IBM

Series/1

SL23-0105-0

EDX Communications Facility Operator's Guide

Version 2

Introduction	Design and Installation Guide	Operator's Guide
Master Index and Glossary	Programmer's Guide	Messages and Codes
WSC High-Level Language Subroutines Programmer's Guide	Debugging Guide	Operator's Reference Summary

IBM

Series/1

SL23-0105-0

EDX Communications Facility Operator's Guide

Version 2

Introduction	Design and Installation Guide	Operator's Guide
Master Index and Glossary	Programmer's Guide	Messages and Codes
WSC High-Level Language Subroutines Programmer's Guide	Debugging Guide	Operator's Reference Summary

First Edition (September 1984)

This edition applies to Version 2.0 of the Licensed Program IBM Series/1 Event Driven Executive Communications Facility, Program Number 5719-CF2, and to all subsequent versions and modifications of this program unless otherwise indicated in new editions or technical newsletters.

Information in this publication is subject to significant change. Any such changes will be reported in new editions or technical newsletters. Use this publication only for the purposes mentioned under "About This Book" on page iii.

It is possible that this material may contain references 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.

Direct all requests for copies of IBM publications to your IBM representative or to the IBM branch office that serves your locality.

Any names of individuals, companies, brands, and products in the examples in this book are fictitious; any similarity to the name of an actual business enterprise is entirely coincidental.

This publication may contain technical or typographical errors. Forms for readers' comments are provided at the back of the book. If the forms have been removed, address your comments to IBM Corporation, System Products Division, 4J1/037, PAS5 MS33, 1501 California Avenue, PO Box 10500, Palo Alto, CA 94304. IBM may use and 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 book is intended for a person responsible for operating an IBM Series/1 Event Driven Executive Communications Facility system.

It is intended to give you the information you need to operate the Communications Facility using operator commands, IBM-supplied transactions, and utility programs. To that end, it has these chapters:

- "Communications Facility Overview" on page 1 gives a general explanation of the parts of the Communications Facility you'll be using to operate the system.

- "Operating the Communications Facility" on page 13 describes the functions you will perform when operating a Communications Facility system. It explains which commands, transactions, and utility programs you will use to perform these functions and includes examples of their use.

- "Operator Command Reference" on page 31 explains the format and parameters to be specified for each operator command and includes examples of them.

- "IBM-Supplied Transactions Reference" on page 231 explains the format and parameters to be specified for each IBM-supplied transaction and includes examples of them.

- "Utility Programs" on page 263 explains what each utility does and shows you how to use it.

- Appendix A, "3270 Polling and Selection Address Table" on page 307 explains how to determine the polling and selection addresses for 3270 terminals.

- The "Glossary" presents the definitions of technical Communications Facility terms and acronyms. For EDX definitions, see the appropriate book.

- The "Index" is a conventional index to the publication.

Terminal as used is this book is a point in a system or network at which data can either enter or leave. A terminal can be a printer or a display device.

X.25 network is used to mean a packet-switching network as defined by the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.25.

The companion book to this one is the *IBM Series/1 EDX Communications Facility Messages and Codes,* SL23-0120.

For quick reference to the information presented in this book, use the *IBM Series/1 EDX Communications Facility Operator's Reference Summary*, SX23-0108.

Other Communications Facility books you might need are the *IBM Series/1 EDX:*

- *Introduction*, GL23-0103, for introductory information about the Communications Facility

- *Design and Installation Guide*, SL23-0104, for detailed information about the structure of a Communications Facility system

- *Debugging Guide*, LL23-0109, for problem determination techniques and control block formats

- *Programmer's Guide*, SL23-0106, for information about the Communications Facility Event Driven Language instructions.

You might want to refer to information in the *IBM Series/1 Event Driven Executive:*

- *Communications Guide*, SC34-0443, for additional information about remote functions

- *Operation Guide*, SC34-0437, and the

- *Operator Commands and Utilities Reference*, SC34-0444, for a working knowledge of the EDX operating system

- *Messages and Codes*, SC34-0445, if you want information about EDX error conditions reported in Communications Facility log messages

- *X.25/HDLC Communications Support Programming and Operating Reference Manual,* SC09-1030, referred to in this book as the XHCS *Programming and Operating Reference.*

Summary of Amendments

Only changes affecting the content of this book are listed in the Summary of Amendments.

Version 2.1 PTF 2

The following changes were made between Communications Facility Version 2.1 PTF 1 and Version 2.1 PTF 2:

Retry Support: A new command, "CP F RETRY—Change Control Unit Station's Retry Maximum" on page 120.3, was added. It allows an operator to specify the maximum number of times a device station can time out.

Cold Start Option: This option was added to "CP S—Start Station" on page 175. It allows a station to receive new messages after discarding the station's disk-queued messages.

Other Changes: The following commands were changed:

- "CP F PU—Change Controlling SNA Physical Unit Station" on page 117
- "CP HELP—Display CP Commands" on page 139.

Version 2.1 PTF 1

The following changes were made between Communications Facility Version 2.1 and Version 2.1 PTF 1:

EDX Secondary SNA2 Support: A new command, "CP F PUNUM—Change SNA Physical Unit Number" on page 120.1, was added and the description of the following commands were changed:

- "CP DEF PU—Define SNA Physical Unit Station" on page 67
- "CP Q NET—Display $.SYSNET Station Definitions" on page 153
- "CP Q PARM—Display System Parameters of Active Station" on page 159
- "CP Q SNA—Display EDX-SNA PU and LU Status" on page 163.

Other Changes: The following commands were changed:

- "CP DEF DEVICE 3101F—Define 3101F Device Station" on page 47
- "CP F MODE—Change Mode of Operation" on page 99
- "CP F POLL—Change Polling Address" on page 111
- "CP F PU—Change Controlling SNA Physical Unit Station" on page 117
- "CP P—Stop Station" on page 143
- "PD M—Send Message" on page 215.

Version 2.1

The following changes were made between the Communications Facility Version 2.0 and Version 2.1:

Series/1-PC Connect Support: A description of the new station, volume station, was added to the section "Stations" on page 5.

A description of the emulated Personal Computer (PC) disk management utility program, $.DSFORM, was added to the section "Utility Programs" on page 283.

The following commands were added:

- "CP DEF LINE PCC—Define Series/1-PC Connect Attachment Line Station" on page 57
- "CP DEF NODE PC—Define Personal Computer Node" on page 65
- "CP DEF VOLUME—Define Volume Station" on page 73.

The following commands were changed:

- "CP F ADDR—Change Device, DTE Line, Ring, or Local Address" on page 75
- "CP F LINE—Change Controlling Line" on page 95
- "CP H—Halt Station" on page 137
- "CP P—Stop Station" on page 143
- "CP Q NET—Display $.SYSNET Station Definitions" on page 153
- "CP Q PARM—Display System Parameters of Active Station" on page 159.

CFBUF Support: The Communications Facility now monitors the use of space in the message buffer pool. All the CP Q examples were modified to reflect the change.

The following commands were added:

- "CP F CLASS—Change Usage Class" on page 83
- "CP SET CFBUF—Set Levels in CFBUF" on page 177
- "CP Q CFBUF—Display CF Message Buffer Pool Usage" on page 151.

Teletypewriter Adapter Support: Since the Communications Facility no longer supports a 3101 device attached to the Series/1 through a teletypewriter adapter, the "CP DEF DEVICE 3101" command has been removed.

Other Changes: The following commands were changed:

- "CP DEF ALIAS—Define Alias Station" on page 37
- "CP DEF CIRCUIT SVC—Define X.25 Switched Virtual Circuit Station" on page 41
- "CP F PKTSIZ—Change Line or Circuit Station's Packet Size" on page 109
- "CP F WINDOW—Change Line or Circuit Station's Window" on page 129
- "CP FILE NAME—Assign or Change Disk-Queue Data Set" on page 131
- "CP FILE PARMS—Display Disk-Queuing Parameters" on page 133
- "CP FILE RESET—Delete Disk-Queue Data Set" on page 135
- "PD Q PSEUDO | REMOTE—Display Remote Disk Definitions" on page 225.

Explanations have been added to the description of the commands.

Version 2.0

The following changes were made between the Communications Facility Version 1.2 and Version 2.0:

X.25 Support: A new IOCP, $.IO0AB8, was added to support communications using Recommendation X.25 protocols.

A description of the new data set, $.SYSX25, was added to the section "System Data Sets" on page 3. The following commands were added:

- "CP DEF CIRCUIT PVC—Define X.25 Permanent Virtual Circuit Station" on page 39
- "CP DEF CIRCUIT SVC—Define X.25 Switched Virtual Circuit Station" on page 41
- "CP DEF LINE DxE—Define HDLC Line Station" on page 53

- "CP F ADDR—Change Device, DTE Line, Ring, or Local Address" on page 75
- "CP F CALLID—Change SVC Circuit Station's Call ID" on page 81
- "CP F FAC—Add or Change SVC Circuit Station's Facilities" on page 87
- "CP F LCI—Change PVC Circuit Station's Logical Channel Identifier" on page 93
- "CP F PKTSIZ—Change Line or Circuit Station's Packet Size" on page 109
- "CP F PROTID—Change SVC Circuit Station's Protocol Identifier" on page 115
- "CP F WINDOW—Change Line or Circuit Station's Window" on page 129
- "CP Q X25—Display Status Information for Active XHCS Lines" on page 169.

The following commands were changed:

- "CP F MODE—Change Mode of Operation" on page 99
- "CP F USER—Change SNA Logon or X.25 Call Request User Data Field" on page 125.

Model 30D and 60D Integrated Disk Storage Support: The remote IPL transaction was changed to support an IPL from the 30 megabyte and 60 megabyte integrated disks. (Communications Facility books call these disks DDSK-30 and DDSK-60 for consistency with the Event Driven Executive books.) The format of the transaction and the responses from the transaction were updated in the section "IPL—Remote IPL Transaction" on page 253.

4980 Display Support: "CP DEF DEVICE 4978—Define 4978/4980 Device Station" on page 49 was changed to define a station for the 4980.

LU-1 Support: Two new modes, 3270 and SCS, were added for the LU-1 support. Their descriptions were added to "CP F MODE—Change Mode of Operation" on page 99.

CP HELP Enhancements: "CP HELP—Display CP Commands" on page 139 presents the new parameters for the HELP command and gives examples of the output.

$.CONFIG Enhancements: You may now use $.CONFIG from a 3101 device. Three new commands, COPY, ASSIST, and SET EXIT add to the usability of $.CONFIG. "$.CONFIG—Configuration Processor Utility Program" on page 285 was changed to present the new commands. Examples of the new screens were included with the description.

Publication Changes: The messages section of the Version 1.2 book is now in a separate book, *Messages and Codes.*

The glossary and index are now in separate sections instead of combined as in previous editions.

Summary of Amendments

The following changes were made between the Communications Facility Version 1.2 and Version 2.0: (Only changes affecting the content of this book are listed.)

X.25 Support: A new IOCP, $.IO0AB8, was added to support communications using Recommendation X.25 protocols.

