetting link stations, or forced availability (closed) of the direct stations. When link stations are closed or the direct stations are closed due to a System Action exception, the link and direct station's buffer pools, outstanding receive frames, and CCBs may be returned to the application program if the SYSTEM\_ACTION\_APPENDG\_OFFSET is set. See the DIR.SET.EXCEPTION.FLAGS on page 3-50.

Whenever a System Action occurs, the application program will be notified if the SYSTEM\_ACTION\_FLAG is set and the using code has a READ outstanding. To receive notification of a System Action Exception resulting from the DIR.CLOSE.ADAPTER and DIR.INITIALIZE commands, the READ command must be issued before the event occurs. The information listed in the following table is copied into the READ command's parameter table and the READ command's semaphore is cleared to post the application program.

For the commands and associated application programs listed below, if the SYSTEM\_ACTION\_APPNDG\_OFFSET is set, the adapter support software will notify the application program of system action exceptions.

- DIR.CLOSE.ADAPTER for all application programs
- DIR.CLOSE.DIRECT for an application program owning the Direct Interface (a previously issued DIR.OPEN.DIRECT command was successful)
- DIR.INITIALIZE for all application programs
- DIR.READ.LOG for an application program owning the Direct Interface (a previously issued DIR.OPEN.DIRECT command was successful)
- DIR.SET.FUNCTIONAL.ADDRESS for all application programs
- DIR.SET.GROUP.ADDRESS for all application programs
- DLC.RESET for application programs owning the affected SAPs

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
NOTIFICATION_FLAG	4	DD	User exception flag
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed or when direct stations are closed
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP or DIRECT buffer pools
RCV_FRAME_COUNT	2	DW	Count of received frames chained to RCV_FRAME_ADDR
RCV_FRAME_ADDR	4	DD	Address of the first received frame of a possible chain of frames
SYSTEM_ACTION_ID	1	DB	System action identifier
SAP_STATION_RESET	1	DB	link station reset

---

### NOTIFICATION\_FLAG

**Explanation:** User Notification Flag.

This user exception flag is SYSTEM\_ACTION\_FLAG as defined using the DIR.SET.EXCEPTION.FLAGS command.

---

### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program if link stations are closed or reset.

---

### RCV\_FRAME\_ADDR

**Explanation:** Address of the first received frame.

All received frames for the affected link stations that were on the completion list at the time of the exception will be queued to this field. The first buffer of each frame will point to the next frame.



## System Action Exceptions

---

### SYSTEM\_ACTION\_ID

**Explanation:** System action identifier code.

The system action identifier code is passed to the user in the SYSTEM\_ACTION\_ID field. This code identifies the command issued that generated the system action exception.

- X'01'** DIR.CLOSE.ADAPTER command issued resulting in the adapter closing
  - X'02'** DIR.INITIALIZE command issued reinitializing the adapter
  - X'03'** DIR.READ.LOG command issued reading the adapter and/or direct interface logs
  - X'04'** DIR.SET.FUNCTIONAL.ADDRESS command issued modifying the functional address
  - X'05'** DIR.SET.GROUP.ADDRESS command issued modifying the group address
  - X'06'** DLC.RESET command issued resetting a single SAP  
When a single SAP is reset the SAP\_STATION\_RESET field will contain the SAP that was reset.
  - X'07'** DLC.RESET command issued resetting all link stations  
When all link stations are reset no link station values are returned.
  - X'08'** DIR.CLOSE.DIRECT command issued resulting in the direct stations closing and becoming available
- 

### SAP\_STATION\_RESET

**Explanation:** Link station reset.

If the SYSTEM\_ACTION\_ID is set to X'06' then this value contains the SAP number that was reset.

## System Action Exceptions for CCB3

This exception is the result of a system administrator issuing commands using the SYSTEM.KEY defined by the configuration parameters. The following commands when issued with the SYSTEM.KEY will result in a System Action exception.

- DIR.CLOSE.ADAPTER
- DIR.CLOSE.DIRECT
- DIR.INITIALIZE
- DIR.READ.LOG
- DIR.SET.FUNCTIONAL.ADDRESS
- DIR.SET.GROUP.ADDRESS
- DLC.RESET

System Action exceptions result in an adapter closing, an adapter initializing, reading of the either the adapter or direct interface logs, modification of the functional or group addresses, resetting link stations, or forced availability (closed) of the direct stations. When link stations are closed or the direct stations are closed due to a System Action exception, the link and direct station's buffer pools and CCBs may be returned to the application program if the user has passed the system action appendage (SYSTEM\_ACTION\_APPNDG\_OFFSET) to the adapter support software by issuing the DIR.SET.EXCEPTION.FLAGS command. See the DIR.SET.EXCEPTION.FLAGS on page 3-50.

Whenever a System Action occurs, the application program will be notified if the SYSTEM\_ACTION\_APPNDG\_OFFSET has been defined to the adapter support software. Once the exception has occurred, the information listed in the table below is copied into the table referenced by the

ES and BX registers when the adapter support software calls the application program's device driver with the appropriate event appendage offset passed in register DI.

For the commands and associated application programs listed below, if the SYSTEM\_ACTION\_APPNDG\_OFFSET is set, the adapter support software will notify the application program of system action exceptions.

- DIR.CLOSE.ADAPTER for all application programs
- DIR.CLOSE.DIRECT for an application program owning the Direct Interface (a previously issued DIR.OPEN.DIRECT command was successful)
- DIR.INITIALIZE for all application programs
- DIR.READ.LOG for an application program owning the Direct Interface (a previously issued DIR.OPEN.DIRECT command was successful)
- DIR.SET.FUNCTIONAL.ADDRESS for all application programs
- DIR.SET.GROUP.ADDRESS for all application programs
- DLC.RESET for application programs owning the affected SAPs

**Application Calls:** When the adapter support software calls the application program device driver entry point, the following information is provided to the using application.

- An Invocation code of X'0001' has been pushed onto the stack. Before returning to the adapter support software the application must remove the invocation code from the stack.
- Register DI contains the offset of the adapter check appendage as defined by the DIR.SET.EXCEPTIONS.FLAG command.
- Register DS contains the application program device driver's protect mode data segment.
- Register CX contains the adapter number.
- Registers ES and BX contain the address of the following 14 byte information table.
- Register AL contains the System Action ID.
- Register AH contains the SAP value associated with the System Action ID.

PARAMETER NAME	BYTE LEN	8086 TYPE	DESCRIPTION
CCB_COUNT	2	DW	Count of CCBs chained to EVENT_CCB_POINTER
EVENT_CCB_POINTER	4	DD	Pointer to the first of a queue of commands that were outstanding when the adapter closed or when the direct stations are closed
BUFFER_COUNT	2	DW	Count of buffers chained to FIRST_BUFFER_ADDR
FIRST_BUFFER_ADDR	4	DD	Address of first buffer in SAP or direct station buffer pools
SYSTEM_ACTION_ID	1	DB	System action identifier
SAP_STATION_RESET	1	DB	link station reset

---

### FIRST\_BUFFER\_ADDR

**Explanation:** Address of the first buffer in SAP and direct station buffer pools.

Buffers provided with the DLC.OPEN.SAP and DIR.OPEN.DIRECT commands are returned to the application program if link stations are closed or reset.

---

**SYSTEM\_ACTION\_ID**

**Explanation:** System action identifier code.

The system action identifier code is passed to the user in the SYSTEM\_ACTION\_ID field. This code identifies the command issued that generated the system action exception.

**X'01'** DIR.CLOSE.ADAPTER command issued resulting in the adapter closing

**X'02'** DIR.INITIALIZE command issued reinitializing the adapter

**X'03'** DIR.READ.LOG command issued reading the adapter and/or direct interface logs

**X'04'** DIR.SET.FUNCTIONAL.ADDRESS command issued modifying the functional address

**X'05'** DIR.SET.GROUP.ADDRESS command issued modifying the group address

**X'06'** DLC.RESET command issued resetting a single SAP

When a single SAP is reset the SAP\_STATION\_RESET field will contain the SAP that was reset.

**X'07'** DLC.RESET command issued resetting all link stations

When all link stations are reset no link station values are returned.

**X'08'** DIR.CLOSE.DIRECT command issued resulting in the direct stations closing.

---

**SAP\_STATION\_RESET**

**Explanation:** Link station reset.

If the SYSTEM\_ACTION\_ID is set to X'06' then this value contains the SAP number that was reset.

---

## **Appendix C. Local Area Network Program Listings**

---

### **The Sample Program Diskette**

Sample program listings are included on the diskette provided with this manual. These listings provide examples of NETBIOS use and the DLC interface use for both DOS and Operating System/2. The direct interface commands necessary to use the DLC interface are shown also.

See the READ.ME file on the diskette for a description of the sample program listings.

The materials on this diskette could include technical inaccuracies, typographical errors, or incompatibilities with IBM products, and could be modified or excluded from the generally available reference materials. IBM expressly reserves the right, without notice to you, to modify the information contained in this manual in a manner that affects the compatibility or usability of the application programs developed by you using the materials on this diskette.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS DISKETTE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

It is possible that this diskette may contain reference to, or information about IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

System Administrators may copy and distribute the sample programs on this diskette in any form without payment to IBM, for the purpose of developing, using, marketing, or distributing application programs for use with the IBM Token-Ring Network and IBM PC Network.

Attach a label that contains the following copyright notice to each copy:

Version 2.0 (C) Copyright International Business Machines Corp. 1986, 1988



---

## Appendix D. Local Area Network Support Program Information

16/4 Mbps Token-Ring Network Adapters	D-2
Token-Ring Network Adapter Failure to Open	D-2
PC BIOS and TIMERINT.SYS	D-3
Considerations When Using Two LAN Adapters	D-3
Using PC Network Adapters	D-4
NETBIOS Differences: Local Area Network Support Program Versions 1.1 and 1.0.	D-4
NETBIOS: Adapter Transmit/Receive Buffer Default Sizes	D-5
NETBIOS: Determination of Session Data Transmit Frame Size	D-5
Device Driver Summary	D-6
Token-Ring Network Device Driver Parameters	D-6
Token-Ring Network Adapter Addresses	D-6
Shared RAM	D-7
PC Network Device Driver Parameters	D-9
PC Network Adapter Addresses	D-9
Work Space	D-9
NETBIOS Device Driver Parameters	D-10
NETBIOS Device Driver Keyword Old Parameter Summary	D-10
NETBIOS Device Driver Keyword New Parameter Summary	D-11
NETBIOS Device Driver Keyword Parameter Descriptions	D-12
The Local Area Network Support Program Interrupt Arbitrator	D-16
Command Line Parameters	D-16
Registration Process Overview	D-17
Language Request CCB	D-18
Interface Registration CCB	D-18
NCB Handler Registration (NETBIOS using 802.2)	D-21
Additional Functions	D-22

The major portion of this Appendix describes how to code parameters for the device drivers that are supplied with the Local Area Network Support Program. These parameters customize the installation of the LAN Support Program for certain network application programs.

Also included are considerations when using the Token-Ring Network and PC Network adapters.

If any of the following applies, this Appendix will be of interest:

- If you want to customize the device drivers for certain network application programs.
- If you are using PC Network device drivers with heavy traffic and/or a large number of sessions (for example, a disk server).
- If you have an PC system that has (or you believe may have) a Basic Input/Output System (BIOS) date prior to June 1985.
- If you are using two Local Area Network (LAN) adapters in one computer.
- If your network is using the 16/4 Mbps Token-Ring Network adapters.
- If a Token-Ring Network adapter fails to open into the network even though the adapter diagnostics run correctly.
- If you are using the original IBM PC Network Adapter.

---

## 16/4 Mbps Token-Ring Network Adapters

The 16/4 Token-Ring Network adapters contain 64 KB of shared RAM that increases the number of available data buffers which allow the adapters to handle more traffic on the network. When installing the adapters you may set the shared RAM size to either 8, 16, 32, or 64 KB. When set to 16 KB the adapters allow access to the entire 64 KB shared RAM by using a technique called RAM Paging.

For most users it is recommended that these adapters be installed with a shared RAM size of 16 KB. This allows the entire 64 KB of shared RAM to be available with RAM Paging and minimizes the possibility of the adapter having a memory addresses conflict with other adapters that may be in the computer. This is especially true if you have two LAN adapters installed.

If 16 KB is not available in the computer then you will need to install the adapter with an 8 KB page size; in this case paging will not be available.

When operating at a data rate of 16 Mbps these adapters support the Early Token Release option. This option increases the utilization of the network by reducing the token latency time. This option can be disabled with a command line parameter for DXMC0MOD.SYS.

---

## Token-Ring Network Adapter Failure to Open

After installing a Token-Ring Network adapter you should run the adapter diagnostics that come on the diskette that is packaged with the adapter. This will verify that the adapter is operating correctly.

**Warning:** The test will not verify that the adapter data rate is set to the same data rate as that of the network.

If the adapter tries to open into the network at an incorrect data rate, the ring will beacon and all data on the network will stop temporarily.

The DXMT0MOD.SYS may automatically open the Token-Ring Network adapter into the network when the computer is started and the device driver is loaded into memory. If the adapter tries to open and then fails, error message "32E" will be displayed on the computer display. See Appendix A in the *IBM LAN Support Program User's Guide*.

If the Local Area Network Support Program device driver does not open the adapter at load time, the adapter will open into the ring at a later time when an application program is started. The application program will then detect the error and display its own error message when it opens the adapter.

If you suspect that the adapter data rate is different than that of the network then you will need to compare the two data rates.

1. Restart your computer. When the DXMC0MOD device driver is started an informational message is displayed that indicates the adapter data rate.
2. You will then need to determine the data rate of the network. If you do not know the data rate, check the adapter data rate of another station on the network that is known to be operating correctly. Another possibility is to ask your LAN administrator or the person that installed your LAN adapter. The data rate will be either 4 Mbps or 16 Mbps.
3. Compare the data rates from steps 1 and 2. If they are different, you will need to change the data rate. Refer to the installation instructions that came with the adapter.

---

## PC BIOS and TIMERINT.SYS

Included on the Local Area Network Support Program diskette is a file called TIMERINT.SYS. This file is a device driver that replaces the timer interrupt code on the IBM PC system BIOS (Basic Input/Output System). It reduces the likelihood of the DOS internal stack error. The CONFIG.SYS "STACKS = ..." statement is no longer required.

**Note:** This device driver should be used only on IBM PCs, PC XTs, and Personal Computer ATs that have BIOS dates earlier than June 1985.

To find out the BIOS date run the BIOSDATE.EXE. To do this, type "BIOSDATE" with the Local Area Network Support Program diskette as the default disk.

If your BIOS is dated prior to June 1985, copy the TIMERINT.SYS file onto your working diskette and include the following statement in the CONFIG.SYS file:

```
DEVICE=TIMERINT.SYS
```

---

## Considerations When Using Two LAN Adapters

The Local Area Network Support Program will support one or two LAN adapters. (Support for the original IBM PC Network Adapter is limited to a single adapter, see restrictions in "Using PC Network Adapters" on page D-4.) When two LAN adapters are used, one must be set up as "PRIMARY" and the other set up as "ALTERNATE." The setup is done when the adapter is installed. Refer to your adapter documentation for information on setting up the adapter.