A description of the new data set, $.SYSX25, was added to the section "System Data Sets" on page 3. The following commands were added:

- "CP DEF CIRCUIT PVC—Define X.25 Permanent Virtual Circuit Station" on page 35
- "CP DEF CIRCUIT SVC—Define X.25 Switched Virtual Circuit Station" on page 37
- "CP DEF LINE DxE—Define HDLC Line Station" on page 51
- "CP F ADDR—Change Device, DTE Line, Ring, or Local Address" on page 67
- "CP F CALLID—Change SVC Circuit Station's Call ID" on page 73
- "CP F FAC—Add or Change SVC Circuit Station's Facilities" on page 77
- "CP F LCI—Change PVC Circuit Station's Logical Channel Identifier" on page 83
- "CP F PKTSIZ—Change Line or Circuit Station's Packet Size" on page 99
- "CP F PROTID—Change SVC Circuit Station's Protocol Identifier" on page 105
- "CP F WINDOW—Change Line or Circuit Station's Window" on page 115
- "CP Q X25—Display Status Information for Active XHCS Lines" on page 151

The following commands were changed:

- "CP F MODE—Change Mode of Operation" on page 89
- "CP F USER—Change SNA Logon or X.25 Call Request User Data Field" on page 113

Model 30D and 60D Integrated Disk Storage Support: The remote IPL transaction was changed to support an IPL from the 30 megabyte and 60 megabyte integrated disks. (Communications Facility books call these disks DDSK-30 and DDSK-60 for consistency with the Event Driven Executive books.) The format of the transaction and the responses from the transaction were updated in the section "IPL—Remote IPL Transaction" on page 233.

4980 Display Support: "CP DEF DEVICE 4978—Define 4978/4980 Device Station" on page 47 was changed to define a station for the 4980.

LU-1 Support: Two new modes, 3270 and SCS, were added for the LU-1 support. Their descriptions were added to "CP F MODE—Change Mode of Operation" on page 89.

CP HELP Enhancements: "CP HELP—Display CP Commands" on page 125 presents the new parameters for the HELP command and gives examples of the output.

$.CONFIG Enhancements: You may now use $.CONFIG from a 3101 device. Three new commands, COPY, ASSIST, and SET EXIT add to the usability of $.CONFIG. "$.CONFIG—Configuration Processor Utility Program" on page 265 was changed to present the new commands. Examples of the new screens were included with the description.

Publication Changes: The messages section of the Version 1.2 book is now in a separate book, *Messages and Codes.*

The glossary and index are now in separate sections instead of combined as in previous editions.

Contents

Figures

Communications Facility Overview

The Communications Facility consists of programs that run under control of the Event Driven Executive system on a Series/1. It manages communication among programs and various input/output devices such as terminals, printers, and communication lines. It supports communication within a Series/1, between Series/1s, and between a Series/1 and other computers.

A Series/1 in the Communications Facility configuration is called a *node*. A *station* is a named unit of hardware or software managed by the Communications Facility. A *message* is a unit of data to be transmitted from one station to another. The Communications Facility maintains a queue of messages for each station. Messages are queued in processor storage. Optionally, messages for specific stations may be queued on disk.

You'll manage stations in your Communications Facility configuration through commands, through special messages called *transactions*, and through utility programs.

The Control Program

The control program is the part of the Communication Facility that provides basic, required services. It consists of three functions—the message dispatcher, the command processor, and the log processor. Its program name is $.CFD if your system uses disk queuing of messages as well as storage queuing, or $.CFS if your system uses just storage queuing. Both versions take the name $.CF after you load either control program.

The Message Dispatcher

The *message dispatcher* is the part of the Communications Facility that determines the destination of a message and places it on the destination station's message queue. The message dispatcher cannot deliver a message if, for example, the destination station is unknown or inactive. Your installation can define a station called $.WASTE to which undeliverable messages are sent. If there is no $.WASTE station, the message dispatcher discards undeliverable messages.

The Command Processor

The *command processor* is the part of the Communications Facility that checks a command for syntactic validity and processes it. The commands give you online control of your Communications Facility system and allow you to display different kinds of information about the system.

The Log Processor

The *log processor* is the part of the Communications Facility that receives error and informational messages from Communications Facility programs. It formats the messages and sends them to the system log.

The Program Dispatcher Program

The *program dispatcher* program, called $.PD, is an optional part of the Communications Facility. It manages the routing of transactions and controls the execution of programs that process transactions. A *transaction* is a special-format, user-defined message, the sending of which causes a program to be executed somewhere in the Communications Facility network. A Series/1 in which the program dispatcher is running is called a *cell.* Transactions sent to the program dispatcher are placed on its message queue by the message dispatcher. When the program dispatcher retrieves a message from its queue, it determines from the information in the transaction where it is to be processed. If it is to be processed in this cell, the program dispatcher routes it to the appropriate transaction processing program. If it is to be processed in a different cell, the program dispatcher routes it to that cell.

Program Dispatcher Tables

The program dispatcher program uses two tables in storage, a transaction table and a path table. The program dispatcher builds these tables from a user created data set called $.SYSPD.

The *transaction table* contains, for each transaction, its identifier and the name and attributes of its associated program. The program dispatcher uses the transaction table to determine which program is to process a transaction and to control the execution of that program.

The *path table* is a table of paths that the program dispatcher uses to route transactions between cells.

Remote Disk Support

Remote disk support is an optional part of the Communications Facility. It allows programs in one Series/1 to access disks attached to a different Series/1. This support is provided by the program dispatcher through a program called $.PD<IO>.

I/O Control Programs

I/O control programs (IOCPs) manage devices such as terminals, printers, and communications lines. The Communications Facility includes several I/O control programs—one for each type of device. They are called $.IO*xxxx*, where *xxxx* identifies the device type. Only those I/O control programs needed to manage the devices in your system are in storage when the Communications Facility is operational.

System Pools

This section describes the Communications Facility system pools.

Message Buffer Pool

The *message buffer pool*, called CFBUF, is an area of processor storage that contains the message queues. CFBUF is allocated from the dynamic storage defined in the PROGRAM statement of the control program. CFBUF also contains information about stations that have message queues on disk. A message remains in the buffer pool, or on a disk queue, until the destination station retrieves it.

System Storage Pool

The *system storage pool*, called S$POOL, is an area of processor storage that contains information about stations and work areas used by various Communications Facility functions. Each station has a control block, called a *station block*, that is created when the station is started and deleted when the station is halted. A station block contains the information required to manage the station, including a pointer to the station's message queue in CFBUF.

System Data Sets

This section explains what you need to know about Communications Facility system data sets.

$.SYSNET

$.SYSNET is a data set containing the definitions of stations in a Communications Facility configuration. Each node has one $.SYSNET data set. Each data set contains the definitions of all stations in its node and the definitions of stations in other nodes with which that node communicates. You use CP commands or utility program $.CONFIG to enter and modify station definitions in $.SYSNET.

$.SYSIPL

You can enter commands in a data set called $.SYSIPL. *$.SYSIPL* is used to start up the Communications Facility when the control program is loaded. You can learn how to create $.SYSIPL in the section "Preparing $.SYSIPL" on page 26.

$.SYSPD

$.SYSPD is the data set containing command processor commands, path definitions, transaction definitions, and transactions. This data set is read by the program dispatcher when it is started. The program dispatcher uses the information in $.SYSPD to build its tables and to start the stations required for transaction processing. You can learn how to create $.SYSPD in the section "Preparing $.SYSPD" on page 26.

$.SYSMSG

$.SYSMSG is a data set that contains the text of Communications Facility log messages. The *Design and Installation Guide* explains how to modify existing messages and define new ones.

$.SYSPRT

$.SYSPRT is a data set that contains control and image stores for 4978 terminals managed by the Communications Facility. It can also contain members used to support Series/1 printers as buffered devices. These members are allocated when you define printer device stations.

$.SYSLCC

$.SYSLCC is a user created data set that contains information used to IPL remote Series/1s on a Local Communications Controller ring. The *Design and Installation Guide* explains how to create this data set.

$.SYSX25

$.SYSX25 is a user-created data set that contains X.25 network addresses of DTEs and their user-defined call IDs. The X.25 I/O control program uses this data set when it needs to translate a call ID to an X.25 network address. The *Design and Installation Guide* explains how to create this data set.

Stations

Each I/O device or program that sends and receives messages is called a station. You identify a station defined to the Communications Facility by its station name and network address.

The *station name* is 1 to 8 alphameric characters. Each station defined in a node must have a unique name. You refer to a station by its name when you issue operator commands or use utility programs.

The *network address* is 4 hexadecimal digits. Each station defined in a network must have a unique network address. The leftmost two digits, called the *node assignment*, identify the node in which the station exists. The rightmost two digits, called the *station address*, identify the station within the node. The message dispatcher and other parts of the Communications Facility refer to a station by its network address.

The following sections describe the different types of stations and their particular characteristics.

Line Station

A *line station* represents a communications link from a Series/1 to a host computer, to another Series/1, to another device or computer, or to a packet-switching network as defined by CCITT Recommendation X.25. The link may be a Local Communications Controller ring, a channel attachment to a host computer, a binary synchronous (BSC) line, or an HDLC line.

A BSC line can connect to:

- a Series/1
- systems using point-to-point protocol
- a 3270 system
- a host computer.

An HDLC line can connect to:

- an X.25 network
- a Series/1 that has been defined as data terminal equipment
- a Series/1 that has been defined as data circuit-terminating equipment
- a device or computer that has been defined as data terminal equipment.

When communicating with a host computer, either over a channel or a BSC line, the Series/1 appears to host programs as if it were a 3270 system. This sort of communication is called *3270 emulation.* I/O control programs manage communication over lines. Each line station has a task control block that is associated with a reentrant task in the I/O control program.

SNA Physical Unit and Logical Unit Stations

SNA stations are used to communicate with a host computer over a synchronous data link control (SDLC) line. In this sort of communication, the Series/1 appears to host programs as if it were a 3270 system. This sort of communication is called *3270 emulation.* An *SNA logical unit station* represents an emulated 3270 terminal. An *SNA physical unit station* is a station block that contains information common to its associated SNA logical unit stations. When you define an SNA logical unit station, you specify the name of its associated SNA physical unit station. An I/O control program manages communication with SNA stations. Each SNA logical unit station has a task control block that is associated with a reentrant task in the I/O control program.

Node Station

A *node station* represents a remote node. When you define a node station for a node that is connected to the same Local Communications Controller ring as the local node, you specify the name of the Local Communications Controller line station. The line station task, which is part of an I/O control program, manages communication with the node station.

When you define a node station for a node that is not connected to the same Local Communications Controller ring as the local node, you link it to the station that provides access to the remote node. That station may be a Series/1-to-Series/1 BSC line station, a circuit station, or a node station associated with a Local Communications Controller line station.

Terminal Station

A *terminal station* represents either a real 3270 terminal attached to a Series/1 through a BSC line or an emulated 3270 terminal. Emulated 3270 terminals are just station blocks with message queues that are used for communication with a host computer over a BSC line or channel. When you define a terminal station, you specify the name of its associated line station. The line station task, which is part of an I/O control program, manages communication with the terminal station.

User Station

A *user station* represents a user application program, a Communications Facility program, such as the program dispatcher, or an I/O control program.

Device Station

A *device station* represents a device attached to a Series/1. The device may be a 4978 terminal, a 4980 terminal, a 3101 terminal, a 7485 terminal, or a Series/1 printer. I/O control programs manage communication with device stations. Each device station has a task control block that is associated with a reentrant task in the I/O control program.

Circuit Station

A *circuit station* represents an X.25 virtual circuit—either permanent or switched. The circuit allows communications with other X.25 defined devices and computers. The physical connection is through an HDLC line, which may be connected directly to the other device or to an X.25 network. When you define the circuit station, you specify the name of its associated line station. The line station task, which is part of the I/O control program, manages communication with the circuit station. You also specify whether the circuit is used strictly for data transfer, whether it can be controlled by an application program, or whether it is the message path between two Communications Facility nodes.

Message Station

A *message station* is a station block with a message queue to be used for various purposes. The station that receives undeliverable messages, $.WASTE, may be a message station. The program dispatcher creates message stations to route transactions to the programs that process them.

Alias Station

An *alias station* is a station block that contains an alias name for another station. You can, for example, define XYZ as an alias for station ABC so that each time you refer to XYZ, you are actually referring to station ABC.

Vector Station

A *vector station* is a station block that represents a station in another node. You cannot define a vector station; the command processor creates it when you start a station located in another node (as indicated by the station's network address). Messages sent to these stations are routed to the line station that represents the connection between the two nodes.

System Station

There is only one system station. It is a station block named $.DISP, which is used by the control program. It is created when you IPL an EDX supervisor that supports the Communications Facility

Operator Commands

The majority of Communications Facility commands are command processor (CP) commands and program dispatcher (PD) commands.

CP commands let you control your Communications Facility system and display information about the system. Figure 1 summarizes these commands and their functions.

PD commands, a subset of CP commands, let you control transactions and transaction routing, send commands to cells, send messages to terminals, and set the time and date in a cell. Figure 2 on page 7 summarizes these commands and their functions.

A few commands are not prefixed by CP or PD. Figure 3 on page 7 summarizes these commands and their functions.

Command	Activity
DEF	Define a new station
F	Modify attributes of an existing station
FILE	Display or modify the disk-queuing parameters of a station
H	Halt a station

Figure 1 (Part 1 of 2). Summary of Command Processor (CP) Commands

Command	Activity
HELP	List CP commands and their functions
LINK	Define a connection between two stations so that messages may be routed between them
P	Stop a station
Q	Display information about stations, BSC and HDLC lines, and EDX terminals
READ	Execute commands contained in a data set
S	Start a station
SET LOG	Assign a log device for system log messages
SET NODE	Set the node address of a Series/1
ST	Display message activity statistics and Local Communications Controller hardware statistics
V	Remove node from the Local Communications Controller ring

Figure 1 (Part 2 of 2). Summary of Command Processor (CP) Commands

Command	Activity
C	Change the cell identifier
CP	Send a command to a cell
F	Modify a path or transaction table entry
H	List PD commands and their functions
I	Insert a path or transaction table entry
ID	Check whether a cell is active
M	Send a message to an EDX terminal
P	Stop a path or transaction
Q	Display the transaction and path tables
R	Remove a path or transaction table entry
RC	Set retry counts for program load
S	Start a path or transaction
T	Send the system time and date to a cell
TRAC	Start or stop a trace of transactions
TRAN	Send a transaction
UP	Set user program ($.UP*xxxx*) local mode

Figure 2. Summary of Program Dispatcher (PD) Commands

Command	Activity
CPRSTART	Restart the Communications Facility
GOTEST	Begin execution of a program in test mode
TRAN	Send a transaction
WSC	Start the work session controller terminal

Figure 3. Summary of Other Commands

IBM-Supplied Transactions

IBM-supplied transactions let you IPL remote cells, check on the stations and programs running in remote cells, schedule transaction processing, and start remote disk activity. These transactions are processed by the program dispatcher, rather than by user-written transaction-processing programs. Figure 4 summarizes these transactions and their functions.

Transaction	Activity
IPL	IPL a remote cell
SCHD	Schedule a transaction to be processed later
SYST CP	Send a command to a remote cell
SYST HL	Load a program
SYST ID	Check whether a cell is active
SYST MS	Send a message to an EDX terminal
SYST RC	Set retry counts for program load
SYST RH	Release held transaction or path
SYST SP	Start remote disk
SYST ST	Check whether a station was created in another cell
SYST TI	Set the system time and date in a cell
SYST TR	Start or stop a trace of transactions
SYST UP	Set user program ($.UP*xxxx*) load mode
SYST WH	Check whether a program was loaded in another cell

Figure 4. Summary of IBM-Supplied Transactions

Utility Programs

Utility programs let you initialize disk-queue data sets, maintain remote Series/1s, control access to remote disks, alter system pool space, and manage message queues. Figure 5 on page 9 summarizes the utility programs described in this book and their functions.

Utility Program	Activity
$.CONFIG	Define, alter, and display station definitions and system log messages
$.DSINIT	Initialize data sets for the disk queuing of messages
$.HMU/$.RMU	Maintain remote Series/1s from a host Series/1
$.PDBSTS	Control a remote disk volume directory to prevent concurrent updates
$.SETSTG	Alter the size of the space used for disk-queue file control blocks and the message buffer pool
$.UT1	Display system diagnostic information, send and receive messages, and delete stations
$.UT2	Purge and report undeliverable messages on the $.WASTE queue or messages on any other message queue

Figure 5. Summary of Utility Programs

Entering Commands

There are several different ways of issuing commands. They can be entered from a terminal; read from a data set; or submitted from a program, a transaction, or a utility program.

Depending on the method used, the commands may be local (processed by the command processor in this node) or remote (sent to a command processor in some other node).

Commands are translated to upper case before they are processed, regardless of how they are issued.

Entering Commands from a Terminal

You may issue a local command from an EDX terminal by pressing the ATTN key and entering CP or PD followed by the function and parameters you want. For example:

> CP H $.CF $.IO0AE0 $.IO0630

Note that commands don't have to be issued in the partition where the command processor is running. You can issue them from any partition.

You may issue a remote command from an EDX terminal by using the program dispatcher command CP. Enter the identifier of the cell where the command is to be processed and the function and parameters you want. For example, to send a command to start station E32776 in cell S3, enter:

> PD CP S3 S E32776

You may also issue commands from a 3277 Communications Facility terminal, as explained in the description of the I/O control program $.IO0AC0 in the *Design and Installation Guide.*

Entering Commands in a Data Set

You may put local commands in a data set, using any EDX editor. You just enter the function and parameters that you want and terminate the data set with a /*. For example:

```
SET NODE 0100
S BSC58I BSC0B
LINK BSC0B BSC58I
S $.PD
/*
```

The commands are read and processed when a CP READ command that names the data set is issued. When the control program is started, it issues a read for data set $.SYSIPL. You may issue a read for a command data set with any name.

Entering Commands with the SYST CP Transaction

You may use the SYST CP transaction to issue local or remote commands by entering the function and parameters you want as the transaction data. For example, the transaction to start station E32776 in cell S3, where *b* represents a blank is:

SYSTS3*bbbbbb*CPS*b*E32776

Entering Commands with the SEND CP Instruction

Local or remote commands may be issued from a program by means of the SEND CP instruction, as explained in the *Programmer's Guide.* The desired function and parameters are specified as an operand of the instruction. For example, to halt station MSG1:

SEND CP,,'H MSG1'

You cannot use SEND CP to issue a remote command when the connection to the remote node is a BSC multipoint line (one managed by $.IO0AC0 or $.IO0AE0).

Entering Commands with the $.UT1 Utility

Local or remote commands may be issued from the diagnostic aid utility program, $.UT1, as explained in the chapter "Utility Programs." The desired function, parameters, and destination are specified in response to prompts from the program. For example:

```
COMMAND (?): CP
CP: H MSG1
DESTINATION IS '$.DISP', OK? Y
```

You cannot use $.UT1 to issue a remote command when the connection to the remote node is a BSC multipoint line (one managed by $.IO0AC0 or $.IO0AE0).

Communication from the Communications Manager

Through the Local Communications Controller, the Realtime Programming System's Communications Manager can send commands to the Communications Facility. If it sends an invalid command, an error message is logged at the Communications Facility system log.

Operating the Communications Facility

The information in this chapter will:

- Familiarize you with the parts of the Communications Facility you'll be using in operating the system.

- Acquaint you with the major functions you can perform, including starting and stopping the system, modifying the system, defining stations, displaying station information, and maintaining the remote Series/1s in your network.

- Identify the operator commands, IBM-supplied transactions, and utility programs with which you perform these functions. (The format and syntax of the operator commands are covered in "Operator Command Reference" on page 31, transactions in "IBM-Supplied Transactions Reference" on page 231, and the utilities in "Utility Programs" on page 263.)

Starting the Communications Facility

To start the Communications Facility, you'll need to start the control program and, if it is part of your system, the program dispatcher. The control program must be started first.

Assigning a Partition

You'll need to know if the partition currently assigned to your EDX terminal is *mapped* (contains the common area) and is large enough to contain the control program. To find the partition to which your terminal is currently assigned, issue this EDX attention command:

> $A

A command response like the following will be displayed:

```
PROGRAMS AT 08:15:31
IN PARTITION #3 - NONE
            PART. ADDR:  1800 HEX;  SIZE:  59392 DEC. BYTES
```

If you need to change partitions, enter the following EDX attention command:

> $CP *n*

where *n* is the number of the partition you want.

Starting the Control Program

Once you've assigned the appropriate partition, you can load the control program. If your system uses just storage queuing of messages, issue this EDX command:

> $L $.CFS

If your system uses disk queuing as well as storage queuing of messages, issue this EDX command:

> $L $.CFD

If you've loaded the control program into a partition that is not large enough or is not mapped, you'll get the following messages:

```
$.CFS         55P,08:20:50, LP= 4A00, PART= 5
*08:20:50 CF57 E  $.CF CANNOT BE STARTED
$.CF     ENDED AT 08:20:50
```

If the control program was already operational, you'll get the following message and the newly loaded copy terminates itself:

```
$.CFS         55P,08:21:57, LP= 1800, PART= 3
*08:21:57 CF56 E 0000 $.CF      $.CF ALREADY OPERATIONAL
$.CF     ENDED AT 08:21:57
```

If you've successfully loaded the control program, the system will respond with some messages like these:

```
$.CFS         55P,08:24:43, LP= 1800, PART= 3
+08:24:45 CF66 I SYSTEM DATE IS NOW - 11/09/81
 08:24:46 CF58 I 0000 $.CF      EDX/CF IS OPERATIONAL
*08:24:47 CP29 E 0000 $.CPREAD $.SYSIPL DATA SET NOT FOUND
```

If the system startup commands are in $.SYSIPL, the system initialization data set, you will not see the last message displayed. More information about $.SYSIPL is included later in this chapter.

Verifying Control Program Operation

You may want to verify that the control program is operational. You can issue a *CP Query* command, which displays information about active stations:

> CP Q

or

> CP Q *

You'll get a display like this:

08:25:46 QUEUE DISPLAY TYPE: (IN STORAGE)

STATION	TYPE	NA	LINK	STAT	LINE	FIQ	LIQ	ISN	OSN	DISK	MSG	ACT
$.DISP	0000	0000	0000	8000	201C	0000	0000	0000	0000	N/A		

The name of the station is listed under STATION; $.DISP is the station that represents the message dispatcher. The STAT column shows the station's status information in hexadecimal. The station is active if the first status bit is 1, inactive if it's 0. The section "CP Q *—Display Active Station Information" on page 137 explains the information in the other columns.

Changing the System Log Device

When the control program is successfully loaded, the EDX terminal from which it was loaded becomes the Communications Facility system log device. All system messages are logged on this device. If you want a hard copy of the system messages, use the *CP SET LOG* command to change the log device:

> CP SET LOG $SYSPRTR

This causes all system messages to be logged to the printer instead of to the terminal. Both the current and newly assigned log devices will display this message:

```
08:26:21 CP07 I 23DA $.CPSET  $SYSPRTR SYSTEM LOG SET
```

Starting Stations

The Communications Facility provides a *CP Start* command that you'll use to start each station in your configuration.