If you are using both a Token-Ring Network adapter and a PC Network adapter then:

1. The PC Network adapter must be set up as "PRIMARY" and the Token-Ring Network adapter must be set up as "ALTERNATE."
2. The shared RAM and BIOS/MMIO addresses of the Token-Ring Network adapter must be changed to avoid memory conflicts with the PC Network adapter. If your Token-Ring Network adapter allows you to set the shared-RAM size, it is recommended that you verify and set the shared RAM size to 16 KB.
  - The suggested Token-Ring Network shared RAM address is D8000. (This address requires that the shared RAM size is 32 KB or less).
  - The suggested Token-Ring Network adapter BIOS/MMIO address is DC000.

If the Local Area Network Support Program Configuration Aid is used, the shared RAM address is changed to D8000 automatically; otherwise the address must be changed manually. The BIOS/MMIO address is set



when the Token-Ring Network adapter is installed. See the *Local Area Network Support Program User's Guide* for more information.

If the Local Area Network Support Program Configuration Aid is used, any required device drivers will be installed. If additional device driver parameters are desired, then the CONFIG.SYS file will need to be edited.

## Using PC Network Adapters

When using PC Network adapters, you may need to change the environment when there is a large amount of data traffic and/or many sessions are defined in one computer, for example a disk server.

The PC Network adapter device drivers (DXMGnMOD.SYS) have a "work area" parameter to allow for this situation. See "PC Network Device Driver Parameters" on page D-9.

In some situations where many computers are sending large amounts of information to one computer, increasing the receiving computer's work area is an incomplete solution. It may also be necessary to decrease the size of the transmit buffers in the sending computers. This is because the receiving adapter is receiving incoming traffic faster than it can be handled. The sending computer's transmit buffers may be changed in size by one of the following methods:

1. If NETBIOS is being used, the size of the transmit buffer may be changed via command line parameters on the NETBIOS device driver.
2. If the application program issues a DIR.OPEN.ADAPTER command, the size of the transmit buffer may be changed via the command.

**Note:** The following restrictions apply when supporting the original PC Network Adapter with the Device Driver, DXMG2MOD.SYS. This device driver allows the adapter to use IEEE protocols on the PC Network and communicate with the other supported PC Network adapters. The following restrictions apply when using this device driver.

- Only one IBM PC Network Adapter is supported in each PC system.
- No other communications adapters should be used in the PC system (e.g. LAN adapters, modems, etc.). The device driver cannot be used in LAN bridges or gateways.
- The performance of the device driver may not meet your needs if used on a PC system configured by the PC LAN Program as a file or print server.

---

## NETBIOS Differences: Local Area Network Support Program Versions 1.1 and 1.0.

1. Support for the 16/4 Mbps Token-Ring Network adapter has been added.
2. Default values of the STATIONS and SESSIONS parameters has been increased from 6 to 16.
3. The maximum DHB.SIZE parameter has been increased to 17960 bytes.
4. The X'4E' return code has been eliminated. An application program will now need to issue a LAN\_STATUS\_ALERT command to be notified of temporary errors that continue for more than 60 seconds.
5. New NCB commands have been added. See Chapter 4, "NETBIOS" for a complete description of these commands.
  - LAN\_STATUS\_ALERT.
  - SEND\_NO\_ACK.
  - CHAIN\_SEND\_NO\_ACK.

6. A 'X'FE' return code has been added. This return code is issued if NETBIOS commands are issued when the adapter is explicitly opened and NETBIOS is not operational. The adapter must first be closed prior to issuing any NETBIOS commands.

---

## NETBIOS: Adapter Transmit/Receive Buffer Default Sizes

When the adapter is opened by NETBIOS, the adapter's transmit and receive buffer sizes are determined by NETBIOS, unless specifically overridden. The adapter type and the number of link stations are used to determine these values:

Table D-1. Token-Ring Network adapter with 8 KB of Shared RAM			
Number of Link Stations	Number of Transmit Buffers	Transmit Buffers in Bytes	Receive Buffers in Bytes
1 to 6	2	1048	280
7 to 12	1	1048	192
13 to 18	1	600	144
19 to 24	1	600	112
more than 24	1	600	96

Table D-2. Token-Ring Network adapter with 16 KB of Shared RAM			
Number of Link Stations	Number of Transmit Buffers	Transmit Buffers in Bytes	Receive Buffers in Bytes
1 to 32	2	2040	280
33 to 48	2	1048	280
49 to 64	1	1048	280
more than 64	1	600	144

Table D-3. PC Network adapter			
Number of Link Stations	Number of Transmit Buffers	Transmit Buffers in Bytes	Receive Buffers in Bytes
1 to 32	2	2040	280
33 to 48	2	1048	280
49 to 64	1	1048	280
more than 64	1	600	144

---

## NETBIOS: Determination of Session Data Transmit Frame Size

When a session is established between two NETBIOS nodes, the length of the largest data frame sent between the two sessions is determined by three factors:

1. The length of the sender's transmit buffer
2. The amount of receive-buffer space in the receiver
3. The maximum frame that can pass any bridge in the path.

On each side (node) the receive buffer space calculation is made:

- The number of receive buffers is calculated.
- One-half of the number of receive buffers is multiplied by the data-size of one buffer. This determines the maximum size frame that may be transmitted to the session. The purpose for this is to ensure that one adapter may hold two maximum-size received frames.

---

## Device Driver Summary

The Local Area Network Support Program has seven different device drivers.

- One device driver (that is required in all installations) is an interrupt arbitrator that uses only one parameter, which defines the language that load-time error messages are displayed in. Languages other than English are supported in certain countries outside of North America. The parameter value is the same as used by the COUNTRY command defined in the DOS OPERATING SYSTEM REFERENCE manual.

**DXMA0MOD.SYS** Required in all installations

- The other device drivers can be used with parameters to customize the installation. For example, an adapter's address can be changed as well as other factors, such as work area size. Subsequent pages provide a description of the various parameters, and how to code them.

The device drivers fall into three categories:

The first two categories are adapter support device drivers. One of these device drivers is required for each adapter type. The third category is the NETBIOS Interface. It is required to allow NETBIOS application programs a program interface to the adapter support device drivers.

- Token-Ring Network adapter support device drivers

**DXMC0MOD.SYS** Supports IBM Token-Ring Network adapters

**DXMC1MOD.SYS** Use instead of DXMC0MOD when running the 3270 Workstation Program

- PC Network adapter support device drivers

**DXMG0MOD.SYS** Supports IBM PC Network adapters except the original IBM PC Network Adapter.

**DXMG1MOD.SYS** Use instead of DXMG0MOD when running the 3270 Workstation Program. This device driver does not support the *original* PC Network Adapter.

**DXMG2MOD.SYS** Supports the original IBM PC Network Adapter. See restrictions on use in "Using PC Network Adapters" on page D-4.

- NETBIOS Interface

**DXMT0MOD.SYS** Requires at least one of the adapter support drivers.

---

## Token-Ring Network Device Driver Parameters

There are three parameters that can be coded with the Token-Ring Network device drivers: adapter addresses, shared RAM locations, Early Token Release option. If your computer has two Token-Ring Network adapters, these parameters can be set individually for each adapter.

### Token-Ring Network Adapter Addresses

This address is in the range that is administered universally by the standards committees for local area networks. This UNIVERSALLY ADMINISTERED ADDRESS is the default address for the adapter if no other address is assigned.

If your network administrator chooses, he may assign a LOCALLY ADMINISTERED ADDRESS that will override the UNIVERSALLY ADMINISTERED ADDRESS on the adapter. This address must consist of 12 hexadecimal digits that range from '4000 0000 0000' to '4000 7FFF FFFF'. Values outside this range may be incompatible with other products on the network.

It is recommended that the address be restricted to decimal values. The LOCALLY ADMINISTERED ADDRESS would then range from '4000 0000 0000' to '4000 7999 9999' decimal.

**Note:** Make sure the address is recorded on the physical and logical address Locator Charts. These charts and an explanation of them can be found in the *IBM Token-Ring Network Introduction and Planning Guide*.

**Warning:** Each address must be unique on the LAN segment. An error will result when two or more adapters with the same address attempt to use the same LAN segment. Also it is recommended that each address be unique on the network.

To code the adapter address, see "Coding Parameters for the Token-Ring Network Device Drivers" on page D-7.

## Shared RAM

The Token-Ring Network Adapters contain random access memory (RAM) that is shared with the computer. The default segment location for the primary adapter (where the computer locates the RAM) is D800. (Note that segment D800 is memory address D8000.) For the alternate adapter it is D400. If the features that your system unit has installed use these locations, you will need to assign other locations that are not used. Valid locations must consist of four hexadecimal digits that represent the upper four digits of a five-digit hexadecimal address.

The Token-Ring Network PC Adapter Shared RAM address should be on a 8 KB boundary. The adapter needs 8 KB of memory above this address.

The Token-Ring Network PC Adapter II and the Token-Ring Network Adapter/A Shared RAM address should be on a 16 KB boundary. These adapters need 16 KB of memory above this address.

The Token-Ring Network 16/4 Adapter and the Token-Ring Network 16/4 Adapter/A allow you to set the shared RAM size to either 8, 16, 32, or 64 KB. The shared RAM address should be on the same boundary as the shared RAM size. For most application programs you should select the shared RAM size to be 16 KB. For this size the adapter uses a technique called RAM Paging to give you access to the entire 64 KB shared RAM. When you set a larger shared RAM size, you will need to be sure that the address range does not conflict with other adapters that may be in the computer.

To find locations that can be used, see the Technical Reference for your computer for a list of PC system memory locations. To code the shared RAM location, see the next section.

### Notes:

1. If you have two adapters, the shared RAM address for the alternate adapter must be different from the primary adapter.
2. If the Token-Ring Network Adapter/A or Token-Ring 16/4 Adapter/A is used then the shared RAM address parameter will be ignored. The shared RAM address is set when the adapter is installed.

**Coding Parameters for the Token-Ring Network Device Drivers:** These parameters are valid for DXMC0MOD.SYS and DXMC1MOD.SYS.

DEVICE = DXMCnMOD.SYS addr0,mem0,etr0,addr1,mem1,etr1

**addr0** Replace with the locally administered address for the primary or only adapter.

**mem0** Replace with the RAM location for the primary or only adapter.

**etr0** 0 = use Early Token Release, 1 = Do not use Early Token Release (parameter is ignored if adapter data rate is 4 Mbps).

- addr1** Replace with the locally administered address for the alternate adapter.
- mem1** Replace with the RAM location for the alternate adapter.
- etr1** 0 = use Early Token Release, 1 = Do not use Early Token Release. (parameter is ignored if adapter data rate is 4 Mbps)

If two Token-Ring Network adapters are installed in the same computer, the parameters are entered with the primary coded first and the alternate second. When coding a parameter list, observe these rules:

- Use a blank to separate the parameter list from 'DEVICE=DXMCnMOD.SYS'. (Either DXMC0MOD.SYS or DXMC1MOD.SYS.)
- Use a comma to separate one parameter from another.
- Indicate an omitted parameter with a comma.
- If one or more parameters are omitted from the end of the list, the commas are unnecessary.

#### Parameter List Coding Examples:

To specify a LOCALLY ADMINISTERED ADDRESS for the primary adapter, add a parameter to the CONFIG.SYS file, as in the following example:

```
DEVICE=DXMCnMOD.SYS 400000000001
```

In the example above, '400000000001' is the new address for the primary adapter, which will override the default address encoded on the adapter. All other defaults will be retained.

To specify a new shared RAM address for the primary adapter, add the parameter to the CONFIG.SYS file, as in the following example:

```
DEVICE=DXMCnMOD.SYS ,C400
```

In the example above, 'C400' is the new shared RAM segment (the physical address is C4000) for the primary adapter. The default location for the primary adapter will be overridden. All other defaults will be retained. Note the comma before the RAM location. This indicates that a parameter (the primary adapter address) has been omitted.

To disable the Early Token Release option add the parameter to the CONFIG.SYS file, as in the following example.

```
DEVICE=DXMCnMOD.SYS ,,1
```

In the example above, 1 disables the Early Token Release option. The parameter can be either 0 or 1. This parameter is only used for adapters with a data rate of 16 Mbps. The default value for 16 Mbps is to enable the option. For adapters with a data rate of 4 Mbps Early Token Release is not used.

Here are three more coding examples:

```
DEVICE=DXMCnMOD.SYS 400000000001,,400000000002
```

In the example above, '400000000001' is the address for the primary adapter, and '400000000002' is the address for the alternate adapter. The second comma in the list indicates that the default value for the primary shared RAM location is being retained. The third comma in the list indicates that the default value for Early Token Release is to be used. The alternate shared RAM location will also be retained since nothing is coded after the alternate adapter address in the list.

```
DEVICE=DXMCnMOD.SYS ,C400,,D800
```

In the example above, 'C400' is the shared RAM location for the primary adapter, and 'D800' is the shared RAM location for the alternate adapter. The default adapter addresses for the primary and alternate adapters will be retained.

DEVICE=DXMCnMOD.SYS 400000000001,C400,,400000000002,D800

In the example above, the defaults for both adapter addresses and both shared RAM locations have been changed.

---

## PC Network Device Driver Parameters

There are two kinds of parameters that can be coded with the PC Network device drivers: adapter addresses and work area size.

### PC Network Adapter Addresses

For adapter address information, see "Token-Ring Network Adapter Addresses" on page D-6.

### Work Space

The PC Network Device Drivers allow a maximum internal work area of 64 KB bytes per adapter, all of which may or may not be needed. The amount of work space depends on the requirements of the application program. The default work space is 8 KB bytes for each adapter in your computer.

Requirements for One Adapter	Work Space
Default	8 KB
Less than 16 sessions	8 KB
16 to 32 sessions	12 KB
24 to 32 sessions	16 KB

#### Notes:

1. The above table is a guideline. As the number of sessions and traffic increases, the work area may have to be increased.
2. To code the work space parameter, see the example below.

**Coding Parameters for the PC Network Device Drivers:** These parameters are valid for DXMG0MOD.SYS, DXMG1MOD.SYS and DXMG2MOD.SYS.

DEVICE = DXMGnMOD.SYS addr0,wrk0,addr1,wrk1

**addr0**        Replace with the locally administered address for the primary or only adapter.

**wrk0**        Replace with the work area for the primary or only adapter.

**addr1**        Replace with the locally administered address for the alternate adapter.

**wrk1**        Replace with the work area for the alternate adapter.

To code the adapter address, see the address examples in "Coding Parameters for the Token-Ring Network Device Drivers" on page D-7.

To specify a work area different from the default, add the parameter as in the following example:

DEVICE=DXMGnMOD.SYS ,20

In the example above, '20' is the new size of the work area for the primary adapter (20 KB). All other defaults will be retained. Note the comma before the work area size: this indicates that a parameter (the primary adapter address) has been omitted.