If your system uses $.WASTE, the station for undeliverable messages, you should start it before you start any other stations. Use the following CP command to start $.WASTE:

> CP S $.WASTE

You'll get messages like these:

```
08:30:14 CP10 I 0000 $.CPS    $.WASTE         MESSAGE STATION STARTED
08:30:15 CP67 I 0000 $.CPS    $.WASTE         DISK QUEUING IS ACTIVE
08:30:16 CP65 I 0000 $.CPS    WASTFILE,EDX002 IN USE FOR DISK QUEUING
```

Assume your configuration matches the one in Figure 6 on page 17 and the stations have already been defined as described in the *Design and Installation Guide*. To start the stations, issue this command:

> CP S HOST CU71 T770 T4978

You'll get these messages although not necessarily in this order:

```
1  08:41:21 CP14 I 0000 $.CPS     HOST    LINE START REQUESTED
2  08:41:24 CP15 I 0000 $.CPS     CU71    TERMINAL STARTED
3  08:41:27 CP15 I 0000 $.CPS     T770    TERMINAL STARTED
4  08:41:29 IO52 I 0000 $.IOOAEO HOST    LINE OPENED
5  08:41:32 CP18 I 0000 $.CPS     T4978   DEVICE START REQUESTED
6  08:41:35 IO62 I 2186 $.IO0670 T4978   DEVICE IS OPEN
7 *08:41:36 IO10 E 0414 $.IOOAEO HOST    TIME-OUT
```

where:

Messages 1, 2, 3, and 5 are displayed by the command processor start program, $.CPS, when the station is started.

Message 4 confirms that the 3270 emulation I/O control program is controlling the line.

Message 6 confirms that the terminal I/O control program for the device type 4978 is controlling the device station T4978.

Message 7 is an error message indicating that the emulated control unit is not being polled by the host system. This is true with many host systems because the host removes the control unit device address from the polling list if there is no response from that control unit. In this case, you should call the host operator and ask that the 3271 cluster or terminal be activated.

You can detect when polling has started. If the Series/1 is equipped with a programmer console, bit 14 of the display blinks. If the Series/1 is equipped with a communications indicator panel connected to the BSC line to the host system, bit 0 and bit 1 stay on and bits 2 through 7 flash every time the line is polled.

Verifying Station Operation

You may want to verify that the stations you've started are active. Issue this CP Query command:

CP Q

You'll get a display like this:

```
08:43:08  QUEUE DISPLAY TYPE:    (IN STORAGE)
STATION   TYPE  NA   LINK STAT LINE FIQ  LIQ  ISN  OSN  DISK MSG ACT
$.DISP    0000 0000 0000 8000 201C 0000 0000 0000 0000   N/A
$.WASTE   0C00 01AC 0000 8000 0000 0000 0000 0000 0000   NO      YES
$.IOOAE0 0200 01AA 0000 8000 0000 0000 0000 0000 0001   N/A
$.IO0670 0200 01AB 0000 8000 0000 0000 0000 0000 0001   N/A
CU71      04E0 01C0 0000 8000 0109 0000 0000 0000 0000   N/A
T770      04E2 01C1 0124 8000 0109 0000 0000 0000 0000   N/A
T4978     0678 0124 01C1 8000 0916 0000 0000 0000 0000   N/A
HOST      0AE0 0109 0000 8002 0000 0000 0000 0000 0000   N/A
```

Notice that there are two active I/O control programs. $.IO0670, which manages 4978 devices, was started when you started station T4978. $.IO0AE0, which manages BSC lines used for 3270 emulation, was started when you started station HOST. The STAT column shows the station's status information in hexadecimal. The station is active if the first status bit is 1, inactive if it's 0. Refer to the section "CP Q *—Display Active Station Information" on page 137 for a description of each column displayed.

Linking Stations

You can use the *CP LINK* command to define a connection between two stations so that one station has the other as the default destination of messages it sends. Notice the LINK field of stations T770 and T4978 in the above display. They should be linked to each other; that is, they should contain each other's network address.

If the stations were not linked at installation, use a CP LINK command to link them at this time:

> CP LINK T770 T4978 BOTH

You'll get these messages:

```
08:44:05 CP32 I 0124 $.CPLINK T770      LINKED
08:44:07 CP32 I 01C1 $.CPLINK T4978     LINKED
```

The 4978 device is now a Communications Facility terminal and is able to communicate with the host system as a 3270 terminal. You'll get a display of the host system screen by pressing the ENTER key at the 4978 device.

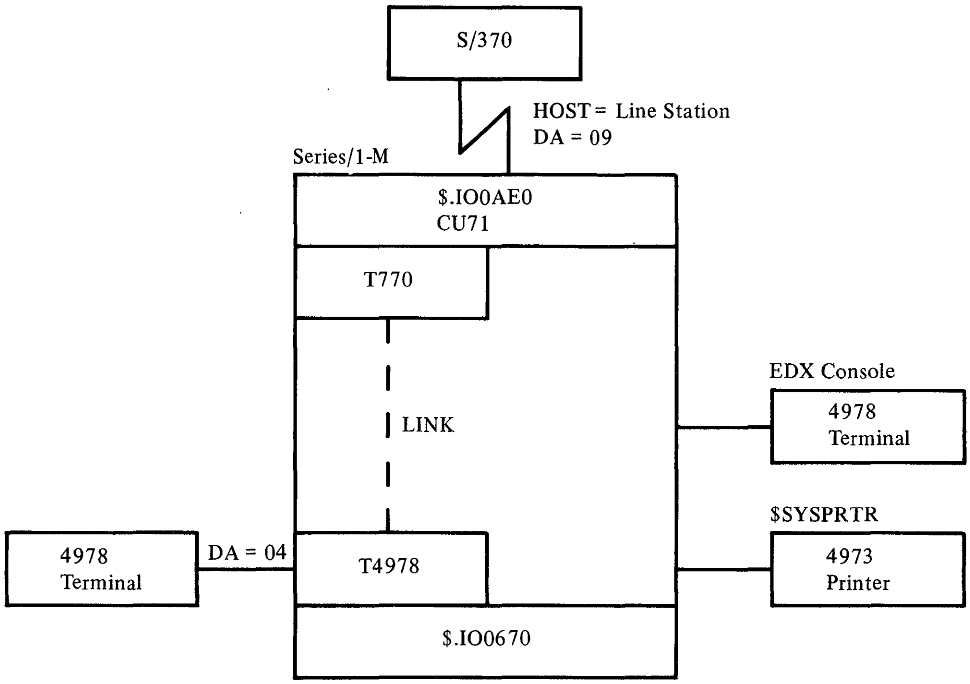

Figure 6. 3270 Pass-through

Starting the Program Dispatcher Program

Assume your network configuration matches the one shown in Figure 7 on page 18 and there is a $.SYSPD data set for each Series/1 as shown in Figure 8 on page 21, Figure 9 on page 21, and Figure 10 on page 22 . Also assume that you have already started the control program and the stations for 3270 pass-through in each Series/1, as described in the first part of this chapter. To activate the rest of the network, you need only start the program dispatcher in each Series/1. To start the program dispatcher, issue this command:

> CP S $.PD

When the program dispatcher is started, it reads its $.SYSPD data set. It issues the commands in the first section of the data set, builds its path and transaction tables from the statements in the second and third sections, and then processes the transactions in the last section.

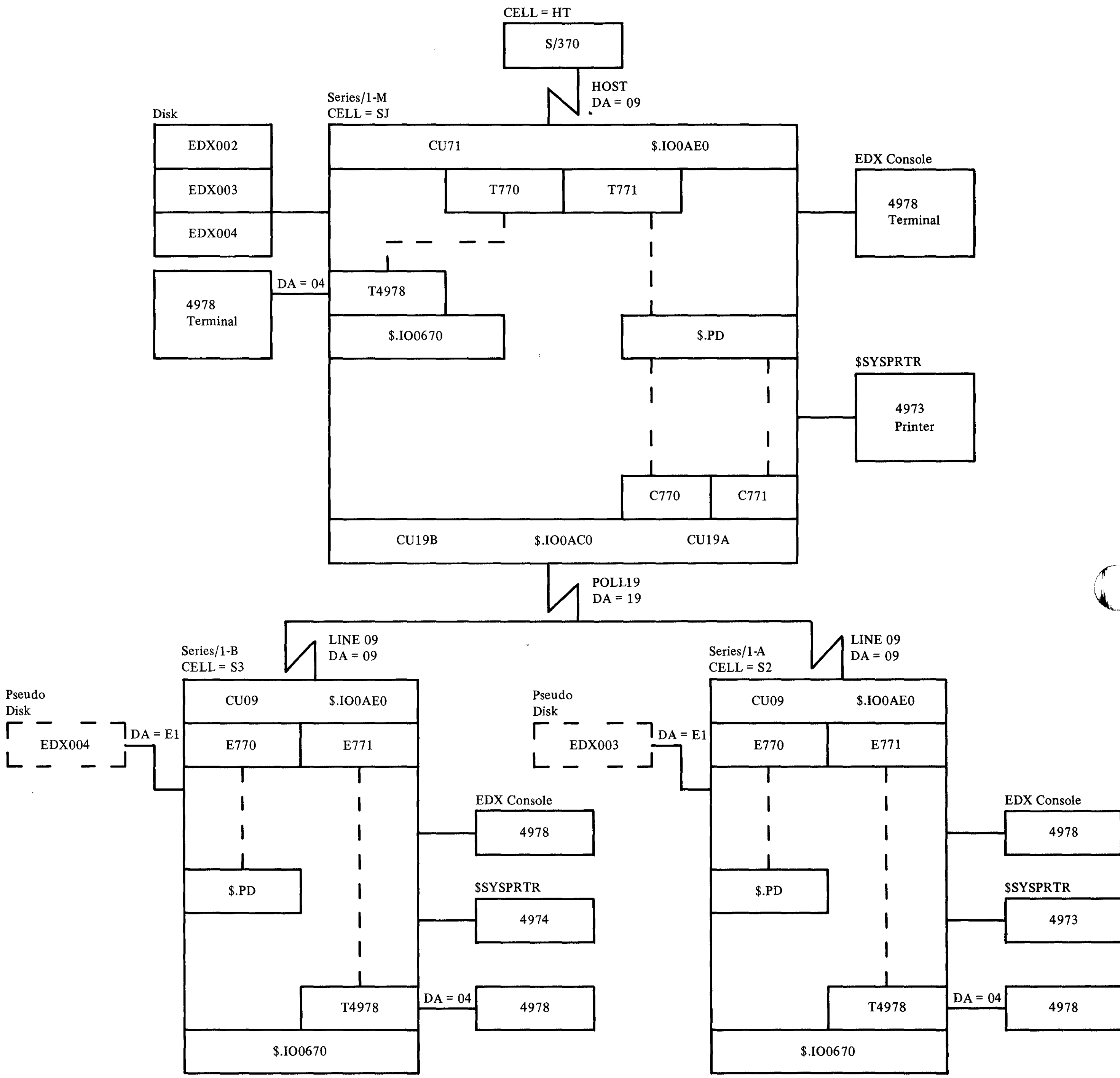

Figure 7. Transaction Processing

Starting the program dispatcher in Series/1-M will result in log messages:

```
1  08:45:00 CP17 I 0000 $.CPS     $.PD     USER STARTED
2  08:45:05 CP07 I 23DA $.CPSET   $SYSPRTR SYSTEM LOG SET
3  08:45:08 CP14 I 0000 $.CPS     POLL19   LINE START REQUESTED
4  08:45:12 IO52 I 0000 $.IOOACO  POLL19   LINE OPENED
5  08:45:18 CP15 I 0000 $.CPS     CU19A    TERMINAL STARTED
6  08:45:22 CP15 I 0000 $.CPS     CU19B    TERMINAL STARTED
7  08:45:25 CP40 I 0000 $.CPF     T771     DISK DEFINITION MODIFIED
8  08:45:28 CP15 I 0000 $.CPS     T771     TERMINAL STARTED
9  08:45:33 CP32 I 01AE $.CPLINK  T771     LINKED
10 08:45:39 CP40 I 0000 $.CPF     C770     DISK DEFINITION MODIFIED
11 08:45:42 CP15 I 0000 $.CPS     C770     TERMINAL STARTED
12 08:45:44 CP32 I 01AE $.CPLINK  C770     LINKED
13 08:45:49 CP40 I 0000 $.CPF     C771     DISK DEFINITION MODIFIED
14 08:45:52 CP15 I 0000 $.CPS     C771     TERMINAL STARTED
15 08:45:54 CP32 I 01AE $.CPLINK  C771     LINKED
16 08:45:59 PD01 I 0500 $.PD      OPERATIONAL
```

Message 1 indicates the program dispatcher is being started.