---

## NETBIOS Device Driver Parameters

The NETBIOS Interface has many optional parameters available to customize the driver for certain network application program. Only one of the parameters is position dependent. It is the Enable parameter and, if used, must be coded as the first parameter. See the following example.

```
DEVICE=DXMTOMOD.SYS ENABLE
```

**Note:** 'ENABLE' may be abbreviated as 'E'.

If not coded, the processing of a NETBIOS command occurs when the '5C' interrupt is issued. Because interrupts are disabled for a short time, some loss of data may occur in an asynchronous adapter if it is used simultaneously with a LAN adapter.

If the computer contains an asynchronous adapter that is used simultaneously with a LAN adapter then the Enable parameter should be used. In this case the NETBIOS command will be processed at a later time during a LAN adapter appendage interrupt. Since interrupts are enabled, data is not lost. Because of the short delay there may be some loss in performance.

## NETBIOS Device Driver Keyword Old Parameter Summary

There is a migration selection for users of previous NETBIOS Program releases. It allows the option of selecting the positional parameters that were used in the past. To do this, see the following example:

```
DEVICE=DXMTOMOD.SYS OLD.PARMS.0=x y z OLD.PARMS.1=a b c
```

"OLD.PARMS.0" is the keyword that indicates that the old form of positional parameters will be used for the primary or only adapter. The "x y z" indicates the old parameters:

- x** PC RAM used as internal work space, the default is 9 (9 KB). For large systems, this value may have to be increased (the maximum value is 64 - 64 KB).
- y** Extra SAPs. Indicates the number of "extra" (non-NETBIOS) SAPs that will be defined when NETBIOS opens the adapter. The default value is 0 and the maximum value is 99.
- z** Extra link stations. Indicates the number of "extra" (non-NETBIOS) link stations that will be defined when NETBIOS opens the adapter. The default value is 0 and the maximum value is 99.

**Note:** If 'y' is coded, 'x' must be coded. If 'z' is coded, both 'x' and 'y' must be coded.

"OLD.PARMS.1" is the keyword for positional parameters for the alternate adapter. The 'a b c' indicates the positional parameters.

Table D-4. NETBIOS 2.X (Default = OLD.PARMS) Parameters		
Load Parameters	Valid Values	2.X Defaults
Work Area	0 to 64 for 2.X	9
EXTRA.SAPs	0 to 99 *	0
EXTRA.STATIONS	0 to 99 *	0
* If this value is too large, the adapter open will fail. The maximum value allowed here typically exceeds current adapter maximum values.		

## NETBIOS Device Driver Keyword New Parameter Summary

On the following page is a table which indicates all the new NETBIOS Interface parameters. They are not position dependent. They need to be separated by a blank as shown below. It is also required that they be on the same line.

Note that keywords may be defined for either one or two adapters: if there are two adapters, the keywords are separated by a '/'. Further, any undefined keywords take the default value, on a per-adapter basis.

One or more keywords may be coded, per adapter:

- For the primary or only adapter, or to code keywords for the primary adapter and use all keyword defaults for the alternate adapter. In this instance, note that 64 sessions are defined:

```
DEVICE=DXMT0MOD.SYS ST=64 S=64 N=32 C=32
```

- To code keywords for the alternate adapter and use all keyword defaults for the primary adapter, if it exists (note the '/')

```
DEVICE=DXMT0MOD.SYS / ST=16 S=32 C=32
```

- To code keywords for both the primary and alternate adapters (note the '/')

```
DEVICE=DXMT0MOD.SYS ST=16 S=32 C=16 / ST=8 S=8
```

- To code keywords for the primary adapter and define the alternate adapter the same way (note the '='):

```
DEVICE=DXMT0MOD.SYS ST=16 S=32 C=16 / =
```

### Notes:

1. These parameters are used to customize the system, if required.
2. May not be used if OLD.PARMS.0 and/or OLD.PARMS.1 are used.
3. The keywords STATIONS, SESSIONS, and COMMANDS provide maximum values that will be used: if a program issues a RESET command, and the RESET indicates a value larger than indicated by these keywords, the value used will be the keyword value.
4. If any of these keywords are used, all other keywords will assume the default values indicated here.
5. The following section provides more detail on each parameter.

Table D-5 (Page 1 of 2). NETBIOS Device Driver Keyword Values					
Keyword	Abbreviations	Valid Values	2.0 Default	2.1 Default	2.X Default
STATIONS	ST	0 to 254 (1)	6	6	16
SESSIONS	S	0 to 254	6	6	16
COMMANDS	C	0 to 255	12	12	12
NAMES	N	0 to 254	17	17	17
CLOSE.ON.RESET	CR	Y(Yes)/N(No)	No	No	No
TRANSMIT.TIMEOUT	TT	0 to 20	2	1	1
TRANSMIT.COUNT	TC	0 to 10	3	6	6
DLC.MAXOUT	MO	0 to 9	2	2	2
DLC.MAXIN	MI	0 to 9	1	1	1
RING.ACCESS	RA	0 to 3	0	0	0
DLC.RETRY.COUNT	RC	0 to 255	8	8	8
EXTRA.SAPS	ES	0 to 99 (1)	0	0	0



Table D-5 (Page 2 of 2). NETBIOS Device Driver Keyword Values					
Keyword	Abbreviations	Valid Values	2.0 Default	2.1 Default	2.X Default
EXTRA.STATIONS	EST	0 to 99 (1)	0	0	0
DLC.T1	T1	0 to 10	5 (4)	5	5
DLC.T2	T2	0 to 11	2 (4)	2	2
DLC.Ti	Ti	0 to 10	3 (4)	3	3
DHB.SIZE	DS	200 to 17960 (1)	1048 (2)	1048 (2)	(3)
DHB.NUMBER	DN	0 to 9 (1)	(2)	(2)	(3)
RECEIVE.BUFFER.SIZE (4)	R	0 to 9999 (1)	280	280	(3)
OPEN.ON.LOAD	O	Y(Yes)/N(No)	Yes	Yes	Yes
DATAGRAM.MAX	DG	Y(Yes)/N(No)	No	No	No
RESET.VALUES	RV	Y(Yes)/N(No)	N/A (5)	No	No
ADAP.ADD.NOT.REVERSED	ANR	Y(Yes)/N(No)	N/A (5)	No	No
REMOTE.NAME.DIRECTORY	RND	0 to 255	N/A (5)	0	0
REMOTE.DATAGRAM.CONTROL	RDC	Y(Yes)/N(No)	N/A (5)	No	No

**Notes:**

1. If this value is too large the adapter open will fail. The maximum value allowed here typically exceeds current adapter maximum values.
2. The exception is that in the case of NETBIOS 2.0 or 2.1, the DHB size of 2040 bytes is used for a Token-Ring Network adapter with a 16Kb of shared RAM.
3. The default value for NETBIOS 2.2 set by the NETBIOS Interface Device Driver is based upon the number of stations.  
The valid value for NETBIOS 2.0 and 2.1 versions are 0 and 200 to 9999.
4. If a value is specified, its value is not checked by the NETBIOS Interface Device Driver. The value is checked by the adapter on open. Current adapters require a minimum value of 96.
5. The keyword option is not implemented in NETBIOS 2.0.

**NETBIOS Device Driver Keyword Parameter Descriptions**

Each keyword has an abbreviated version, which is indicated in parentheses following the keyword.

**STATIONS = (ST =)**

- Indicates the maximum number of NETBIOS Interface link stations that may be defined.
- Must be a decimal number from 0 to 254.
  - The realistic maximum is affected by the size of the shared RAM on the adapter.
  - The EXTRA.STATIONS value reduces the maximum for the NETBIOS Interface.
- If omitted or 0, the default value is 16.

**SESSIONS = (S =)**

- Indicates the maximum number of NETBIOS Interface Device Driver sessions that may be defined.
- Must be a decimal number from 0 to 254.
- If omitted or 0, the default value is 16.
  - If the value is less than STATIONS, it will be set to the value of STATIONS.

**COMMANDS = (C =)**

- Indicates the maximum number of NETBIOS Interface commands (MCB) that may be outstanding at one time.
- Must be a decimal number from 0 to 255.
- If omitted or 0, the default value is 12.

**NAMES = (N =)**

- Indicates the maximum number of NETBIOS Interface Device Driver names that may be defined, including the NAME\_NUMBER\_1.
- Must be a decimal number from 0 to 254.
- If omitted or 0, the default value is 17.
- If 1, the value will be set to 2.

**OPEN.ON.LOAD = (O =)**

- If OPEN.ON.LOAD= YES, the NETBIOS Interface will open the adapter at load time. This will eliminate the delay caused by adapter open when an application program later is loaded and issues the first MCB.
- Must be coded as either Y(yes) or N(no).
- If omitted, the default is YES.

**DATAGRAM.MAX = (DG =)**

- Indicates that the maximum length datagram transmitted by the NETBIOS Interface Device Driver will be based on the size of the adapter's transmit buffer or data hold buffer (DHB). Normally, a datagram is limited to 512 bytes.
- Must be coded as either Y(yes) or N(no).
- If coded as DG=Y, the maximum length of a datagram will be:
  - DHB length - 86 (The size of the transmit buffer less 86 bytes.)
- If omitted or N, this function will not be performed.

**CLOSE.ON.RESET = (CR =)**

- Indicates that the NETBIOS Interface will close the adapter and then re-open it when a RESET is issued.
- Must be coded as either Y(yes) or N(no).
- If omitted or N, this function will not be performed.

**DHB.SIZE = (DS =)**

- Indicates the size of the DHB: the adapter's transmit buffer(s).
- Must be a decimal number within the range of acceptability to the adapter.
  - Note that the minimum size allowed by NETBIOS Interface is 200, therefore the value coded must be either 0 or a number 200 to 17960.
- If omitted or 0, the value will be determined by NETBIOS Interface.

**DHB.NUMBER = (DN =)**

- Indicates the number of DHBs allocated by the adapter.
- Must be a decimal number within the range of acceptability to the adapter.
- If omitted or 0, the value will be determined by NETBIOS Interface Device Driver.

**RECEIVE.BUFFER.SIZE = (R =)**

- Indicates the size of the adapter's receive buffers. A minimum of 8 adapter buffers of the specified size will be allocated.
- Must be a decimal number within the range of acceptability to the adapter.

- If omitted or 0, the value will be determined by the NETBIOS Interface.

**TRANSMIT.TIMEOUT = (TT =)**

- Indicates the time between transmission of queries (for instance, CALL).
- The time is indicated in 1/2-second periods.
- Must be a decimal number from 0 to 20.
- If omitted or 0, the default is 1 (1/2 second).

**TRANSMIT.COUNT = (TC =)**

- Indicates the number of times queries (for instance, CALL) are transmitted.
- Must be a decimal number from 1 to 10.
- If omitted or 0, the default is 6.

**DLC.MAXOUT = (MO =)**

- Sets the MAXOUT value for all NETBIOS Interface link stations.
- Must be a decimal number from 1 to 9.
- If omitted or 0, the default is 2.

**DLC.MAXIN = (MI =)**

- Sets the MAXIN value for all NETBIOS Interface link stations.
- Must be a decimal number from 1 to 9.
- If omitted or 0, the default is 1.

**RING.ACCESS = (RA =)**

- Indicates the adapter's ring access priority for NETBIOS Device Driver Interface messages.
- Must be a decimal number from 0 to 3.
- If omitted or 0, the default is 0.
- The higher the number, the higher the priority.

**EXTRA.SAPS = (ES =)**

- Indicates additional SAPS requested by the NETBIOS Interface when it implicitly opens the adapter. These SAPs are not used by the NETBIOS Interface.
- Must be a decimal number from 0 to 99.
- If omitted or 0, no additional SAPs are requested.

**Note:** If this value exceeds adapter maximum values, the error will be detected when the adapter is opened. See the *Local Area Network Support Program User's Guide*.

**EXTRA.STATIONS = (EST =)**

- Indicates additional link stations requested by the NETBIOS Interface when it implicitly opens the adapter. These link stations are not used by the NETBIOS Interface.
- Must be a decimal number from 0 to 99.
- If omitted or 0, no additional stations are requested.

**Note:** If this value exceeds adapter maximum values, the error will be detected when the adapter is opened. See the *Local Area Network Support Program User's Guide*.

**REMOTE.NAME.DIRECTORY = (RND =)**

- May be coded as a number from 0-255.
- If omitted or 0, the RND function is not implemented: all CALLS, STATUS QUERIES, and SEND DATAGRAMS will operate as in previous levels of NETBIOS and broadcast messages to all NETBIOS nodes.
- If a non-zero value is coded, the RND function is implemented with the value indicating the number of remote names that may be saved by the local node. Note that four is the

minimum number of name entries: if coded as 1, 2, or 3 the value of 4 is used. When RND is implemented (after the local station has located a remote name), the remote node address is saved and subsequent messages to that name will be to a specific node rather than a broadcast to all nodes. This is true for CALLS, STATUS QUERIES, and SEND DATAGRAMS (see next keyword REMOTE.DATAGRAM.CONTROL).

- **Important:** Using the RND can significantly improve performance in situations where large numbers of DATAGRAMS, CALLS, and REMOTE\_ADAPTER\_STATUS commands are used.
- **Warning:** When the RND function is being used, a duplicate network name will not be detected by a CALL function unless it is the first time that the remote (CALL) name has been used.

#### **REMOTE.DATAGRAM.CONTROL = (RDC =)**

- This keyword is meaningless if REMOTE.NAME.DIRECTORY is omitted or coded as 0.
- If omitted or coded as NO, SEND DATAGRAMS will not use the RND function.
- If coded as YES, SEND DATAGRAMS will also use the RND for sending to remote nodes. Note that for datagrams, if a bridge in the transmission path drops out, there is no indication that the path is no longer valid. Therefore, if this function is used, the user may wish to update the RND if he becomes aware of the fact that the remote node is not receiving datagrams. This may be accomplished by having the application program issue a FIND\_NAME command with a receive buffer length of zero. This is a new function to allow an RND update without having to execute either a CALL or REMOTE ADAPTER STATUS.

#### **RESET.VALUES = (RV =)**

- If omitted or coded as NO, RESET works as it does in earlier releases of NETBIOS.
- If coded as YES, when a RESET is issued:
  - If the number of sessions is coded as zero in the RESET, the default will be the value defined by the keyword SESSIONS= rather than the normal default value of 6.
  - If the number of commands is coded as zero in the RESET, the default will be the value defined by the keyword COMMANDS= rather than the normal default value of 12.

#### **ADAP.ADDR.NOT.REVERSED = (ANR =)**

- If omitted or coded as NO, the adapter's "burned-in address" as used in the ADAPTER\_STATUS command and in "NAME\_NUMBER\_1" will be byte-reversed (if the value is X'010203040506', it will be saved as X'060504030201').
- If coded as YES, the adapter's "burned-in address" as used in the ADAPTER\_STATUS command and in "NAME\_NUMBER\_1" will be in proper order (if the value is X'010203040506', it will be saved as X'010203040506').