Message 2 indicates $SYSPRTR is the system log device.

Messages 3, 5, 6, 8, 11, and 14 are displayed when the line station, the emulated control unit and terminal stations are started. Stations POLL19, CU19A, and CU19B are started as a result of the CP S command in the $.SYSPD data set. Stations T771, C770, and C771 are started by the program dispatcher because they are defined as paths to other cells.

Message 4 is displayed by the I/O control program when communication with the BSC line has been initiated.

Messages 7, 10, and 13 indicate that the stations that are paths to other cells have been modified to the text mode required by the program dispatcher. The CP F command is issued by the program dispatcher only if the stations are not already in text mode.

Messages 9, 12, and 15 indicate that the stations are paths to other cells have been linked to the program dispatcher.

Message 16 confirms that the program dispatcher is ready to process transactions. The hexadecimal number 0500 is the maximum transaction size. You can increase it by using the EDX utility $DISKUT2 to increase the size of the program dispatcher's dynamic storage.

Starting the program dispatcher in Series/1-A or Series/1-B will result in these log messages:

```
1  09:03:00 CP17 I 0000 $.CPS     $.PD      USER STARTED
2  09:03:05 CP07 I 23DA $.CPSET   $SYSPRTR SYSTEM LOG SET
3  09:03:08 CP14 I 0000 $.CPS     LINE09    LINE START REQUESTED
4  09:03:12 CP15 I 0000 $.CPS     CU09      TERMINAL STARTED
5  09:03:18 CP40 I 0000 $.CPF     E770      DISK DEFINITION MODIFIED
6  09:03:22 CP15 I 0000 $.CPS     E770      TERMINAL STARTED
7  09:03:25 CP32 I 01AE $.CPLINK  E770      LINKED
8  09:03:33 PD01 I 0500 $.PD      OPERATIONAL
9  09:03:44 IO52 I 0000 $.IO0AE0  LINE09    LINE OPENED
```

Message 1 indicates the program dispatcher is being started.

Message 2 indicates $SYSPRTR is the system log device.

Messages 3, 4, and 6 are displayed when the line station, the emulated control unit and terminal stations are started. Stations LINE09 and CU09 are started as a result of the CP S command in the $.SYSPD data set. Station E770 is started by the program dispatcher because it is defined as the path to another cell.

Message 5 indicates that the station that is the path to another cell has been modified to the text mode required by the program dispatcher.

Message 7 indicates that the station that is the path to another cell has been linked to the program dispatcher.

Message 8 confirms that the program dispatcher is ready to process transactions.

Message 9 is displayed by the I/O control program when communication with the BSC line has been initiated. This message is last because remote disk support uses the same BSC line. When the start remote disk transaction (SYST type SP) is sent, remote disk support relinquishes control of the BSC line to the I/O control program.

```
SET LOG $SYSPRTR
S HOST CU71 [1]
S POLL19 CU19A CU19B
/*
CELL SJ
PATH T771,HT,C P
PATH C770,S2
PATH C771,S3
RESERVE 4
/*
TID HMU  $.HMU 22
TID WSC  $.WSC 42
TID MENU $.WSMENU,,,N 22
TID IPL  $.PDIPL 22
TID <IO> $.PD<IO>,,,N 32
RESERVE 10
/*
SYST**bbbbbbMS********CELL SJ IS OPERATIONAL**
/*
```

Figure 8. $.SYSPD Data Set for Series/1-M

```
SET LOG $SYSPRTR
S LINE09 CU09
/*
CELL S2
PATH E770,SJ P
RESERVE 2
/*
TID RMU  $.RMU 22
TID WSC  $.WSC 42
TID MENU $.WSMENU,,,N 22
RESERVE 5
/*
SYST00bbbbbbSPE1SJ
SYST**bbbbbbMS********CELL S2 IS OPERATIONAL**
SYSTSJbbbbbbCP PD T S2
/*
```

Figure 9. $.SYSPD Data Set for Series/1-A

1 Probably already started for 3270 pass-through. No harm is done.

```
SET LOG $SYSPRTR
S LINE09 CU09
/*
CELL S3
PATH E770,SJ P
RESERVE 2
/*
TID RMU  $.RMU 22
TID WSC  $.WSC 42
TID MENU $.WSMENU,,,N 22
RESERVE 5
/*
SYST00bbbbbbSPE1SJ
SYST**bbbbbbMS********CELL S3 IS OPERATIONAL**
SYSTSJbbbbbbCP PD T S3
/*
```

Figure 10. $.SYSPD Data Set for Series/1-B

Verifying Program Dispatcher Operation

There are several ways to determine whether or not the program dispatcher is operational.

Broadcast Messages

The following broadcast messages should appear on all terminals of the three Series/1s when the program dispatchers are operational:

```
********CELL SJ IS OPERATIONAL**
********CELL S2 IS OPERATIONAL**
********CELL S3 IS OPERATIONAL**
```

in which the first eight asterisks are replaced by the name of the terminal where the message appears. The broadcast messages result from the SYST MS transactions in the $.SYSPD data sets.

PD Query Command

Another way of verifying that the program dispatcher program is functioning is to issue this *PD Query* command:

```
> PD Q ALL
```

which displays the program dispatcher's path and transaction tables. The resulting display on Series/1-M is:

```
09:05:42 PROGRAM DISPATCHER DISPLAY FOR CELL:  SJ TYPE:  ALL
CELL STATION MSGS MSGR #ERS STAT PFIX
 HT  T771     0000 0000 0000 9202
 S2  C770A    0000 0000 0000 8201
 S3  C770B    0000 0000 0000 8201
  4  OPEN TABLE ENTRIES

ID   PGM NAME VOLUME PT TY PGM ORGN LN VDEA #RLD RS PS OS AS STAT #MSG #ERS
HMU  $.HMU    EDX002  0 22 00002251 0E 26F4 00E8 2A 0C 1A 04 8000 0000 0000
WSC  $.WSC    EDX002  0 42 000043B1 29 26F4 02F8 2C 23 00 09 8000 0000 0000
MENU $.WSMENU EDX002  0 22 0000223F 0F 26F4 0160 0C 0C 00 00 8000 0000 0000
IPL  $.PDIPL  EDX002  0 12 000030FD 14 26F4 02C4 0E 0E 00 00 8000 0000 0000
<IO> $.PD<IO> EDX002  0 22 0000124E 26 26F4 034E 12 12 00 00 8000 0000 0000
  10 OPEN TABLE ENTRIES
```

Refer to "PD Q—Display Program Dispatcher Tables" on page 201 for a description of each field in this display.

PD Message Command

You can confirm that the program dispatcher is operational in each Series/1 by issuing a *PD Message* command that sends a message to all cells in the network. Receipt of the messages verifies that the paths between cells are operational and that the program dispatchers can route transactions between cells.

For example, from cell SJ:

> PD M ** * HELLO FROM CELL SJ

```
11:50:55 PD40 I 0000 $.CPPD   SENT TO ******** IN CELL **
```

From cell S2:

> PD M ** * HELLO FROM CELL S2

```
11:50:59 PD40 I 0000 $.CPPD   SENT TO ******** IN CELL **
```

From cell S3:

> PD M ** * HELLO FROM CELL S3

```
11:51:21 PD40 I 0000 $.CPPD   SENT TO ******** IN CELL **
```

All three hello messages should appear on all EDX terminals of all three systems.

Stopping and Halting Stations

You can use either the *CP Stop* (CP P) or the *CP Halt* command to inactivate a station. When you stop a station, it can no longer receive messages. Its disk-queue data set, if it has one, is closed and its station block is flagged inactive.

When you halt a station, the same actions occur. In addition, the station block is deleted from storage. You cannot halt stations that are controlled by other stations. To halt a terminal, circuit, or node station, you must halt its line station. To halt an SNA logical unit station, you must halt its SNA physical unit station.

After you have stopped or halted a station, you can issue a start command to restart it.

For a summary of the action taken for each station type, see "CP H—Halt Station" on page 123 and "CP P—Stop Station" on page 129.

Halting the Communications Facility

To shut down the Communications Facility, all started stations must be halted. This section shows you how to shut down system Series/1-M.

In order to shut down a system, you must know which stations are started. Issue the CP Query command:

> CP Q

```
12:32:08  QUEUE DISPLAY TYPE:    (IN STORAGE)
STATION   TYPE  NA   LINK STAT LINE FIQ  LIQ  ISN  OSN   DISK MSG ACT
$.DISP    0000 0000 0000 8000 201C 0000 0000 0000 0000   N/A
$.WASTE   0C00 01AC 0000 8000 0000 0000 0000 0000 0000   NO       YES
$.IO0AE0  0200 01AA 0000 8000 0000 0000 0000 0000 0001   N/A
$.IO0670  0200 01AB 0000 8000 0000 0000 0000 0000 0001   N/A
CU71      04E0 01C0 0000 8000 0109 0000 0000 0000 0000   N/A
T770      04E2 01C1 0124 8000 0109 0000 0000 0000 0000   N/A
T4978     0678 0124 01C1 8000 0916 0000 0000 0000 0000   N/A
HOST      0AE0 0109 0000 8002 0000 0000 0000 0000 0000   N/A
$.PD      0200 01AE 0000 8000 0000 0000 0000 0000 0000   N/A
$.IO0AC0  0200 01AD 0000 8000 0000 0000 0000 0000 0000   N/A
POLL19    0AC0 0119 0000 8002 0000 0000 0000 0000 0000   N/A
CU19A     04C0 01C4 0000 8000 0119 0000 0000 0000 0000   N/A
CU19B     04C0 01C7 0000 8000 0119 0000 0000 0000 0000   N/A
C770      04C2 01C5 01AE 8000 0119 0000 0000 0000 0000   N/A
C771      04C2 01C8 01AE 8000 0119 0000 0000 0000 0000   N/A
T771      04E2 01E2 01AE 8000 0109 0000 0000 0000 0000   N/A
$.PD<IO>  0C00 09DE 0000 8000 0000 0000 0000 0000 0000   N/A
```

You could halt each line, device, and user station. However, since you are shutting down the Communications Facility, it is simpler if you halt the controlling programs (the I/O control programs and the program dispatcher). They will halt the line, device, and and user stations they control before they halt themselves. The following examples show the correct sequence for halting the Communications Facility.

Shut down communications with Series/1-A and Series/1-B by halting the I/O control program that manages the line:

> CP H $.IO0AC0

```
12:32:15 CP71 I 0000 $.CPH     $.IO0AC0 HALT REQUEST ISSUED
12:32:18 CP23 I 0000 $.CPP     POLL19   STOP REQUEST ISSUED
12:32:20 IO57 I 0000 $.IO0AC0 POLL19   LINE STOPPED
12:32:24 CP23 I 0000 $.CPP     CU19A    STOP REQUEST ISSUED
12:32:27 CP23 I 0000 $.CPP     CU19B    STOP REQUEST ISSUED
12:32:29 CP23 I 0000 $.CPP     C770     STOP REQUEST ISSUED
12:32:30 CP23 I 0000 $.CPP     C771     STOP REQUEST ISSUED
12:32:32 IO80 I 0000 $.IO0AC0 HALT COMPLETED
```

Notice that the I/O control program shut down the line; if it were managing more than one line, it would have shut down all of them. Shutting down a line also causes a shut down of the terminals controlled by the line.