#### **DLC.RETRY.COUNT = (RC =)**

- Determines the number of retry attempts to be made by the adapter's LLC code.
- May be a value from 1-255.
- If omitted or 0, the default value of 8 is used.

#### **DLC.T1 = (T1 =)**

- Determines the value of the T1 timer in the adapter's LLC code.
- May be a value from 0-10.
- If omitted or 0, the default value of 5 is used.

#### **DLC.T2 = (T2 =)**

- Determines the value of the T2 timer in the adapter's LLC code.
- May be a value from 0-11.
- If the value is 11, the T2 timer function is not implemented.
- If omitted or 0, the default value of 2 is used.

## **DLC.TI= (TI=)**

- Determines the value of the TI timer in the adapter's LLC code.
- May be a value from 0-10.
- If omitted or 0, the default value of 3 is used.

---

## **The Local Area Network Support Program Interrupt Arbitrator**

This is the description of the software interface to, and the registration process of the Local Area Network Support Program Interrupt Arbitrator. The Local Area Network Support Program Interrupt Arbitrator is a software module provided with the Local Area Network Support Program diskette. The purpose of the Local Area Network Support Program Interrupt Arbitrator is to remove the majority of the load order dependencies of the interfaces provided by the Local Area Network Support Program.

**Important:** In this appendix the term CCB will apply only to the IEEE 802.2 adapter support software supplied with the Local Area Network Support Program.

Currently there are two software interfaces to IBM's network adapters.

- NETBIOS
- The IEEE 802.2 Interface

A usability problem arises due to the fact that both of these interfaces utilize the X'5C' software interrupt in their API.

Since two network adapters may be present in a PC system, multiple IEEE 802.2 interfaces and/or NETBIOS may be present in a PC at one time. This currently requires that each interface know whether another interface is also using the X'5C' interrupt, so that it can pass any control blocks that it does not understand (or are not for its adapter) to the next X'5C' interrupt handler in the chain.

This need to know whether the other interfaces are using the X'5C' interrupt results in replicated logic at the entry points to the interrupt handlers. In addition, due to migration considerations, the interrupt handlers may have to be loaded in a specific sequence to avoid conflicts.

The purpose of the Local Area Network Support Program Interrupt Arbitrator is to eliminate the above problems by having a single owner of the X'5C' software interrupt. The Local Area Network Support Program Interrupt Arbitrator takes over the X'5C' interrupt vector. It then allows the other users of the X'5C' interrupt to register with the Local Area Network Support Program Interrupt Arbitrator. Registering programs specify which interface they provide (IEEE 802.2 or NETBIOS) and for which network adapter (0 or 1). The Local Area Network Support Program Interrupt Arbitrator then monitors the flow of control blocks across the X'5C' interrupt and routes the control block to the appropriate interface.

The use of the Local Area Network Support Program Interrupt Arbitrator eliminates the two previously stated problems. Since there is a single owner of the X'5C' interrupt, all logic for determining how to route the control blocks is contained there, not replicated in each individual interface. In addition, since each interface registers its needs with the Local Area Network Support Program Interrupt Arbitrator, the interface modules can be loaded in any sequence. The only requirement is that the Local Area Network Support Program Interrupt Arbitrator must be loaded first.

### **Command Line Parameters**

The command line for the Local Area Network Support Program Interrupt Arbitrator has the following format in the CONFIG.SYS file:

*DEVICE=DXMA0MOD.SYS <Language Code>*: where the language code option is a two character hexadecimal number that corresponds to the DOS country code. This code indicates which language text messages are to be printed. This language code is translated to a simple index that can be requested by any registering interface. This allows all Local Area Network Support Program device drivers to print error messages in the same language.

The valid values for this field and their translated values are shown in the following table.

Language	DOS Country Code	Language Index
US English	01	00
Danish	45	01
German	49	03

If this field is omitted or an unrecognized value is detected, a default of US English is used.

## Registration Process Overview

When a network interface program wishes to register with the Local Area Network Support Program Interrupt Arbitrator, it should perform the following functions during its initialization sequence:

1. Test that the Local Area Network Support Program Interrupt Arbitrator present flag (bit 5 at memory location 0040:00A1) is set. If the Local Area Network Support Program Interrupt Arbitrator present flag is not set the requesting interface will abort its load and print the appropriate error message (in English) indicating that the required software is not present.
2. Build a language request CCB (Refer to "Language Request CCB" on page D-18 for the definition of this CCB).
3. Place the address of the language request CCB in ES and BX.
4. Execute an INT X'5C' instruction.
5. Save the returned language code index.
6. Test the CCB\_RETCODE field.
  - If CCB\_RETCODE is X'83', abort the load.
  - Otherwise continue the load sequence.
7. Build the appropriate network interrupt registration request CCB (Refer to "Interface Registration CCB" on page D-18 and "NCB Handler Registration (NETBIOS using IEEE 802.2)" on page D-21 for the definitions of these CCBs).
8. Place the address of the registration request CCB in ES and BX.
9. Execute an INT X'5C' instruction.
10. Test the return code field; if this field is non-zero the registration request has failed.
  - If the registration request fails, the requesting program should print an error message to the standard output device, and abort the load. The error message displayed depends on the return code.
  - If the return code field is zero, the registration request has been accepted and the requesting program can continue its load sequence.

## Language Request CCB

### LANGUAGE REQUEST

#### Return Codes (CCB\_RETCODE):

X'00': Operation completed successfully

X'83': Local Area Network Support Program Interrupt Arbitrator load failed

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	X'31': Language Request
2	CCB_RETCODE	1	DB	Set by Network Arbitrator on return
3	CCB_LC	1	DB	* Language code for messages
4		12	DB	Reserved
* Indicates a returned value. (for example, a value set by the Local Area Network Support Program Interrupt Arbitrator).				

#### CCB\_ADAPTER

**Explanation:** This command is adapter independent. However, the value contained in this field must be less than X'04'.

#### CCB\_LC

**Explanation:** The Local Area Network Support Program Interrupt Arbitrator uses this field to return a language index to the registering interface. The registering interface should use this value to determine the language in which to print any error messages. Refer to "Command Line Parameters" on page D-16 for the meaning of the values returned in this field.

## Interface Registration CCB

### REGISTRATION REQUEST

#### Return Codes (CCB\_RETCODE):

X'00': Operation completed successfully

X'80': CCB interface already registered for requested adapter

X'81': NETBIOS already registered for requested adapter

X'83': Local Area Network Support Program Interrupt Arbitrator load failed

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	X'32': Network interrupt registration
2	CCB_RETCODE	1	DB	Set by Network Arbitrator on return
3	CCB_WORK	1	DB	Reserved
4	5C_ENTRY	4	DD	@ to pass control for any CCB and NCBs
8	NCB_FLAGS	4	DD	@ of flags passed to NETBIOS is AL
* Indicates a returned value (for example, a value set by the Local Area Network Support Program Interrupt Arbitrator).				
@ Indicates an address throughout this document.				

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
12	CCB_PARM	1	DB	Defines type of registration request
12	CCB_CONFIG	1	DB	Interface configuration data
14	CCB_TRACE	2	DB	Offset to CCB trace routine
* Indicates a returned value (for example, a value set by the Local Area Network Support Program Interrupt Arbitrator).				
@ Indicates an address throughout this document.				

---

## CCB\_ADAPTER

**Explanation:** This field indicates which adapter the registration request is for. If an interface wishes to register for both adapters, it must issue two separate registration requests.

---

## 5C\_ENTRY

**Explanation:** This is the address to which the Local Area Network Support Program Interrupt Arbitrator passes control when it detects a CCB or NCB for the specified adapter. The Local Area Network Support Program Interrupt Arbitrator pushes all registers (except for AX) onto the stack and then transfers control to the specified address by a simulated INT instruction. The registers (except for AX) will be restored after the registered interface executes the IRET instruction to return control to the application program that issued the CCB or NCB.

The registers are pushed in the following order: BX, CX, DX, DI, SI, BP, ES, DS.

---

## NCB\_FLAGS

**Explanation:** This is the address of the 1 byte flag field that is to be passed to the NETBIOS interface that is registered for the same adapter number when an appropriate NETBIOS is detected. The byte pointed to by this field is loaded into the AL register before control is passed to NETBIOS. This field is only meaningful for a CCB interface registration. The contents of this field are ignored for a NETBIOS interface registration.

---

## CCB\_PARM

**Explanation:** This is a bit significant field that is used to determine the type of registration request. The definition of the bits contained in this field are as follows:

**BIT 0** CCB Interface (IEEE 802.2 Interface)

This bit indicates that the requesting interface is an IEEE 802.2 CCB interface.

**BIT 1** NETBIOS (NETBIOS not using IEEE 802.2)

This bit indicates that the requesting interface is a NETBIOS interface.

**BITS 2-6** Reserved

**BIT 7** Override Indicator

This bit indicates that the requesting interface wishes to override any currently registered interface on the specified adapter. An example of the use of this bit would be for the Local Area Network Support Program product. Since Local Area Network Support Program converts the NCB interface on a original PC Network Adapter to an IEEE 802.2 CCB interface, the override bit would be required because the original PC Network Adapter would already be "auto-registered" as an NCB interface. This bit is only meaningful when used in conjunction with bit 2 of this field.



---

## CCB\_CONFIG

**Explanation:** This field is used to determine the configuration characteristics of the registering interface. For CCB interfaces, the contents of this field are passed to any registering NCB interfaces in the CCB\_WORK field of the DEFINE.MIF command used to register the NCB interface. Refer to "NCB Handler Registration (NETBIOS using IEEE 802.2)" on page D-21 for the definition of the DEFINE.MIF command. The definition of the bits contained in this field are:

**BIT 0** PC 3270 Control Program Supported

This bit indicates that the requesting interface supports the 3270 Control Program with the XMA adapter. When this bit is set the Local Area Network Support Program Interrupt Arbitrator activates logic that provides the required environment data to the registered interface via the interface defined in "PC 3270 Support" on page D-22.

**BITS 1-7** Reserved

---

## CCB\_TRACE

**Explanation:** This field contains the offset portion of the address to the CCB trace routine. The segment portion of the address is assumed to be the same as the 5C\_ENTRY segment.

This address allows the Local Area Network Support Program Interrupt Arbitrator to pass trace data to the registered CCB interface's trace routine for:

- NCB's passed to NETBIOS that is using the registered CCB interface.
- CCB's that are rejected by the Local Area Network Support Program Interrupt Arbitrator.

When trace is active the Local Area Network Support Program Interrupt Arbitrator will make a FAR CALL to the specified address with the registers set as follows:

- If the traced control block is as NCB, the registers values are:
  - CH = Adapter Number
  - CL = X'0F' (Trace ID)
  - DH = Return Code
  - DL = Command Code
  - SI = Number of bytes pushed on the stack since the INT X'5C' plus 4. (This includes the four bytes pushed on the stack during the FAR CALL.)
    - This value is used to determine the location of the interrupt address on the stack.
- If the traced control block is a CCB, the register values are:
  - CH = X'00'
  - CL = Adapter Number
  - DH = Return Code
  - DL = Command Code
  - SI = Number of bytes pushed on the stack since the INT X'5C' plus 4. (This includes the four bytes pushed on the stack during the FAR CALL.)
    - This value is used to determine the location of the interrupt address on the stack.

## NCB Handler Registration (NETBIOS using IEEE 802.2)

NCB registration is done using a modified DIR.DEFINE.MIF.ENVIRONMENT command. The command is essentially the same as the DIR.DEFINE.MIF.ENVIRONMENT command with one exception:

- The CCB\_WORK field has been changed to the CCB\_CONFIG field.

After the Local Area Network Support Program Interrupt Arbitrator has used the DEFINE.MIF command to register the interface with the Local Area Network Support Program Interrupt Arbitrator, the CCB is forwarded to the appropriate CCB interfaces for further processing.

### NCB REGISTRATION

#### Return Codes (CCB\_RETCODE):

- X'00': Operation completed successfully
- X'01': Invalid command code
- X'1B': Invalid CCB\_PARM\_TAB pointer
- X'1C': Invalid pointer in the CCB parameter table
- X'1D': Invalid Adapter Number (No CCB interface registered)
- X'82': NCB interface already registered for requested adapter

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	CCB_ADAPTER	1	DB	Adapter 0 or 1
1	CCB_COMMAND	1	DB	X'2B': DIR.DEFINE.MIF.ENVIRONMENT
2	CCB_RETCODE	1	DB	Command return (completion) code
3	CCB_CONFIG	1	DB	* CCB interface configuration data
4	CCB_POINTER	4	DD	Queue pointer and work area
8	CCB_CMD_CMPL	4	DD	Command completion user appendage
12	CCB_PARM_TAB	4	DD	Pointer to CCB parameter table
* Indicates a returned value (for example, a value set by the Local Area Network Support Program Interrupt Arbitrator).				

### CCB\_CONFIG

**Explanation:** The Local Area Network Support Program Interrupt Arbitrator passes the configuration data from the registered CCB interface, to the registering NCB interface in this field. The definition of the bits contained in this field are:

#### PC 3270 Control Program Supported

This bit indicates that the requesting interface supports the 3270 Control Program with the XMA adapter. When this bit is set the Local Area Network Support Program Interrupt Arbitrator activates logic that provides the required environment data to the registered interface via the interface defined in "PC 3270 Support" on page D-22.

**Bits 1-7** Reserved

## CCB Parameter Table structure

OFF-SET	FIELD NAME	BYTE LEN	8086 TYPE	DESCRIPTION
0	NCB.INPUT@	4	DD	Address of NCB input module
4	NCB.OPEN@	4	DD	Address of the NCB open module
8	NCB.CLOSE@	4	DD	Address of the NCB Close module
12	NCB.ENABLE@	4	DD	* Address of the interrupt enable module
* Indicates a returned value (for example, a value set by the Local Area Network Support Program Interrupt Arbitrator).				

## PARAMETERS

---

### NCB\_INPUT@

**Explanation:** This is the address to which the Local Area Network Support Program Interrupt Arbitrator passes control when it detects an NCB for the specified adapter. The Local Area Network Support Program Interrupt Arbitrator pushes all registers (except for AX) onto the stack and then transfers control to the specified address by a simulated INT instruction. The registers (except for AX) will be restored after the registered interface executes the IRET instruction to return control to the application program that issued the NCB.

The registers are pushed in the following order: BX, CX, DX, DI, SI, BP, ES, DS.

---

### NCB\_ENABLE@

**Explanation:** It should be noted that the value returned in this field is from the CCB interface registered for adapter 1 if multiple CCB interfaces are registered with the Local Area Network Support Program Interrupt Arbitrator.

## Additional Functions

**Monitored Control Blocks:** There are certain CCB\_COMMAND values that are adapter independent. These commands are intended for all currently active CCB interfaces. These CCB\_COMMAND values include:

- X'2B' - DIR.DEFINE.MIF.ENVIRONMENT
- X'24' - PDT.TRACE.ON
- X'25' - PDT.TRACE.OFF

The Local Area Network Support Program Interrupt Arbitrator monitors the CCB interface for the occurrence of one of these CCB\_COMMANDS. When one of these commands is detected the Local Area Network Support Program Interrupt Arbitrator propagates the CCB to all currently registered CCB interfaces. If the CCB\_CMD\_CMPL field is not zero, then the Local Area Network Support Program Interrupt Arbitrator gives control to the defined appendage after the CCB has been passed to all of the registered CCB interfaces.

**PC 3270 Support:** The Local Area Network Support Program Interrupt Arbitrator supports the PC 3270 Control Program including bank swapping with the XMA adapter via the following interface.

When a CCB or NCB interface registers with the PC 3270 Control Program Supported bit set (Bit 0 in the CCB\_IG field). The Local Area Network Support Program Interrupt Arbitrator will activate the following interfaces:

For CCB Interfaces:

- The AX register will be used to indicate the currently active bank to the registered CCB interface.
  - AX = X'FFFF' indicates that the 3270 Control Program is not loaded, or bank swapping is not active so all programs are in common memory.
  - AX = X'00FF' indicates that the 3270 Control Program is loaded and bank swapping is active. However the “CCB in common memory” bit (bit 3 in the CCB\_ADAPTER field) is set indicating that the control block is in common memory.
  - AX <> X'FFFF' or <> X'00FF' indicates that the 3270 Control Program is loaded and bank swapping is active. AX contains the current environment and bank ID. This information is in the same format that would be returned via an interrupt X'7A' function X'9D' request to the 3270 Control Program.

For NETBIOS Interfaces:

- The first two bytes of the NCB reserved space (offset X'32') will be used to indicate the currently active bank to the registered NETBIOS interface.
  - NCB offset X'32' = X'FFFF' indicates that the 3270 Control Program is not loaded, or bank swapping is not active so all programs are in common memory.
  - NCB offset X'32' <> X'FFFF' indicates that the 3270 Control Program is loaded and bank swapping is active. Offset X'32' in the NCB/NCB contains the current environment and bank ID. This information is in the same format that would be returned via an interrupt X'7A' function X'9D' request to the control program.



---

## Appendix E. Operating System/2 Extended Edition 1.1 Information

CONFIG.SYS COMMANDS .....	E-2
Adapter Parameters .....	E-3
DLC Parameters .....	E-6
NETBIOS 3.0 Parameters .....	E-9
The Operating System/2 Trace Facility .....	E-10
Trace Code Definition .....	E-11
Trace Entry Format .....	E-11
Trace Code Formats .....	E-12
Operating System/2 NETBIOS Trace Facility .....	E-15

---

## CONFIG.SYS Commands

The Operating System/2 System Initialization facility processes the configuration commands in the CONFIG.SYS file, establishing the final operating environment.

The CONFIG.SYS file contains commands used to configure the PC system. During PC system initialization, Operating System/2 opens and reads the CONFIG.SYS file and interprets the commands within the file. The adapter support software relies on the DEVICE command to install its device drivers (LANDD.SYS/TRNETDD.SYS/PCNETDD.SYS) and NETBIOS (NETBDD.SYS).

The DEVICE command is used to specify the path and filenames of the adapter support software's device drivers in order that they may be installed.

The keyword "cfg=" is provided on the DEVICE command line which is used to specify the path and filename of a configuration file. The configuration file is created by the configuration aid in Operating System/2 Extended Edition 1.1. If the parameter is not provided or the file is not found a set of defaults will be used as described in the following section.

There is also a trace level parameter passed to the adapter support software with the DEVICE command to enable OS/2 PC system tracing for the adapter support software (see -- Heading 'OS2TR' unknown -- for more information about the trace). The keyword t=n (where n can be 2,3, or 4) is provided on the DEVICE command line.

These CONFIG.SYS commands might appear as follows in the CONFIG.SYS file:

```
DEVICE=LANDD.SYS (Using NETBIOS or adapter support software Device Driver Interface)
DEVICE=TRNETDD.SYS CFG=pathname\filename.CFG t=n (Token-Ring Network)
DEVICE=PCNETDD.SYS CFG=pathname\filename.CFG t=n (PC Network)
DEVICE=NETBDD.SYS CFG=pathname\filename.CFG t=n
```

These command statements are only examples, see the *IBM Operating System/2 Extended Edition Version 1.1 Command Reference* for details on how these commands may be used.

**Note:** The DEVICE=LANDD.SYS command must appear first in the CONFIG.SYS file.

The following list summarizes the global parameters that define the user's environment for each adapter.

- OPEN\_OPTIONS (Not used for PC Network support)  
**Note:** Some of these options can only be set by the application program.
- NODE\_ADDRESS
- GROUP\_ADDRESS (Set only by the application program)<sup>00</sup>
- FUNCTIONAL\_ADDRESS (Set only by the application program)<sup>00</sup>

- NUMBER\_RCV\_BUFFERS
- RCV\_BUFFER\_LENGTH
- DHB\_BUFFER\_LENGTH
- DATA\_HOLD\_BUFFERS (Not used for PC Network support)
- PRODUCT\_ID
- SRAM\_ADDRESS (Not used for PC Network support)
- DLC\_MAX\_SAP
- DLC\_MAX\_STATIONS
- DLC\_MAX\_GSAP
- DLC\_MAX\_GSAP\_MEM
- DLC\_T1\_TICK\_ONE
- DLC\_T2\_TICK\_ONE
- DLC\_T1\_TICK\_ONE
- DLC\_T1\_TICK\_TWO
- DLC\_T2\_TICK\_TWO
- DLC\_T1\_TICK\_TWO
- SYSTEM\_KEY
- MAX\_USERS
- ADAPTER\_WORK\_SIZE (Used only for PC Network adapters)
- QSIZE
- GDTSIZE

None of the parameters are required, as defaults are used if the configuration file does not exist.

## Adapter Parameters

- OPEN\_OPTIONS (ignored by PC Network adapters): Various options used when the adapter is physically opened.
  - Argument specified as a 2 byte character string hex number, for example X'2000'
  - Character set supported (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
  - Optional - Default is X'2000'
  - Range - See option settings below.

Various options, each represented by a bit where the bit on (1) indicates that the option is to be taken. Bit 15 is the leftmost (high order) bit:



- OPEN OPTION - Bit 15: WRAP INTERFACE

The adapter will not insert into the ring. Instead, it will cause all user transmit data to appear as received data.

- OPEN OPTION - Bit 14: DISABLE HARD ERROR\*

If this option is taken, the Network Status HARD ERROR and TRANSMIT BEACON interrupts will not occur.

- OPEN OPTION - Bit 13: DISABLE SOFT ERROR\*

If this option is taken, the Network Status SOFT ERROR interrupt will not occur.

- OPEN OPTION - Bit 12: PASS ADAPTER MAC FRAMES\*

Pass Adapter Class MAC frames which are received but not supported by the adapter as direct interface data to the PC system.

If this option is not taken, frames in this class will be rejected.

- OPEN OPTION - Bit 11: PASS ATTENTION MAC FRAMES\*

All Attention MAC frames that are not the same as the last received Attention MAC frame will be passed as direct interface data to the PC system.

If this option is not taken, these frames will not be passed to the PC system.

- OPEN OPTION - Bits 9-10: Reserved. Should be set to 0, but not checked.

- OPEN OPTION - Bit 8: CONTENDER

The adapter will participate in Monitor Contention, if that possibility presents itself.

- OPEN OPTION - Bit 7: PASS BEACON FRAMES\*

The adapter will pass beacon frames up to the attached processor.

- OPEN OPTION - Bits 5-6: Reserved. Should be set to 0, but not checked.

- OPEN OPTION - Bit 4: TOKEN RELEASE

Setting this bit selects the non-default state of Token Release.

Adapter Data Rate	Default	Token Release
16Mbps	Early Token Release	Regular Token Release

- OPEN OPTION - Bits 0-3: Reserved. Should be set to 0, but not checked.

- NODE\_ADDRESS: This station's ring address

The 12 hexadecimal digit specific node address of this station on the LAN. The two high order bits of this value must be B'01'. If this value is coded as zero then the permanently encoded address will be used and if this value is coded to a non-zero value then the coded value will be used.

- Argument specified as a 6 byte character string hex number
- Optional - Default is permanently encoded address
- Recommended Range - X'400000000000' to X'400079999999'

- GROUP\_ADDRESS: Group address

- Argument specified as a 4 byte character string hex number

---

\* These open options are set by the application program, not by the OS/2 configuration.

- Optional - Default is X'00000000'
- Range - X'00000000' to X'FFFFFFF'

Group address for which the adapter will receive messages.

- **FUNCTIONAL\_ADDRESS:** Functional address

Functional address for which the adapter will receive messages.

Note that Bits 0 and 31 are ignored

- Argument specified as a 4 byte character string hex number
- Optional - Default is X'00000000'
- Range - X'00000000' to X'7FFFFFFE'

- **NUMBER\_RCV\_BUFFERS:** Number of receive buffers

The number of receive buffers required for the adapter support software to initialize. The adapter will configure all RAM remaining after other storage requirements have been met as receive buffers. If the number available is less than the number specified, then a message will be displayed to the user stating inadequate receive buffers for adapter to open: Initialization will fail and all commands will be rejected with CCB\_RETCODE set to X'30'. If the number available is greater than the number requested, no action will be taken.

- Argument specified as a character string decimal number
- Optional - Default is 8 if value is less than 2 or not specified
- Range - 2 to upper boundary, determined by the amount of memory available in shared RAM

- **RCV\_BUFFER\_LENGTH:** Length of receive buffers

The length of the adapter's receive buffers, which reside in the adapter's shared RAM. A receive buffer has eight bytes of overhead, for example when a buffer is defined as 112 bytes it can hold only 104 bytes of received data.

Receive buffers are allocated from the remaining shared RAM after all other requirements are satisfied.

- Argument specified as a character string decimal number
- Optional - Default is 112 if value is 0 or not specified
- Range - 96 to 2048 bytes in multiples of 8 bytes.

- **DHB\_BUFFER\_LENGTH:** Length of transmit buffers

The length of the adapter's transmit data hold buffers (DHB). When using Token-Ring Network adapters, the DHB resides in the adapter's shared RAM. When the PC Network adapters are used the DHB are located in PC system memory. A transmit buffer has six bytes of overhead, for example when a buffer is defined as 600 bytes it can hold only 594 bytes of transmit data (see TRANSMIT command).

- Argument specified as a character string decimal number
- Optional - Default is 600 if value is 0 or not specified
- Range - 96 to 17960 bytes in multiples of 8 bytes.

**Note:** It is the responsibility of the sending node to ensure that the size of transmitted frame does not exceed the receive buffer capacity of the receiving node.

- **DATA\_HOLD\_BUFFERS** (Ignored by PC Network adapters): Number of transmit buffers (DHBs)

This defines the number of adapter holding buffers which contain transmit data passed from the PC system to the adapter.

The adapter will accept any value 1-255, but the user should be aware that the integrity of adapter operation cannot be guaranteed if the value is greater than two. The control information for the first two DATA\_HOLD\_BUFFERS are located in read-only memory, thus application programs cannot write over this control information. However, once more than 2 buffers have been specified the control

information for the remaining buffers is no longer isolated from the application program's write operations; the control information is located in read/write memory. Requesting two DHBs may improve adapter performance by allowing a frame to be moved into the second DHB while the adapter is transmitting from the first. However, this will reduce the storage available for receive buffers.

- Argument specified as a character string decimal number
  - Optional - Default is 1 if value is not specified
  - Range - 1 to 255.
- **PRODUCT\_ID:** 18 byte product ID
    - Argument specified as a character string as described in Table 3-2 on page 3-41.  
If less than 18 characters are specified the product ID will be padded with blanks on the right. This character string is provided for RAS purposes and should describe the PC system attached to the network.
    - Character set supported - All upper case letters (A-Z) and decimal digits (0-9)
    - Optional - Default is 18 blanks if a value is not specified
  - **SRAM\_ADDRESS** (Not used for PC Network support): Defines the segment in PC system storage where the adapter's shared RAM is. This value should be on a boundary that is equal to the size of shared RAM, for example if shared RAM is 8K the SRAM\_ADDRESS should be on an 8K boundary or if shared RAM is 16K the SRAM\_ADDRESS should be on a 16K boundary. When using 16/4 Token-Ring Network adapters shared RAM, paging can be used, which would allow the use of a 16 KB boundary. The SRAM\_ADDRESS parameter is only used for a PC system with PC I/O Bus since there is no SETUP facility for these PC systems.
    - Argument specified as a 2 byte character string hex number, for example X'D000'
    - Optional - If value is 0 or not specified the defaults are:
      - Adapter 0 (primary) - Segment value X'D800'
      - Adapter 1 (secondary) - Segment value X'D400'
    - Range - Starting Address Range X'C00' to X'DE00' on the appropriate boundaries.

## DLC Parameters

The parameter ranges that represent quantities (ranges for DLC\_MAX\_SAP, DLC\_MAX\_STATIONS, DLC\_MAX\_GSAP and DLC\_MAX\_GSAP\_MEM parameters) are defined to ensure that values specified can be stored in the allotted memory sizes and do not represent the capabilities of the supported network adapters.

- **DLC\_MAX\_SAP:** Maximum number of SAPs  
Indicates the maximum number of SAP stations that may be opened.  
The maximum value of this parameter is 126. However, the maximum allowable value may be limited by the amount of available internal PC system RAM work area or by the amount of available Adapter Shared RAM.
  - Argument specified as a character string decimal number, for example 6.
  - Character set supported (0,1,2,3,4,5,6,7,8,9)
  - Optional - Default is 2 if value is not specified
  - Range - 0 to 126 (upper limit is 125 for PC Network support)
- **DLC\_MAX\_STATIONS:** Maximum number of link stations  
Indicates the maximum number of link stations that may be opened.  
The maximum allowable value may be limited by the amount of available internal PC system RAM work area or by the amount of available Adapter Shared RAM.