Disconnect the 4978 device from the Communications Facility by halting its I/O control program:

> CP H $.IO0670

```
12:32:51 CP71 I 0000 $.CPH    $.IO0670 HALT REQUEST ISSUED
12:32:53 CP23 I 0000 $.CPP    T4978    STOP REQUEST ISSUED
12:32:55 IO63 I 0000 $.IO0670 T4978    DEVICE IS CLOSED
12:32:59 IO80 I 0000 $.IO0670 HALT COMPLETED
```

Notice that the I/O control program shut down the device; if it were managing more than one device, it would have shut down all of them.

Shut down communication with the host by halting the I/O control program that manages the line:

> CP H $.IO0AE0

```
12:33:21 CP71 I 0000 $.CPH    $.IO0AE0 HALT REQUEST ISSUED
12:33:23 CP23 I 0000 $.CPP    HOST     STOP REQUEST ISSUED
12:33:27 IO57 I 0000 $.IO0AE0 HOST     LINE STOPPED
12:33:32 CP23 I 0000 $.CPP    CU71     STOP REQUEST ISSUED
12:33:34 CP23 I 0000 $.CPP    T770     STOP REQUEST ISSUED
12:33:35 CP23 I 0000 $.CPP    T771     STOP REQUEST ISSUED
12:33:37 IO80 I 0000 $.IO0AE0 HALT COMPLETED
```

Shut down the program dispatcher:

> CP H $.PD

```
12:45:10 CP71 I 0000 $.CPH    $.PD     HALT REQUEST ISSUED
12:45:14 PD19 I 0000 $.PD     NORMAL TERMINATION
```

The program dispatcher shuts down each transaction-processing program it manages but no messages are logged to show this. In this example, the remote disk support program, $.PD<IO>, is shut down.

A display shows that there are still two active stations, the message dispatcher and the undeliverable message station:

> CP Q

```
12:45:18  QUEUE DISPLAY TYPE:   (IN STORAGE)
STATION  TYPE NA   LINK STAT LINE FIQ  LIQ  ISN  OSN   DISK MSG ACT
$.DISP   0000 0000 0000 9000 201C 0000 0000 0000 0000   N/A
$.WASTE  0C00 01AC 0000 8000 0000 0000 0000 0000 0000   NO       YES
```

If the system included remote disk support, there would be another active station, $.PD>IO<. If there were alias stations, these would also still be active. $.WASTE, $.PD>IO<, and alias stations are shut down when you halt the control program:

> CP H $.CF

```
12:46:29 CP71 I 0000 $.CPH    $.CF     HALT REQUEST ISSUED
12:46:34 CP23 I 0000 $.CPP    $.WASTE  STOP REQUEST ISSUED
12:46:34 CF99 I  $.CF HALT COMPLETED
```

You can also use the name $.DISP to halt the control program. When you request a halt of the control program, a program called $.CFSHUT is loaded to determine if there are stations other than $.DISP, $.WASTE, $.PD>IO<, and aliases. The control program is not halted until these are the only stations. If there are other stations, this message is logged approximately once a minute:

```
*12:47:45 CF97 E 0000 $.CFSHUT WARNING-NO CHANGE IN # OF STATIONS IN STORAGE
```

Restarting the Communications Facility

If a halt of the control program is pending because there are active stations, you can restart the system either by starting any station or by issuing the *CPRSTART* command.

Because program $.CFSHUT recognizes the CPRSTART command, your terminal must be assigned to the same partition as $.CFSHUT. Issue the EDX $A command to find where $.CFSHUT is running, issue the EDX $CP command to change partitions if necessary, and then restart the system:

> CPRSTART

```
*12:50:35 CF98 E 0000 $.CFSHUT CANNOT HALT-RESTARTING
```

Preparing Data Sets to Start the Communications Facility

If data sets $.SYSIPL and $.SYSPD contain the commands required to start your Communications Facility configuration, you don't have to issue those commands individually every day. All you would need to do is load the control program, $.CFS or $.CFD, and start the program dispatcher program, $.PD. If you include the S $.PD command in the $.SYSIPL data set, you need only load the control program.

Preparing $.SYSIPL

The system initialization data set, $.SYSIPL, contains commands that are processed when you load the control program, $.CFS or $.CFD.

Use any EDX text editor to prepare the $.SYSIPL data set. Enter one command per line and a /* after the last command. For example:

SET LOG $SYSPRTR
S HOST CU71 T770 T4978
S $.PD
/*

Preparing $.SYSPD

The program dispatcher data set, $.SYSPD, contains information that is processed when you start the program dispatcher program, $.PD. Use any EDX text editor to prepare the $.SYSPD data set. The data set contains four groups of statements, each terminated by a /*:

Commands
/*
Path definitions
/*
Transaction definitions
/*
Transactions
/*

See the *Design and Installation Guide* for additional information on the $.SYSPD data set and for a description of the path and transaction definition statements.

Note that you can use PD commands to modify the path definition and transaction tables, which are built from information in the $.SYSPD data set when the program dispatcher is started. These modifications are temporary. Permanent changes to $.SYSPD can only be made by editing the data set with one of the EDX editors. The command that inserts entries in the table can be used only if space for additional entries has been allocated by means of RESERVE statements in $.SYSPD.

Starting the Communications Facility at IPL

You can cause the Communications Facility to be started when you IPL a Series/1. There are two ways to do this:

- Rename $.CFS or $.CFD to $INITIAL with the EDX utility program $DISKUT1.

- Create a small program called $INITIAL that loads $.CFS or $.CFD.

Both methods cause the control program to be loaded at IPL time, and both methods require that the partition into which the control program is loaded be mapped.

Once loaded, the control program issues a CP READ command to read and process the commands in the $.SYSIPL data set. Assuming the S $.PD command was included in $.SYSIPL, all the commands, path and transaction definitions, and transactions contained in the $.SYSPD data set will also be processed. In this way you can start your Communications Facility system automatically when you IPL.

Renaming the Message Dispatcher

After you rename $.CFD or $.CFS to $INITIAL, EDX will load it into partition 1 (or the partition specified on the EDX SYSTEM statement) when you IPL. (The partition must be mapped.) If you want the control program in a different partition, create a $INITIAL program.

Creating a $INITIAL Program

After you create a program like the one shown below, two things happen when you IPL. First, EDX loads $INITIAL and second, $INITIAL loads $.CFS or $.CFD into the partition you specified. (The partition must be mapped.) A sample $INITIAL program follows:

```
$INITIAL   PROGRAM   BEGIN
BEGIN      LOAD      $.CFD,PART=7
           PROGSTOP  LOGMSG=NO
           ENDPROG
           END
```

In this example, $INITIAL loads $.CFD into partition 7.

Starting a Network

You can start an entire network from one Series/1 in which the program dispatcher is running by sending an IPL transaction to each of the other Series/1s in the network. The Communications Facility will become operational in the remote systems, provided they have a $INITIAL program and $.SYSIPL data set as previously described.

You can put the IPL transactions in the $.SYSPD data set or issue the *PD TRAN* command to send them. If you put them in $.SYSPD, you must be careful about stopping and then restarting the program dispatcher in the host Series/1; each restart of the program dispatcher will cause a reIPL of the remote systems.

Halting the Communications Facility Without an Operator

You can cause the system to shut down when no operator is present by scheduling transactions that cause data sets containing halt commands to be read at the desired time.

You do this by scheduling a transaction to send *SYST CP*, a system transaction, to do a CP READ. The CP READ command will read a data set containing the halt commands for the network.

For example, assume the network shown in Figure 6 on page 17 is to be shut down at midnight.

You need a data set at each cell that contains the halt commands for its configuration. The data set at cell SJ, called SHUTSJ, contains:

```
H $.IO0AE0 $.IO0670 $.IO0AC0 $.PD $.CF
/*
```

Cell S2 has a similar data set called SHUTS2, and cell S3 has one called SHUTS3.

You also need a data set called SHUTNET at your cell (assume it's SJ) that contains these PD commands:

```
PD CP S2 READ SHUTS2
PD CP S3 READ SHUTS3
/*
```

Send the following transactions (they can be in your $.SYSPD data set) to schedule the shutdown:

```
SCHDSJSYSTSJ3000000CP READ SHUTNET
SCHDSJSYSTSJ3000500CP READ SHUTSJ
```

At midnight, the first SCHD transaction executes. It sends a SYST CP transaction to read data set SHUTNET. The CP READ command causes the commands in the data set to be processed. The commands, which send CP READ commands to the remote cells, cause them to be shut down.

The second SCHD transaction executes at five minutes after midnight, which allows time for the flow of messages from the remote cells to stop. It causes data set SHUTSJ to be read, which shuts down cell SJ.

Modifying Your System Configuration

You can modify parts of the system while the rest of the system is operating. You can, for example:

- Add more stations with the *CP DEF* command. This command creates a station definition in $.SYSNET, the network configuration data set.
- Modify a station definition in $.SYSNET with the *CP F* command. If the station is active when the command is issued, its station block is also modified.
- Change the connection between stations with the *CP LINK* command.
- Insert an entry in the path table so the program dispatcher can route transactions to a new cell with the *PD I PATH* command.
- Insert an entry in the transaction table with the *PD I TID* command.
- Change entries in the path and transaction tables with the *PD F PATH* and *PD F TID* command.

For example, assume your system configuration matches the one in Figure 6 on page 17 and a new Series/1 has been added to the network. The new Series/1 is attached to the same line as Series/1-A and Series/1-B. It has already been IPLed, and the Communications Facility, including the program dispatcher, is operational. Now you, the operator at Series/1-M, want to modify your system to recognize the new Series/1. First you must define the emulated terminals that the program dispatcher will use to route transactions to the new Series/1:

```
> CP DEF CU19C 01C9 TERM 3271 POLL19 C47F 0000
> CP DEF C770C 01CA TERM 3277 POLL19 C440 E440
```

Now you can start them:

```
> CP S CU19C C770C
```

Next add an entry in the path table so the program dispatcher can route transactions to the new Series/1, cell S4:

```
> PD I PATH C770C,S4
```

Now the new Series/1 is included in the network and can communicate with the other cells.

Finally, you need to make the path entry permanent by adding a path definition statement to data set $.SYSPD. The PD Insert and PD Modify commands alter the program dispatcher tables in processor storage. They do not alter $.SYSPD.

Managing Message Queues

Utility program $.UT2 allows you to manage message queues. You can obtain a report of messages in a disk queue, and you can purge messages from a disk or storage queue. When you purge messages, you can also obtain a report of them, and you can send each message on to the destination you specify. You can use most of the functions of $.UT2 with any station's message queue. A few of the functions can be used only with the undeliverable message queue, $.WASTE.

Controlling Remote Series/1s

The Communications Facility provides two utility programs, $.HMU and $.RMU, to help you manage the remote Series/1s in your network. With these utilities you can allocate or delete a data set, execute a program, send and receive data sets, dump storage to disk, and communicate with remote programs from your terminal.

Controlling a Remote Disk Volume Directory

Utility program $.PDBSTS allows you to gain exclusive control of a disk volume directory in a remote Series/1 before executing a program that would update that directory. This prevents another program from accessing the remote disk volume you're using.

Altering Space for Stations with Disk Queuing

Utility program $.SETSTG allows you to alter the size of the space used for disk-queue file control blocks and the message buffer pool (CFBUF) if your Communications Facility system uses disk queuing for messages. You will need to do this if your configuration has more than ten stations with disk queues active concurrently.