- Argument specified as a character string decimal number, for example 6.
  - Character set supported (0,1,2,3,4,5,6,7,8,9)
  - Optional - Default is 6 if value is not specified
  - Range - 0 to 255.
- **DLC\_MAX\_GSAP:** Maximum number of Group SAPs  
The maximum number of Group SAPs that may be opened at one time.  
If 0, no Group SAPs are allowed.
    - Argument specified as a character string decimal number, for example 2.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 0 if value is not specified
    - Range - 0 to 125 (upper limit is 124 for PC Network support)
  - **DLC\_MAX\_GSAP\_MEM:** Maximum members per Group SAP  
The maximum number of SAPs that may be members in any given Group.
    - Argument specified as a character string decimal number, for example 2.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 0 if value is not specified
    - Range - 0 to 126 (upper limit is 125 for PC Network support)
  - **DLC.T1\_TICK\_ONE:** Define DLC timer T1 interval, group one  
Indicates the number of 40 millisecond intervals between timer ticks for DLC timer T1, timer values 1-5.
    - Argument specified as a character string decimal number, for example 2.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 5 (200-400 milliseconds) if value is 0 or not specified
    - Range - 0 to 255.
  - **DLC\_T2\_TICK\_ONE:** Define DLC timer T2 interval, group one  
Indicates the number of 40 millisecond intervals between timer ticks for DLC timer T2, timer values 1-5.
    - Argument specified as a character string decimal number, for example 2.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 1 (40-80 milliseconds) if value is 0 or not specified
    - Range - 0 to 255.
  - **DLC\_TI\_TICK\_ONE:** Define DLC timer Ti interval, group one  
Indicates the number of 40 millisecond intervals between timer ticks for DLC timer Ti, timer values 1-5.
    - Argument specified as a character string decimal number, for example 20.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 25 (1-2 seconds) if value is 0 or not specified
    - Range - 0 to 255.
  - **DLC\_T1\_TICK\_TWO:** Define DLC timer T1 interval, group two  
Indicates the number of 40 millisecond intervals between timer ticks for DLC timer T1, timer values 6-10.
    - Argument specified as a character string decimal number, for example 20.
    - Character set supported (0,1,2,3,4,5,6,7,8,9)
    - Optional - Default is 25 (1-2 seconds) if value is 0 or not specified
    - Range - 0 to 255.
  - **DLC\_T2\_TICK\_TWO:** Define DLC timer T2 interval, group two  
Indicates the number of 40 millisecond intervals between timer ticks for DLC timer T2, timer values 6-10.

- Argument specified as a character string decimal number, for example 100.
- Character set supported (0,1,2,3,4,5,6,7,8,9)
- Optional - Default is 10 (400-800 milliseconds) if value is 0 or not specified
- Range - 0 to 255.

- **DLC\_TI\_TICK\_TWO:** Define DLC timer Ti interval, group two

Indicates the number of 40 millisecond intervals between timer ticks for DLC timer Ti, timer values 6-10.

- Argument specified as a character string decimal number, for example 100.
- Character set supported (0,1,2,3,4,5,6,7,8,9)
- Optional - Default is 125 (5-10 seconds) if value is 0 or not specified
- Range - 0 to 255.

- **SYSTEM\_KEY:** System key code. This key code is used to enable only a system administrator to perform operations that could stop ring communication for application programs.

This key code is used for the following:

- Change functional address
- Change group addresses
- Reset selected or all SAPs and all stations
- Relinquish ownership of direct stations
- Force a physical close for an adapter
- Force the adapter to initialize
- Read and reset adapter error and direct interface logs.

The SYSTEM\_KEY is not typically used by application programs, but for maintenance and RAS operations.

The SYSTEM\_KEY should be defined as listed below:

- Argument specified as a 2 byte character string hex number
- Range - X'0001' - X'FFFF'

- **MAX\_USERS:** Maximum number of users to concurrently use the Token-Ring Network adapter

This parameter provides a mechanism whereby the adapter support software can determine the size of memory needed for its work area. If requests issued to adapter support software continue to fail with CCB\_RETCODE set to X'59', then either the number of MAX\_USERS or the QSIZE parameter should be increased and the adapter support software restarted by re-IPLing the system.

- Argument specified as a character string decimal number
- Optional - Default is 3 if value is 0 or not specified
- Range - 0 to 5.

- **ADAPTER\_WORK\_SIZE:** This PC Network only parameter defines the size of memory allocated by the adapter support software during the initialization process. This memory is used to contain Link Station Control Blocks, SAP Control Blocks, Group SAP Control Blocks and receive buffers.

- Argument specified as a character string decimal number
- Optional - Default is X'2000'
- Range - X'2000' to X'FFFF'

- **QSIZE:** Number of queue elements

This parameter may be used to specify the number of internal queue elements that the adapter support software should allocate for all application programs. If no parameter is provided then the number of queue elements will be calculated using the number of users (MAX\_USER). The number of queue

elements allocated per user is 200. Thus, if 5 users are defined to be the maximum number of users at one time 1000 queue elements will be allocated.

- Argument specified as a character string decimal number
- Optional - Default is calculated using MAX\_USER.
- Range - 200 to 1400. The default is 200.

- **GDTSIZE:** Number of GDT selectors

This parameter may be used to specify the number of Global Descriptor Table (GDT) selectors that the adapter support software should allocate for all application programs. If no parameter is provided then the number of GDT selectors will be calculated using the number of users (MAX\_USER), but not to exceed 30. The number of GDT selectors allocated per user is 10. Thus, if 2 users are defined to be the maximum number of users at one time 20 GDT selectors will be allocated.

- Argument specified as a character string decimal number
- Optional - Default is calculated using MAX\_USER.
- Range - 10 to 30. The default is 20.

---

## NETBIOS 3.0 Parameters

NETBIOS version 3.0 is the NETBIOS interface executing under OS/2.

Table E-1. NETBIOS 3.0 System Default Values			
Load Parameters	Abbr.	Valid Values	Default
STATIONS	ST	0 - 254	32
SESSIONS	S	0 - 254	32
COMMANDS	C	0 - 255	32
NAMES	N	0 - 254	17
TRANSMIT.TIMEOUT	TT	0 - 20	1
TRANSMIT.COUNT	TC	0 - 10	6
DLC.MAXOUT	MO	0 - 9	2
DLC.MAXIN	MI	0 - 9	1
DLC.RETRY.COUNT	RC	0 - 255	*
DLC.T1	T1	0 - 10	5
DLC.T2	T2	0 - 11	2
DLC.TI	TI	0 - 10	3
RING.ACCESS	RA	0 - 3	0
DATAGRAM.MAX	DG	Y(yes)/N(no)	NO
REMOTE.NAME.DIRECTORY	RND	0 - 255	0
REMOTE.DATAGRAM.CONTROL	RDC	Y(yes)/N(no)	NO
ADAP.ADDR.NOT.REVERSED	ANR	Y(yes)/N(no)	NO
* The default values are determined by the adapter and the adapter interface code.			

---

## The Operating System/2 Trace Facility

The adapter support software utilizes the Operating System/2 System Trace facilities. In order for the adapter support software to use the Operating System/2 System Trace facilities, the application program must provide a keyword "t=" on the "DEVICE =" command line of the CONFIG.SYS file when defining the adapter support software device drivers. The keyword is defined below.

The trace keyword "t=" must appear on the "DEVICE =" command line of the CONFIG.SYS file when defining the adapter support software's device drivers.

```
DEVICE=TRNETDD.SYS CFG=pathname filename.CFG t=n
```

The value of n can be 2, 3, or 4.

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

<b>t=2</b>	A value of 2 indicates the adapter support software will trace the following codes. See "Trace Code Definition" on page E-11 for a definition of what each trace code means.
------------	--

X'00'  
X'02'  
X'08'  
X'09'  
X'0A'  
X'0D'

<b>t=3</b>	A value of 3 indicates the adapter support software will trace the following codes. See "Trace Code Definition" on page E-11 for a definition of what each trace code means.
------------	--

X'00'  
X'02'  
X'08'  
X'09'  
X'0A'  
X'0B'  
X'0C'  
X'0D'

<b>t=4</b>	A value of 4 indicates the adapter support software will trace the following codes. See "Trace Code Definition" on page E-11 for a definition of what each trace code means.
------------	--

X'00'  
X'01'  
X'02'  
X'03'  
X'04'  
X'05'  
X'06'  
X'07'  
X'08'  
X'09'  
X'0A'  
X'0B'  
X'0C'  
X'0D'

---

## Trace Code Definition

### Trace Entry Format

Each trace entry is sixteen bytes in length and each entry varies slightly depending upon the type of trace data (determined by the trace code).

The trace code in each trace entry identifies the point at which the trace entry was created and the format of the remainder of the trace entry. The Trace Codes are:

- X'00': CCB command received
- X'01': CCB command verified, queue element created
- X'02': CCB command error
- X'03': CCB command issued to the adapter (SRB) (not used by Token-Ring Network)
- X'04': CCB queue element added to receive "queue"
- X'05': CCB queue element added to read queue
- X'06': CCB queue element added to SRB wait queue
- X'07': CCB queue element added to SSB wait queue
- X'08': Completion of a CCB command or transmit command event occurred
- X'09': Completion of a received frame event occurred
- X'0A': Completion of a network status, DLC status, PC system action, or exception event occurred
- X'0B': Adapter interrupt received
- X'0C': Adapter timer expired
- X'0D': Adapter interrupt error detected



## Trace Code Formats

Byte:

0	1	2	3	4	5	6	7	8-9	10-13	14-15
Trace Code	Adapter Number (0/1)	Adapter Work Flag (76543210)	---	Command Code	RC / Correlator	Element Flag	Appl ID	Process ID	CCB Virt Addr	Timer Ticks Lo Word

Byte 2: Adapter Work Flags:

- Bit 7: Adapter initialized successfully
- Bit 6: Adapter initialization in progress
- Bit 5: Adapter initialization failure
- Bit 4: Adapter opened successfully
- Bit 3: Adapter open in progress
- Bit 2: Adapter open failure
- Bit 1: Adapter reset in progress
- Bit 0: Adapter interrupt in progress

Byte 6: Element Flag:

- Bit 7: Chain transmits/receives by SAP
- Bit 6: Chain transmits/receives by link station
- Bit 5: Purge the link station
- Bit 4: Purge the SAP
- Bit 3: Purge the application program
- Bit 2: Not used
- Bit 1: No active CCB corresponding to this CCB queue element
- Bit 0: Completion flag set for this completion event

Bytes 10-13: CCB Virtual Address:

Points to the corresponding application program CCB

Figure E-1. CCB Trace Entry (Trace Codes X'00' - X'07')

Byte:

0	1	2	3	4	5	6	7	8-9	10-13	14-15
Trace Code	Adapter Number (0/1)	Adapter Work Flag (76543210)	Event Code	Command Code	RC / Correlator	Element Flag	Appl ID	Process ID	CCB Virt Addr	Timer Ticks Lo Word

Byte 3: Event Code:

- Bit 7: Not used
- Bit 6: PC system action
- Bit 5: Network status
- Bit 4: Critical exception
- Bit 3: DLC status
- Bit 2: Receive
- Bit 1: Transmit
- Bit 0: CCB command complete

Figure E-2. CCB Completion Trace Entry (Trace Code X'08')

Byte:

0 1 2 3 4-5 6 7 8-9 10-13 14-15

Trace	Adapter	Adapter	Event	---	Element	Appl	Process	SAP	Timer
Code	Number	Work Flag	Code		Flag	ID	ID	Virt	Ticks
	(0/1)	76543210						Addr	Lo Word

Bytes 10-14: SAP Virtual Address:

Points to the corresponding received frame's first SAP buffer

Figure E-3. Receive Completion Trace Entry (Trace Code X'09')

Byte:

0 1 2 3 4 5 6 7 8-9 10-13 14-15

Trace	Adapter	Adapter	Event	Event	---	Element	Appl	Status	Status	Timer
Code	Number	Work Flag	Code	SubCode		Flag	ID		Data	Ticks
	(0/1)	76543210								Lo Word

Byte 4: Event SubCode:

For PC system action event code:

X'01': DIR.CLOSE.ADAPTER

X'02': DIR.INITIALIZE

X'03': DLC.RESET for a single SAP station

X'04': DLC.RESET for all SAP stations

For critical exception event code:

X'01': Network status

X'02': Adapter check

X'03': PC error

X'04': PC system action

Bytes 8-9: Status:

The two byte status code corresponding to the type of event  
(network status code, DLC status code, adapter check code, PC system error code)

Bytes 10-13: Status Data:

The first four bytes of the data corresponding to the type of event

Figure E-4. Status/Exception Completion Trace Entry (Trace Code X'0A')

Adapter Support Software Trace Entry:

Byte:

0      1            2            3   4   5    6-7   8-13 14-15

Trace	Adapter	Adapter	---	ISRP	ISRP	Timer	---	Timer
Code	Number	Work Flag		High	Low	Count		Ticks
	(0/1)	76543210						Lo Word

Byte 4: ISRP High:

- Bit 7: NMI disabled (always on)
- Bit 6: Interrupt enable (always on)
- Bit 5: Parity error in Shared RAM
- Bit 4: Timer expired (100 millisecond programmable timer)
- Bit 3: Adapter check
- Bit 2: Shared RAM access violation
- Bit 1: Always on
- Bit 0: Adapter number (0 or 1)

Byte 5: ISRP Low:

- Bit 7: IMPL received
- Bit 6: Adapter check
- Bit 5: SRB response
- Bit 4: ASB free
- Bit 3: ARB command
- Bit 2: SSB response
- Bit 1: Always 0
- Bit 0: Always 0

Bytes 6-7: Timer Count:

Number of timer interrupts from adapter 0 and 1

Adapter Support Software Trace Entry:

Byte:

0      1            2            3            4            5            6-9            10-13 14-15

Trace	Adapter	Adapter	Activation	Command	Return	Command	SS:SP	Timer
Code	Number	Work Flag	Record	Code	Code	specific		Ticks
	(0/1)	76543210	Reason code					Lo Word

Figure E-5. Interrupt Received Trace Entry (Trace Codes X'0B' - X'0C')

Byte:

0      1            2            3-13\*      14-15

Trace Code	Adapter Number (0/1)	Adapter Work Flag	SRB/SSB/ARB Work Area	Timer Ticks Lo Word
------------	----------------------	-------------------	-----------------------	---------------------

Bytes 3-13: SRB/SSB/ARB Work Area

First eleven bytes of the SRB/SSB/ARB that corresponds to the interrupt  
(The format of the eleven bytes depends upon the command, interrupt, ...)

\* These eleven bytes of work area are not used by the adapter support software.

Figure E-6. Interrupt Error Trace Entry (Trace Codes X'0D')

---

## Operating System/2 NETBIOS Trace Facility

The NETBIOS major trace code is X'A4'. There are two minor trace codes: X'20' indicates an Operating System/2 Device Driver Interface level trace entry and X'21' indicates an Operating System/2 Dynamic Link Routine Interface level trace entry.

Further, receipt of NETBIOS protocol message "remote trace off" is not supported by NETBIOS 3.0 (the message is ignored).

The specific trace data in the system is shown in the table below.

### Notes:

1. The basic information is similar to trace data in previous levels of NETBIOS. See the NCB.TRACE command in Chapter 4, "NETBIOS."
2. There is a new entry type: SYSTEM ACTION. This is indicated in the trace with the entry type and modifier X'EE04'.