Initializing Disk-Queue Data Sets

Utility program $.DSINIT initializes data sets to be used for disk queuing of messages. These data sets must be allocated by the EDX utility program $DISKUT1 and initialized by the $.DSINIT utility program.

Operator Command Reference

This chapter describes each of the Communications Facility operator commands. They are presented in alphabetical order by command verb. The command description includes the command function, its format, and an example of its use.

The majority of the commands are prefixed with CP or PD; commands that do not begin with CP or PD are the CPRSTART, GOTEST, TRAN, and WSC commands. They are merged with the CP and PD commands alphabetically.

Command Format

The command format for CP commands is:

CP	*command verb*	*parameters*

The *command verb* specifies the command function to be performed.

Parameters are separated by a blank except where a comma is indicated. The parameters are positional and must be entered in the order given. It is not necessary to specify all the parameters for all commands.

The program dispatcher PD commands are a subset of the command processor CP commands. The format of PD commands is:

CP	**PD**	*command verb*	*parameters*

For simplicity, you may omit CP and use this format:

PD	*command verb*	*parameters*

The *command verb* specifies the command function to be performed.

Parameters are separated by a blank except where a comma is indicated. The parameters are positional and must be entered in the order given. It is not necessary to specify all the parameters for all commands. However, in some instances, you must use commas as placeholders for the parameters not specified.

SYNTAX Conventions

Figure 11 shows the conventions this chapter uses in describing the syntax of each command.

CAPITALS	Parameters in bold **CAPITAL** letters must be entered exactly as shown. For example: **CP F** *T3277* **LINE** *TSO*
,	Commas must be coded exactly as shown when they occur.
[ ]	Brackets enclose optional data. You can enter the data or omit it. For example, the command **CP HELP** [*command*] gives you the option of specifying a command's functions to be displayed. If omitted, all command processor commands are displayed.
{ }	Braces enclose two or more items. You may choose one of these items. Items are separated by vertical bars (|). For example: **PD TRAC {ON | OFF}** means you must specify if the trace is to be ON or OFF.
<u>BOLD UNDERLINED</u>	Bold type underlined items in capital letters indicate options selected by default if you don't enter the keyword parameter with one of the options. For example: {*part* | **<u>0</u>**} means you may optionally specify either *part* or **<u>0</u>**. If you don't, the bold type, underlined item, in this case 0, is assumed by default.
italics	Parameters in italics mean you must substitute a value. For example: **PD R P** *cell-id* means you must supply the cell ID.
...	Ellipsis points indicate that the item preceding them may be repeated two or more times. Separate the items with blanks. For example: *station-name1...station-namen* means you may enter one or more station names.

Figure 11. Syntax Notation

CP DEF ALIAS—Define Alias Station

This command defines an alias name for a station. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **AL[IAS]** *alias*

station-name
is the original station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

alias
is the alias station name, 1 to 8 alphameric characters.

CP DEF ALIAS Example

```
CP DEF PGM1 019C ALIAS PROGA
```

CP DEF CIRCUIT PVC—Define X.25 Permanent Virtual Circuit Station

This command defines a station representing a permanent virtual circuit. The circuit allows communications with other X.25 defined devices and computers. This command creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **CI[RCUIT]** **PVC** *line-name* **{CF | STD | STD+}** *lci* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

line-name
is the name of the controlling DTE or DCE line station, which must be defined before this station can be defined.

CF | STD | STD+
defines the usage of the station. Specify:

CF — When the circuit connects two Communications Facility systems. Communications Facility message headers are preserved and the Communications Facility message dispatchers control the circuit.

STD — When the circuit is used only for data transfer. Any control messages sent to a standard circuit will be discarded.

STD+ — When the circuit is used for data transfer and can be controlled by an application program.

lci
is the decimal number of the logical channel used by the PVC.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

Special Considerations

Window size and packet size for each circuit default to the values in the line station definition. If you require a packet or window size different from the line definition, use the CP F PKTSIZ or CP F WINDOW command to change the circuit station definition. (You cannot use $.CONFIG to change the circuit station's window or packet size.)

For a DTE line, use the CP F ADDR command to add the X.25 network address to the station definition.

CP DEF CIRCUIT PVC Examples

CP DEF CIRC1 01C1 CIRCUIT PVC LINE1 CF 104

This command defines a circuit station (CIRC1) whose messages are routed by the Communications Facility message dispatcher. The IOCP sends the messages over the logical channel 104.

CP DEF CIRC2 01E8 CIRCUIT PVC LINE2 STD 6

This command defines a circuit station (CIRC2) that sends only data messages. The IOCP sends the messages over the logical channel 6.

CP DEF CIRCUIT SVC—Define X.25 Switched Virtual Circuit Station

This command defines a station representing a switched virtual circuit. The circuit allows communications with other X.25 defined devices and computers. The physical connection is through an HDLC line into an X.25 network. This command creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **CI[RCUIT]** **SVC** *line-name* **{CF | STD | STD+}** **{WAIT** {*call-id* | **0**}[,*prot-id*] | **INIT** *call-id*[,*prot-id*] | **USERINIT}** [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

line-name
is the name of the controlling DTE or DCE line station, which must be defined before this station can be defined.

CF | STD | STD+
defines the usage of the station. Specify:

CF When the circuit connects two Communications Facility systems. Communications Facility message headers are preserved and the Communications Facility message dispatchers control the circuit.

STD When the circuit is used only for data transfer. Any control messages sent to a standard circuit will be discarded.

STD+ When the circuit is used for data transfer and can be controlled by an application program.

WAIT | INIT | USERINIT
defines the action you want the Communications Facility to take when this circuit station is started. Specify:

WAIT If you want the station to wait for an incoming call. By specifying the *call-id* and/or the *prot-id* parameter, you tell the

IOCP whether to wait for a specific call or any call. If you select the WAIT option, you must specify a *call-id;* you may specify a *prot-id.*

INIT — If you want the Communications Facility to initiate a call (send a call request packet) to a specific network address when this circuit is opened. If you select INIT, you must also specify *call-id.* You may specify a *prot-id.*

USERINIT — If the call request is initiated by an application program. The IOCP waits for the call request control message before initiating a call. You cannot specify USERINIT if you specified the usage as CF.

call-id

is a two-digit call ID that is defined in the $.SYSX25 data set. If you specified a contact type of WAIT, the station waits until it receives a call from the network address represented by this call ID. If the station accepts calls from any network address, specify 0 instead of a call ID. If you specified a contact type of INIT, the IOCP sends a call request to the network address represented by this call ID when the station is started. You must specify this parameter if the contact type is WAIT or INIT. Do not specify this parameter for a contact type of USERINIT.

prot-id

is the optional 8-digit hexadecimal protocol identifier. It is valid only for contact types of WAIT and INIT. If the contact type is WAIT, the IOCP accepts only incoming calls that carry this protocol ID. If the contact type is INIT, the IOCP sends this protocol ID on the call request packet as the first 4 bytes of user data. (These 4 bytes are in addition to the user data field you define with the CP F USER command of with $.CONFIG.)

The first 2 bits of the prot-id are significant to public data networks. Depending on their value, the protocol identifier and the user data field will be used in accordance with the specifications of the listed bodies:

00—Recommendation X.29
01—Network Administrations
10—International User Bodies
11—No constraints

A protocol identifier whose first 2 bits are other than 11 may cause a protocol to be implemented within public data networks.

dsname

is the name of the data set to be used for disk queuing.

vol-id

is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N

specifies whether or not disk queuing is to be active when the station is started.

Special Considerations

Window size and packet size for each circuit default to the values in the line station definition. If you require a packet or window size different from the line definition, use the CP F PKTSIZ or CP F WINDOW command to change the circuit station definition. (You cannot use $.CONFIG to change the circuit station's window or packet size.)

If you have subscribed to X.25 network facilities, you can assign them to the circuit station by using the CP F FAC command or $.CONFIG. If you have specified a contact type of WAIT, you may assign facilities only if the station's usage type is STD+.

If you specify a contact type of INIT, you may send up to 12 bytes of user data on the call request. Use the CP F USER command or $.CONFIG to add this data to the circuit definition.

CP DEF CIRCUIT SVC Example

CP DEF CIRC3 21CB CIRCUIT SVC LINE3 STD INIT 24,C000C3C2

This command defines a circuit station (CIRC3) that receives and sends only data messages. When the circuit is started, the IOCP initiates a call to the address represented by call ID 24 and sends the protocol ID of C000C3C2 as the first 4 bytes of call user data.

CP DEF CIRC4 04F8 CIRCUIT SVC LINE4 STD+ WAIT 0,C7C1E3C5

This command defines a circuit station (CIRC3) that receives and sends data and control messages. When the station is started, the IOCP waits for a call from the network. The IOCP accepts calls from any network address if they carry the protocol ID C7C1E3C5 (GATE in EBCDIC).

CP DEF DEVICE PRINTER—Define Printer Device Station

This command defines a station representing a 4973, 4974, 4975, 5219, 5224, or 5225 printer. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **DE[VICE]** **PRINTER** **{Y | N}** *buffsize* *poll* *sel* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00, specify FF.

Y | N
specifies whether or not the printer is to be supported as a buffered device. If you specify Y, a member with the same name as the station is allocated in data set $.SYSPRT.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 800 to 7FDE (2048 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide*.

poll
is the 4-digit hexadecimal station polling address. Specify any hexadecimal value other than 0000.

sel
is the 4-digit hexadecimal station selection address. Specify any hexadecimal value.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF DEVICE PRINTER Example

CP DEF PRINT 0101 DEVICE PRINTER N 9C4 4040 0000

CP DEF DEVICE 3101—Define 3101 Device Station

This command defines a station representing a 3101 device attached to the Series/1 through a teletypewriter adapter. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **DE[VICE]** **3101** *buffsize* *poll* *hdcopy* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
: is the station name, 1 to 8 alphameric characters.

na
: is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00, specify FF.

buffsize
: is the size of the buffer required for the station. Specify a hexadecimal value from 800 to F46 (2048 to 3910 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide*.

poll
: is the 4-digit hexadecimal station polling address. Specify any hexadecimal value other than 0000.

hdcopy
: is the name of the EDX terminal to be used as the hardcopy device—the device on which the screen display is printed when you press PA6.

dsname
: is the name of the data set to be used as the station's disk queue.

vol-id
: is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
: specifies whether or not disk queuing is to be active when the station is started.

CP DEF DEVICE 3101 Example

```
CP DEF T3101 0108 DEVICE 3101 9C4 4040 $SYSPRTR
```

CP DEF DEVICE 3101F—Define 3101F Device Station

This command defines a station representing a 3101 or a 7485 device attached to the Series/1 through an asynchronous adapter. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **DE[VICE]** **3101F** *buffsize* *poll* *hdcopy* *bit-rate* **{ON | OFF}** [*dsname*[,*vol-id*]{Y | N}]

station-name
is the station name, 1 to 8 alphanumeric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00 specify FF.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 800 to F46 (2048 to 3910 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide.*

poll
is the 4-digit hexadecimal station polling address. Specify any hexadecimal value other than 0000.

hdcopy
is the name of the EDX terminal to be used as the hardcopy device—the device on which the screen display is printed when you press PA6.

bit-rate
is the bit rate of the feature programmable communications adapter or the multifunction attachment. Specify one of these decimal values for the feature programmable communications adapter: 0, 300, 600, 1200, 2400, 4800, or 9600. Specify one of these decimal values for the multifunction attachment: 0, 1200, 2400, 4800, or 9600. You must specify a bit rate of 9600 for a 7485 terminal. Specify 0 to indicate that the attachment card must get its bit rate from an external clocking device, such as the modem.