Byte Offset	Usage
0	Adapter number indicator X'AA' = Adapter 0 X'BB' = Adapter 1
1	X'00' or an environment ID assigned by NETBIOS
2	Entry type
3	Entry modifier
4-7	Reserved
8-11	Type-dependent data
12-15	Address of data in bytes 16 through 31
16-31	Type-dependent data



---

## List of Abbreviations

<b>AC</b>	access control field	<b>LSB</b>	least significant bit
<b>A/C</b>	address recognized/frame copied	<b>LSN</b>	least significant nibble
<b>AIP</b>	adapter identification PROM	<b>MAC</b>	medium access control
<b>ARB</b>	adapter request block	<b>MMIO</b>	memory mapped input output
<b>ASB</b>	adapter status block	<b>MSB</b>	most significant bit
<b>CCB</b>	command control block	<b>MSN</b>	most significant nibble
<b>CSMA/CD</b>	carrier-sense multiple-access with collision detect	<b>NCB</b>	network control block
<b>DB</b>	define byte	<b>PDU</b>	protocol data unit
<b>DD</b>	define doubleword	<b>PIO</b>	programmed input output
<b>DHB</b>	data holding buffer	<b>POS</b>	program option selection
<b>DISC</b>	disconnect	<b>POST</b>	power-on self test
<b>DLC</b>	data link control	<b>PROM</b>	programmable read only memory
<b>DM</b>	disconnect mode	<b>PU</b>	protocol unit
<b>DMA</b>	direct memory access	<b>RAM</b>	random access memory
<b>DSAP</b>	destination service access point	<b>RPL</b>	Remote Program Load
<b>DW</b>	define word	<b>RRR</b>	RAM relocation register
<b>ED</b>	ending delimiter	<b>RR/RNR</b>	receiver ready/receiver not ready
<b>ETR</b>	early token release	<b>RSAP</b>	remote service access point
<b>FC</b>	frame control field	<b>SABME</b>	set asynchronous balanced mode extended
<b>FCS</b>	frame check sequence field	<b>SAP</b>	service access point
<b>FRMR</b>	frame reject	<b>SD</b>	starting delimiter
<b>FS</b>	frame status field	<b>SDLC</b>	synchronous data link control
<b>GDT</b>	global descriptor table	<b>SNA</b>	System Network Architecture
<b>GSAP</b>	group service access point	<b>SRAM</b>	shared random access memory
<b>I</b>	information (frame)	<b>SRB</b>	system request block
<b>IEEE</b>	Institute of Electrical and Electronic Engineers, Inc.	<b>SRPR</b>	shared RAM paging register
<b>I/O</b>	input/output	<b>SSB</b>	system status block
<b>IOPL</b>	input/output parameter list	<b>SSAP</b>	source service access point
<b>ISRA</b>	interrupt status register, adapter	<b>TCR</b>	timer control register
<b>ISRP</b>	interrupt status register, PC	<b>TVR</b>	timer value register
<b>LAN</b>	local area network	<b>UA</b>	unnumbered acknowledgment
<b>LDT</b>	local descriptor table	<b>UI</b>	unnumbered information
<b>LLC</b>	logical link control	<b>WRBR</b>	write region base management register
<b>LPDU</b>	logical link control protocol data unit	<b>WWCR</b>	write window close management register
<b>LSAP</b>	local service access point	<b>WWOR</b>	write window open management register
		<b>XID</b>	exchange identification



---

# Glossary

Definitions from draft proposals and working papers under development by the ISO/TC97 vocabulary subcommittee are identified by the symbol "(TC97)," indicating that final agreement has not yet been reached among its participating members.

## A

**access channel control.** The collection of logic and protocol machines that manages the transfer of data from the link stations to medium access control and from medium access control to the link stations.

**access control byte.** The byte following the start delimiter in a token or frame. It is used to control access to the ring.

**access priority.** The maximum priority that a token can have for the adapter to use it for transmission.

**access procedure.** The procedure or protocol used to gain access to a shared resource. In a local area network the shared resource is the medium. The medium access procedures specified by the IEEE 802 standard are CSMA/CD token, bus, and ring.

**active monitor.** A function in a single adapter on a ring network that initiates the transmission of tokens and provides token error recovery facilities. Any active adapter on a network has the ability to provide the active monitor function if the current active monitor fails.

**adapter.** In an IBM Network, the circuit card within a communicating device, and its associated software, that enable the device to communicate over the network.

**adapter support software.** The software used to operate adapters in a PC system and provide a common interface to application programs. In this document the term "adapter support software" also means any of the following:

- The adapter support software (TOKREUI.COM) provided with the original Token-Ring Network PC Adapter
- The adapter support software provided with the Local Area Network Support Program
- The adapter support software provided with the Communications Manager in the Operating System/2 Extended Edition 1.1.

**address segment.** A section of PC system memory that can be up to 64 Kilobytes.

**appendage.** An application program routine provided to assist in handling a specific occurrence.

**attach.** To connect a device logically to a ring. To participate in the data passing protocol of the ring.

**auto removal.** Removing a device from the data passing activity without human intervention. This action is accomplished by the adapter.

## B

**beacon.** A frame sent by an adapter indicating a serious network problem, such as a broken cable.

**beaconing.** To send beacon frames continuously.

**bridge.** A device that links one LAN segment to another LAN segment of the same type.

**buffer.** A memory area reserved for use in performing input/output operations.

**bypass.** To eliminate a component from a network by allowing the path to go past it.

## C

**checksum.** A value that when added with a sum calculated from accumulating all the bits in the field will result in zero.

**completion code.** The final return code provided by the adapter as a result of an issued command.

**CCB.** The Command Control Block as it would apply to all adapter support software interfaces supplied with the original Token-Ring Network PC Adapter, Token-Ring Network PC Adapter II, the Local Area Network Support Program, and Operating System/2 Extended Edition 1.1.

**CCB1.** The Command Control Block description applied specifically to the IEEE 802.2 adapter support software provided with the original Token-Ring Network PC Adapter, Token-Ring Network PC Adapter II, and the IEEE 802.2 adapter support software supplied with the Local Area Network Support Program.

**CCB2.** The Command Control Block description applied specifically to the Operating System/2 Dynamic Link Routine Interface (as supplied with Operating System/2 Extended Edition 1.1).

**CCB3.** The Command Control Block description applied specifically to the Operating System/2 Device



Driver Interface (as supplied with Operating System/2 Extended Edition 1.1).

**control block.** In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the adapter support software to request an operation.

**cyclic redundancy check.** A numeric value derived from the bits in a message that is used to check for any bit errors in transmission.

## D

**data frame.** See frame.

**datagram.** A particular type of information encapsulation at the network layer of the adapter protocol for NETBIOS. No explicit acknowledgment for the information is sent by the receiver. Instead, transmission relies on the "best effort" of the link layer.

**data link.** Any physical link, such as a wire or a telephone circuit, that connects one or more devices or communication controllers.

**data link layer (or level).** (TC97) In open systems architecture, the layer that provides the functions and procedures used to establish, maintain, and release data link connections between elements of the network.

**data rate.** On a local area network, the data rate is the frequency of the signal being transmitted.

**delimiter.** In the IBM Token-Ring Network, a bit pattern that defines the limits of a frame or token on a ring.

**device.** An input/output unit such as a terminal, display, or printer.

**distribution panel.** A wiring board that provides a patch panel function and mounts in a rack.

**downstream.** On a network, the direction of data flow.

## E

**early token release.** Early token release allows a transmitting station to release a new token as soon as it has completed frame transmission, whether or not the frame header has returned to that station.

## F

**faceplate.** A plate for mounting cable data and telephone jack connectors. It comes in two varieties, wall-mounted and surface-mounted.

**frame.** The unit of transmission in the Token-Ring Network. It includes delimiters, control characters, information, and checking characters. A frame is created when a token has data appended to it by a node.

## H

**hard error.** A serious error on the network that requires that the network be reconfigured or that the source of the error be removed before the network can resume reliable operation.

## I

**idles.** Signals sent along a network when neither frames nor tokens are being transmitted.

**initialize.** In the IBM Token-Ring Network, to prepare the adapter (and the adapter support software if used) for use by an application program.

## L

**latency.** 1. The time interval between the instant at which an instruction control unit initiates a call for data and the instant at which the actual transfer of data begins. Synonymous with waiting time. 2. See also ring latency.

**link.** In the IBM Token-Ring Network, the logical connection between nodes including the end-to-end link control procedures.

**link connection.** All physical components and protocol machines that lie between the communicating link stations of a link. The link connection may include a switched or leased physical data circuit, a local area network, or an X.25 virtual circuit.

**link station.** A protocol machine in a node that manages the elements of procedure required for the exchange of data traffic with another communicating link station.

**lobe.** The section of cable that connects a device to an access unit.

**lobe receptacle.** An outlet on a wiring concentrator for connecting a lobe.

**local area network.** A network in which communications are limited to a moderate-sized

geographic area such as a single office building, warehouse, or campus, and which do not extend across public rights-of-way.

**local busy.** A state that may occur for a given link station during which I-frame reception is suspended. May occur because of (1) application program request, (2) lack of buffers in the SAP buffer pool.

**Logical Link Control Protocol Data Unit (LPDU).** The unit of information exchanged between link stations in different nodes. The LPDU consists of the DSAP and SSAP address fields, the control field, and the information field (if present).

## M

**MAC frame.** Frames that control the operation of the IBM Token-Ring Network and any ring station operations that affect the ring.

**medium.** A physical carrier of electrical energy.

**medium access control.** The sub-component of DLC that supports medium-dependent functions and uses the services of the physical layer to provide services to logical link control. MAC includes the medium access port.

**memory mapped I/O (MMIO).** In an IBM PC, a method of accessing an input or output port as if it were a memory location.

**monitor.** In the IBM Token-Ring Network, the function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

See active monitor.

## N

**NAUN.** Nearest active upstream neighbor. For any station on a ring, the station that is sending frames or tokens directly to it.

**NETBIOS.** The IBM NETBIOS API provides a programming interface to the LAN so that an application program can have LAN communication without knowledge and responsibility of the Data Link Control (DLC).

**network.** (1)(TC97) An interconnected group of nodes. (2) The assembly of equipment through which connections are made between data stations.

**network administrator.** A person who manages the use and maintenance of a network.

**network management.** The conceptual control element of a station that interfaces with all of the layers of that station and is responsible for the resetting and setting of control parameters, obtaining reports of error conditions, and determining if the station should be connected to or disconnected from the medium.

**network status.** The condition of the network.

**node.** An endpoint of a link or a junction common to two or more links in a network.

**node address.** The address of an adapter on the ring.

**numbered frames.** Information segments arranged in numbered order for accountability.

## O

**open.** (1) To make an adapter ready for use. (2) A break in an electrical circuit.

## P

**patch cable.** In the IBM Token-Ring Network, a cable with IBM Cabling System connectors at each end, used to connect together components, devices, and cables of the network.

**path.** In a network, a route between any two nodes.

**PC system.** A term used specifically in this book to refer to all IBM Personal Computers and all IBM Personal System/2s.

**programmed I/O (PIO).** In an IBM PC, a method of accessing an input or output port with specific instructions.

**post.** To affix to a usual place. Used with NETBIOS to indicate providing items such as a return code at the end of a command or function. Also used to define an appendage routine.

## R

**receive.** To obtain a message transmitted from a device.

**remove.** To stop an adapter from participating in data passing on the network.

**return code.** A hexadecimal value provided by the adapter to indicate the result of an action.

**ring (network).** A network configuration consisting of a series of attaching devices connected by unidirectional transmission links to form a closed path.

**Ring Diagnostic.** In the IBM Token-Ring Network, a program to be run in PC systems that allows the user to monitor ring activity.

**ring latency.** In a Token-Ring Network, the time, measured in bit times at the data transmission rate, required for a signal to propagate once around the ring. Ring latency includes the signal propagation delay through the ring medium, including drop cables, plus the sum of propagation delays through each data station connected to the Token-Ring Network.

**ring in.** On an access unit, the receive or input receptacle.

**ring out.** On an access unit, the transmit or output receptacle.

**ring sequence.** The order in which components are connected to a ring.

**ring topology.** A logically circular, unidirectional transmission path without defined ends. Control can be distributed or centralized.

**routing.** The assignment of the path by which a message will reach its destination.

## S

**SABME.** Set Asynchronous Balanced Mode Extended. An LLC command used in the establishing of a link.

**segment.** A portion of a network. It may contain cables, components, or lobes.

See address segment.

**service access point.** The logical point at which an  $n+1$ -layer entity acquires the services of the  $n$ -layer. In this document, the layer is assumed to be DLC. A single SAP can have many links terminating in it. These link "end-points" are represented in DLC.LAN by link stations.

**session.** The data transport connection resulting from a call or link between two devices.

**soft error.** An error on a network that can impair the network's performance but does not, by itself, affect its reliability.

**start delimiter.** The first byte of a token or frame. It consists of a special recognizable bit arrangement.

**star wiring.** A wiring arrangement in which an individual cable runs from each work area to a concentration point.

## T

**terminal.** A device or work station.

**token.** A sequence of bits passed from station to station along the network. It consists of a starting delimiter, a frame control field, and an ending delimiter. The frame control field contains a token indicator bit that indicates to a receiving station that the token is ready to accept information. If the station has data to send along the network, it appends the data to the token. The token then becomes a *frame*.

**token ring.** A network with a ring topology that passes tokens from station to station.

**transmission medium.** A physical carrier of electrical energy.

**transmit.** To send information from one place for reception elsewhere.

## U

**unnumbered acknowledgment.** A DLC command used in establishing a link and in answering receipt of LLC frames.

**upstream.** On a network, the direction opposite that of data flow.

## W

**wire fault.** An error condition caused by a break in the wires or a short between the wires in a segment of cable.

**wiring closet.** A room that contains one or more distribution racks and panels that are used to connect various cables together (via patch cables) to form physical networks.

**work area.** An area in which terminal devices (such as displays, keyboards, and printers) are located.

**work station.** An input/output device that allows either the transmission of data or the reception of data (or both) from a host system, as needed to perform a job; for example, a display station or printer.

# Index

## A

- access control field 1-10
  - for all CCBs B-44
  - for CCB1 B-40
  - for CCB2 B-40
  - for CCB3 B-42
  - reason codes B-43
- adapter
  - operating 7-26
  - registers 7-10
- adapter ID PROM (AIP) 7-20
  - contents 7-20
- adapter identification area
- adapter interface
  - overview 7-3
- adapter registers 7-10
- adapter request block (ARB) 7-26, 7-68
  - commands 7-58
- adapter status block (ASB) 7-27
- adapter support software 1-10, 2-2
- adapter-to-PC system commands 7-58
- address 2-23
  - functional 2-23, 7-28
  - group 2-23, 7-28
  - node 2-23, 7-28
  - ring 2-23
- addressing 2-23
- Add\_Group\_Name\_Query 5-11
- Add\_Name\_Query 5-12
- Add\_Name\_Response 5-20
- APPC/PC Program 1-9
- appendage 2-4, 2-19, 4-6, B-25
  - exception 2-4
  - receive data 2-4
- application calls B-46
- attachment control area 7-11
  - communication 7-11
  - registers 7-11
  - summary 7-19

## B

- BIOS
  - program storage 7-10
- bit notation 1-12
- bridge commands 7-65
  - initiating transmission 7-71
  - transmitting forwarded frames 7-70
- bridge functions 7-64
  - transmit buffer layout 7-72
- bring-up error codes
  - for all CCBs B-48

- bring-up errors
  - for all CCBs B-48
- broadcast 5-4
  - general 5-4
  - single-route 5-4
- buffers 2-35
  - pools 2-35
  - receive 2-36
  - transmit 2-41
- BUFFER.FREE 3-3
- BUFFER.GET 3-4

## C

- cable 8-10
- cabling system 8-10
- call a name: name found - start session 5-58
- call a name: name not in RND but is on network - start session 5-56
- call a name: name not on network or RND 5-55
- CCB return codes
  - cause and action B-12
- CCB1 commands
  - return codes B-4
- CCB2 commands
  - return codes B-6
- CCB3 commands
  - return codes B-9
- coding parameters. D-7
- command 3-2
  - BIOS/MMIO 7-9
  - direct 7-30
  - directory A-1
  - no-wait 4-3
  - PC, to the 7-58
  - PIO 7-6
  - reserved A-1
  - sequences, DLC 2-27
  - wait 4-3, A-1
- command control block for DOS
  - CCB1 2-6
- command control block for Operating System/2 Extended Edition 1.1, Device Driver Interface
  - CCB3 2-16
- command control block for Operating System/2 Extended Edition 1.1, Dynamic Link Routine Interface
  - CCB2 2-11
- command control block (CCB) 2-17, 3-2
  - return codes B-1
- completion code 2-3, 2-18, 4-4
- CONFIG.SYS commands E-2
  - adapter parameters E-3
- control block 1-8, 2-2, 2-17, 3-2, 7-30

## D

- data holding buffer (DHB) 7-32
- data transfer 7-4
- Datagram 4-2, 5-16
- Datagram\_Broadcast 5-17
- Data\_Ack 5-24
- Data\_First\_Middle 5-25
- Data\_Only\_Last 5-27
- destination SAP (DSAP) 2-25
- direct interface 1-8, 3-2
  - commands 7-30
- DIR.CANCEL.TIMER.GROUP 3-58
- DIR.CLOSE.ADAPTER 3-25, 7-30
- DIR.CLOSE.DIRECT 3-27
- DIR.CONFIG.BRIDGE.RAM 7-66
- DIR.DEFINE.MIF.ENVIRONMENT 3-27
- DIR.INITIALIZE 3-29
- DIR.INTERRUPT 3-33, 7-31
- DIR.MODIFY 7-32
- DIR.MODIFY.OPEN.PARMS 3-33
- DIR.OPEN.ADAPTER 3-34, 7-32
- DIR.OPEN.DIRECT 3-46
- DIR.READ.LOG 3-47, 7-37
- DIR.RESTORE.OPEN.PARMS 3-50, 7-38
- DIR.SET.EXCEPTION.FLAGS 3-50
- DIR.SET.FUNCT.ADDRESS 3-52, 7-39
- DIR.SET.GROUP.ADDRESS 3-52, 7-39
- DIR.SET.USER.APPENDAGE 3-53
- DIR.STATUS 3-54
- DIR.TIMER.CANCEL 3-57
- DIR.TIMER.SET 3-59
- DISC 2-29, B-25, B-26
- diskette C-1
- DLC interface 1-8
- DLC parameters E-6
- DLC SRB commands 7-40
- DLC status 2-29, B-25
- DLC status codes B-25
- DLC.CLOSE.SAP 3-5, 7-41
- DLC.CLOSE.STATION 3-6, 7-42
- DLC.CONNECT.STATION 3-7, 7-43
- DLC.FLOW.CONTROL 3-8, 7-44
- DLC.MODIFY 3-9, 7-45
- DLC.OPEN.SAP 3-11, 7-47
- DLC.OPEN.STATION 3-16, 7-50
- DLC.REALLOCATE 7-52
- DLC.RESET 3-20, 7-53
- DLC.STATISTICS 3-23, 7-54
- DLC.STATUS 7-59
- DOS 2-2

## E

- end delimiter 1-10
- errors
  - adapter check B-40
  - adapter open B-49

## errors (continued)

- bring-up B-48
  - PC System Detected B-53
  - PC System Detected for CCB2 B-54
  - PC System Detected for CCB3 B-56
- ring B-39, B-44
- exception indications B-40

## F

- frame 1-11
- frame check sequence field 1-11
- frame status B-39
- frame type 1-12
- FRMR 2-29, B-25, B-26
- functional address 2-23, 7-28

## G

- group address 2-23, 7-28
- group SAP (GSAP) 2-24

## H

- header format overview 5-6

## I

- IEEE 1-8
- initialization 2-27, 7-26
- interface registration CCB D-18
- interfaces 1-7
  - direct 3-2
  - NETBIOS 4-2
- interrupt 2-2, 7-5
  - hardware 2-3
  - level selection 7-8
  - register pairs 7-13
  - software 2-3
  - status registers 7-10
  - status (adapter pair) 7-13
  - status (PC system pair) 7-14
  - X'27' 2-3
  - X'5C' 2-3

## L

- LAN header 1-11
- Language Request CCB D-18
- layer
  - physical 4-2
  - session 4-2
- link states
  - primary 2-29
  - secondary 2-29
- Local Area Network Support Program D-2
  - device drivers D-6
  - using two LAN adapters D-3

Local Area Network Support Program Interrupt  
Arbitrator D-16

## M

Medium Access Control (MAC) 1-12  
MMIO 7-7, 7-9  
    control area 7-10  
    domain 7-9.  
    layout 7-10  
monitored control blocks D-22

## N

Name Management examples 5-38  
    add a name to the network 5-38  
    add a name: name already on network 5-39  
    add a name: receive multiple responses 5-40  
names 4-2  
    group 4-2  
    table 4-2  
Name\_In\_Conflict 5-13  
Name\_Query 5-18  
Name\_Recognized 5-21  
NCB handler registration D-21  
NCB.ADD.GROUP.NAME 4-14  
NCB.ADD.NAME 4-14  
NCB.CALL 4-15  
NCB.CANCEL 4-16  
NCB.CHAIN.SEND 4-17  
NCB.CHAIN.SEND.NO.ACK 4-18  
NCB.DELETE.NAME 4-19  
NCB.FIND.NAME 4-20  
NCB.HANG.UP 4-21  
NCB.LAN.STATUS.ALERT 4-22  
NCB.LISTEN 4-22  
NCB.RECEIVE 4-23  
NCB.RECEIVE.ANY 4-24  
NCB.RECEIVE.BROADCAST.DATAGRAM 4-25  
NCB.RECEIVE.DATAGRAM 4-26  
NCB.SEND 4-30  
NCB.SEND.BROADCAST.DATAGRAM 4-31  
NCB.SEND.DATAGRAM 4-32  
NCB.SEND.NO.ACK 4-32  
NCB.SESSION.STATUS 4-33  
NCB.STATUS 4-35  
NCB.TRACE 4-37  
NCB.UNLINK 4-41  
NETBIOS 1-9, 4-2, D-4  
    commands 5-5  
    frames protocol overview 5-1  
    remote name directory (RND) 5-5  
    return codes B-27  
NETBIOS commands 5-5  
NETBIOS device driver parameters D-10  
    device driver keyword new parameter  
    summary D-11  
    device driver keyword old parameter  
    summary D-10

NETBIOS device driver parameters (*continued*)  
    device driver keyword parameter  
    descriptions D-12

NETBIOS Frame Formats 5-11, 5-18

    Add\_Group\_Name\_Query 5-11, 5-12  
    Add\_Name\_Response 5-20  
    Datagram 5-16  
    Datagram\_Broadcast 5-17  
    Data\_Ack 5-24  
    Data\_First\_Middle 5-25  
    Data\_Only\_Last 5-27  
    Name\_In\_Conflict 5-13, 5-14  
    Name\_Recognized 5-21  
    No\_Receive 5-33  
    Receive\_Continue 5-35  
    Receive\_Outstanding 5-34  
    Session\_Alive 5-36  
    Session\_Confirm 5-29  
    Session\_End 5-30  
    Session\_Initialize 5-31  
    Status\_Query  
    Status\_Response 5-22  
    Terminate\_Trace 5-16, 5-23  
NETBIOS frame header fields 5-7  
NETBIOS frame summary 5-8  
    listed alphabetically 5-8  
    listed by function 5-9  
    listed by transmission type 5-10  
    listed numerically 5-8  
NETBIOS protocol examples with RND 5-51  
NETBIOS protocol examples without RND 5-37  
NETBIOS protocol SEND.NO.ACK examples 5-57  
NETBIOS 3.0 parameters E-9  
network control block (NCB) 2-17, 4-3  
    field descriptions 4-3  
network status B-44  
    for CCB1 B-44  
    for CCB2 B-44  
    for CCB3 B-46  
non-MAC frames 1-12  
No\_Receive 5-33

## O

open 1-10, 2-27  
open errors  
    for all CCBs B-49, B-53  
    recommended actions B-50  
Operating System/2  
    general information E-1  
    trace keyword E-10  
Operating System/2 NETBIOS Trace facility E-15  
Operating System/2 Trace facility  
overview of Communications Manager 1-3  
overview of OS/2 Extended Edition 1-3

## P

- parameters 2-17, 2-20, 4-8, 7-28
  - DLC status table B-38
  - selecting values 2-32
- PC BIOS D-3
- PC Network
  - open errors for all CCBs B-53
  - reason codes for all CCBs B-44
  - status codes for all CCBs B-48
- PC Network device driver parameters D-9
  - adapter addresses D-9
  - coding parameters D-9
  - workspace D-9
- PC registers 2-2, 2-4, B-25
- PC system-to-adapter
  - commands 7-29
- PDT.TRACE.OFF 3-67
- PDT.TRACE.ON 3-59
- physical trailer field 1-11
- pins 8-2, 8-6, 8-10
- problem determination
  - bring-up errors B-48
  - DLC status B-25
  - exception conditions B-40
  - network status B-44
  - PC System Detected errors B-53
  - PC System Detected errors for CCB2 B-54
  - PC System Detected errors for CCB3 B-56
- product ID
- program option selection (POS) 7-75
- programmed I/O 7-6
  - commands 7-5
- protocol 3-2

## R

- read-only memory 7-13
- receive 2-28, 3-77
- RECEIVED.BRIDGE.DATA 7-69
- RECEIVE.CANCEL 3-83
- RECEIVE.DATA 7-60
- RECEIVE.MODIFY 3-84
- Receive\_Continue 5-35
- Receive\_Outstanding 5-34
- registration process D-17
- relocation register 7-11
- remote adapter status examples 5-41
  - remote adapter status data: segmentation 5-43
  - remote adapter status for a name not on network 5-41
  - remote adapter status for a name on network 5-42
- remote adapter status for a name not in RND but on the network 5-53, 5-54
- remote adapter status for a name not on network or RND 5-52
- remote name directory
  - function 5-5

- remote name directory (RND)
- return code 2-3, 2-18, 4-4
  - adapter check B-43
  - bring-up B-48
  - CCB\_RETCODE B-2
  - Direct B-2
  - DLC B-2
  - exception B-40
  - NCB\_RETCODE B-25
  - open errors B-49
- ring 1-10
- ring diagnostic
- ring traffic 1-10
- RING.STATUS.CHANGE 7-62

## S

- SABME 2-29, 3-7, B-25, B-26
- sample program C-1
- SAP
  - global 2-25
  - null 2-25
  - VALUE 2-24
- send session data: one SEND.NO.ACK and multiple receive 5-60
- send session data: one SEND.NO.ACK and no receive 5-59
- send session data: one SEND.NO.ACK and one receive 5-60
- session 4-11
- session data transfer examples 5-47, 5-59
  - send session data: one send and multiple receives 5-48
  - send session data: one send and one receive 5-47
  - send session data: one SEND.NO.ACK and multiple receives 5-60
  - send session data: one SEND.NO.ACK and no receive 5-59
  - send session data: one SEND.NO.ACK and one receive 5-60
  - send session data: segmentation and multiple receives 5-50
  - send session data: segmentation and one receive 5-49
- session establishment examples 5-44, 5-58
  - call a name: name found - start session 5-46, 5-58
  - call a name: name not on network 5-44
  - call a name: name on network but no listen 5-45
- session establishment RND examples 5-55
  - call a name: name not in RND but is on network - start session 5-56
  - call a name: name not on network or RND 5-55
- Session\_Alive 5-36
- Session\_Confirm 5-29
- Session\_End 5-30
- Session\_Initialize 5-31

- setup 7-7
- shared RAM 7-25, 7-26, 7-64
  - domain 7-8
  - layout 7-23, 7-65
  - read-only area 7-13
  - relocation register 7-11
  - write management register pairs 7-12
- shared RAM paging
- signal 8-10
  - on pins 8-2, 8-6
- source SAP (SSAP)
- star wired 1-2
- start delimiter 1-10
- station
  - ID 2-24
  - types 2-24
- station ID 2-24
- Status\_Response 5-22
- system action exceptions
  - for CCB2 B-58
  - for CCB3 B-60
- system request block (SRB) 7-26
  - commands 7-30
- system status block (SSB) 7-26

## T

- Terminate\_Trace 5-16, 5-23
- timer control register (TCR) 7-15
- timer value register (TVR) 7-16
- TIMERINT.SYS D-3
- timers
  - programmable 3-57, 3-58
  - rates 2-31
  - Ti inactivity 2-31
  - T1 response 2-31
  - T2 acknowledgment 2-31
- Token Ring Network device driver parameters
  - adapter addresses D-6
  - shared RAM D-7
- token-ring 1-2, D-2
- Token-Ring Network D-6
  - adapter interface 7-3
  - additional functions 7-64
  - AIP 7-20
  - bridge functions 7-64
  - open errors for all CCBs B-49
  - status codes for all CCBs B-47
- trace code definition E-10
  - trace code formats E-12
  - trace entry format E-11
- transmit 2-28, 2-40
- transmit commands 7-56
  - command summary 7-57
- TRANSMIT.DATA.REQUEST 7-62
- TRANSMIT.DIR.FRAME 3-88
- TRANSMIT.I.FRAME 3-88

- TRANSMIT.TEST.CMD 3-89
- TRANSMIT.UI.FRAME 3-90
- TRANSMIT.XID.CMD 3-90
- TRANSMIT.XID.RESP.FINAL 3-90
- TRANSMIT.XID.RESP.NOT.FINAL 3-91, 7-67
- TTL loads 8-4, 8-6
- type 1 communication 1-9
- type 2 communication 1-9

## U

- UA 2-29, 3-7, 3-22, 3-69, 3-77
- using PC Network adapters D-4

## V

- variables 2-32
- voltages 8-2, 8-6

## W

- write management relocation pair 7-12

## Numerics

- 16/4 adapter D-2





**Publication No. SC30-3383-2**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**Note:** Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity      Accuracy      Completeness      Organization      Coding      Retrieval      Legibility

If you wish a reply, give your name, company, mailing address, date, and location of your local IBM branch office:

---

---

---

---

---

What is your occupation? \_\_\_\_\_

Number of latest Newsletter associated with this publication: \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

**Reader's Comment Form**

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Dept. E02  
P.O. Box 12195  
Research Triangle Park, N.C. 27709-9990



Fold and tape

Please Do Not Staple

Fold and tape