ON | OFF
specifies whether or not the IOCP needs to support the auto-answer function. Specify **ON** for remote connections and **OFF** for local connections. You must specify **OFF** for a 7485 terminal.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF DEVICE 3101F Examples

CP DEF T3101F 0168 DEVICE 3101F 9C4 4040 PRINTER2 4800 OFF

This command defines a 3101 terminal (T3101F) on node 01 with a station address of 68, managed as a 3277 terminal. The buffer size is 9C4 (2500 decimal), the polling address is 4040, and the bit rate of the asynchronous adapter is 4800. PRINTER2 is the name of the EDX terminal to be used as the hardcopy device.

CP DEF T3101F 0168 DEVICE 3101F 9C4 4040 PRINTER2 9600 OFF

This command defines a 7485 terminal (T3101F), asynchronous adapter, managed as a 3277 terminal.

CP DEF DEVICE 4978—Define 4978/4980 Device Station

This command defines a station representing a 4978 or 4980 device. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **DE[VICE]** **4978** *buffsize* *poll* *sel* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00, specify FF.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 800 to 7FDE (2048 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide.*

poll
is the 4-digit hexadecimal station polling address. Specify any hexadecimal value other than 0000.

sel
is the 4-digit hexadecimal station selection address. Specify any hexadecimal value.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF DEVICE 4978 Example

```
CP DEF T4978 0114 DEVICE 4978 9C4 4040 0000
```

CP DEF LINE—Define BSC or Channel Attachment Station

This command defines a station representing a BSC line or a channel attachment. This command creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **LI[NE]** **{PTPT | CPU | 3271C | 3271E | CA}** *buffsize* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00, specify FF.

PTPT | CPU | 3271C | 3271E | CA
is the subtype and defines the type of line the station represents. You may specify one of the following:

- PTPT, to define a point-to-point attachment line
- CPU, to define a Series/1-to-Series/1 line
- 3271C, to define a multipoint 3271 control line
- 3271E, to define a multipoint tributary 3271 emulator line
- CA, to define a channel attachment.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 50 to 7FDE (80 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide.*

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF LINE Examples

CP DEF HOST 0109 LINE 3271E 9C4

CP DEF CHANNEL 0122 LINE CA 9C4

CP DEF LINE DxE—Define HDLC Line Station

This command defines a station representing an HDLC line. The line must also be defined to EDX X.25/HDLC Communications Support (XHCS). You use this command to define the Series/1 as data circuit-terminating equipment (DCE) or data terminal equipment (DTE). It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **LI[NE]** **{DCE | DTE}** *buffsize* *pktsize* *window* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 6 alphameric characters. This name must be same as the DDMNAME (a six-character name) that you defined to XHCS using the XHCS utility $XHCUT1.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address. The station address must be the device address pinned on the attachment feature that controls the device. It may not be 00; for a device with hardware address 00, specify FF.

DCE | DTE
is the subtype and defines the type of line the station represents. You may specify one of the following:

- DCE, to define an HDLC line that connects this Series/1 to data terminal equipment (DTE) without an intervening X.25 network. You must have specified DCE for this line when you defined it to XHCS with the utility $XHCUT1. This type of line can support only permanent virtual circuits (PVCs).
- DTE, to define an HDLC line that connects this Series/1 to data terminal equipment through an X.25 network or to a Series/1 defined as a DCE with no intervening X.25 network. You must have specified DCE for this line when you defined it to XHCS with the utility $XHCUT1. This type of line can support permanent virtual circuits (PVCs) and switched virtual circuits (SVCs).

buffsize
is the size of the buffer required for the station. Specify a hexadecimal size from 50 to 7FDE (80 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide*.

pktsize
is a 1- to 4-digit decimal size, in bytes, of the maximum user data field required for a data packet. This is the default for all circuits on this line unless you use the CP F PKTSIZ command to change the circuit station definition. Specify a size of 16, 32, 64, 128, 256, 512, and 1024.

window
is the number of packets that can be sent before an acknowledgment is required. Specify a decimal number from 1 to 7. If you require a window greater than 7, use the CP F WINDOW command to change the line station definition. This is the window size for all the circuits on this line unless you use the CP F WINDOW command to change the circuit station definition.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF LINE DxE Example

CP DEF HDLC1 0109 LINE DTE 80 128 2

This command defines a line station, HDLC1, at network address 0109. This station appears as data terminal equipment to an X.25 network or to another Series/1. The largest data message expected for this line is 128 bytes, the packet size is 128 bytes, and two packets are sent before an acknowledgement is required.

CP DEF LINE LCC—Define Local Communications Controller Line Station

This command defines a station representing a Local Communications Controller device. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **LI[NE]** **LCC** *buffsize* *device-address* [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 50 to 7FDE (80 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide.*

device-address
is the Local Communications Controller device subchannel 0 address, specified as two hexadecimal digits. The value must be from 00 to FC and a multiple of 4.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF LINE LCC Example

CP DEF LCCDEV 01A1 LINE LCC 9C4 F0

CP DEF LU—Define SNA Logical Unit Station

This command defines an SNA logical unit station representing an emulated 3270 terminal. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **LU** *subtype* *puname* *local-address* [{{**1** | **2** | **3**}[,*user-field*] | {**0** | **HOST**}}] [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

subtype
defines the type of emulated terminal the station represents. Specify 3277, 3278, 3279, 3286, 3287, or 3289.

puname
is the name of the controlling SNA physical unit station, which must be defined before this station can be defined.

local-address
is a logical unit number from 1 to 32 defined in the EDX-SNA configuration, or 0 to allow EDX-SNA to assign the next available logical unit number.

0 | 1 | 2 | 3 | HOST
is the identification of an SNA application to be connected to when the user of this station logs on. The SNA applications are defined in the physical unit station; valid values for *id* are 1, 2, 3, or HOST. HOST specifies that the session will be initiated by the host application when the station is started. If the users are to be prompted to select the desired application when they log on, omit this operand; if the station is to have a disk queue, indicate that the operand is null by specifying 0.

user-field
is a 1- to 16-character user data field to be passed to the specified logon application. Do not specify *user-field* when a logon ID of 0 or HOST is used.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF LU Examples

CP DEF SNALU1 01D1 LU 3277 SNAPU 1

CP DEF SNALU2 01D2 LU 3277 SNAPU 24 2,TSOID/PASSWORD

CP DEF SNALU3 01D3 LU 3277 SNAPU 16 0 LU3DQ,MSGVOL N

CP DEF MSG—Define Message Station

This command defines a station representing a message queue. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **{MS[G] | ME[SSAGE]}** [*dsname*[,*vol-id*]**{Y | N}**]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF MSG Example

```
CP DEF XYZ 0199 MSG XYZDQ Y
```

CP DEF NODE—Define Remote Node

This command defines a station representing a remote node. It creates a station definition in $.SYSNET, the network configuration data set. If the remote node is not on the same Local Communications Controller ring as the local node, omit the ring-address, line name, and disk-queuing parameters.

CP DEF	*station-name* *na* **NODE** **{CF | CM}** [*ring-address* *line-name* [*dsname*[,*vol-id*]**{Y | N}**]]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network.

If this node communicates with the Communications Manager, over the Local Communications Controller, the first two digits must be the hexadecimal representation of the alphabetic characters the Communications Manager uses to address this node. For example, if this node is known to the Communications Manager as BB, the node address must be C200.

The rightmost two digits must be 00.

CF | CM
specifies whether the remote node is a Communications Facility (CF) or a Communications Manager (CM) system.

ring-address
is the 2-digit hexadecimal Local Communications Controller ring address. The address must have odd parity and a value, without the parity bit, of 16-125.

line-name
is the name of the controlling Local Communications Controller device station, which must be defined before this station can be defined.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF NODE Examples

CP DEF NODE2 0200 NODE CF 10 LCCDEV

CP DEF CMNODEB C200 NODE CM 10 LCCDEV

CP DEF X25NODE4 0400 NODE CF

CP DEF PU—Define SNA Physical Unit Station

This command defines an SNA physical unit station. It creates a station definition in $.SYSNET, the network configuration data set. Up to three SNA applications may be defined for the station. Users communicating with associated SNA logical unit stations may select one of these applications when they log on. The applications are defined by means of the CP F (Modify) command, as described later in this chapter.

CP DEF	*station-name* *na* **PU** **3274ES** *buffsize* *pu-number*[2]

station-name
is the station name, 1 to 8 alphanumeric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

buffsize
is the size of the buffer required for the station. Specify a hexadecimal value from 50 to 7FDE (80 to 32734 decimal). If you need information about I/O buffer sizes, see the *Design and Installation Guide*.

pu-number
is the one-digit EDX SNA PU number associated with this station.[2]

CP DEF PU Examples

```
CP DEF SNAPU 01D0 PU 3274ES 9C4 1
```

This command defines an SNA physical unit station (SNAPU) with a buffer size of 9C4 (2500 decimal) and an EDX SNA PU number of 1.[2]

---

[2] This field is valid only if you've installed the EDX Secondary SNA2 support.

61

CP DEF TERM—Define Terminal Station

This command defines a station representing a real or emulated 3270 terminal accessed from a BSC line or an emulated 3270 terminal accessed from a channel attachment. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **TERM** *subtype* *line-name* {*poll select* | *port-number*} [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

subtype
defines the type of terminal the station represents. Specify 3271, 3277, or 3286 for a terminal accessed from a BSC line. Specify PORT for a terminal accessed from a channel attachment.

line-name
is the name of the controlling 3271C, 3271E, or CA line station, which must be defined before this station can be defined.

poll
is the 4-digit hexadecimal station polling address. Appendix A contains the values to be specified. Specify this parameter for terminal subtypes 3271, 3277, and 3286.

select
is the 4-digit hexadecimal station selection address. Appendix A contains the values to be specified. Specify this parameter for terminal subtypes 3271, 3277, and 3286.

port-number
is a decimal port number, from 0 to 31. Specify this parameter for terminal subtype PORT.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF TERM Examples

CP DEF S1CRT1 01C0 TERM 3277 HOST 4040 6040

CP DEF S1CRT8 01C2 TERM PORT CHANNEL 8

CP DEF USER—Define User Station

This command defines a station representing either an application program or an I/O control program. It creates a station definition in $.SYSNET, the network configuration data set.

CP DEF	*station-name* *na* **USER** [*dsname*[,*vol-id*]{**Y** | **N**}]

station-name
is the station name, 1 to 8 alphameric characters.

na
is the 4-digit hexadecimal network address. The leftmost two digits are the node assignment. A node assignment is a hexadecimal value from 01 to FF representing a single Series/1 in your network. Each Series/1 in your network must have a different node assignment. If you have only one Series/1 in your network, the node assignment can be any hexadecimal value from 00 to FF. The rightmost two digits are the station address within a node and may be any hexadecimal value from 01 to FF. Each station address within a node must be unique.

dsname
is the name of the data set to be used for disk queuing.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

Y | N
specifies whether or not disk queuing is to be active when the station is started.

CP DEF USER Example

CP DEF PROGA 018A USER MSGFILE,EDX003 Y

CP F ADDR—Change Device, DTE Line, Ring, or Local Address

This command changes one of the following addresses:

- the device address of a Local Communications Controller device station
- the ring address of a Local Communications Controller node station
- the local address of an SNA logical unit station
- the X.25 network address of a DTE line station.

The command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **ADDR** {*device-address* | *ring-address* | {*local-address* | **0**} | {*X25-addr* | **0**}}

station-name
is the station name, 1 to 8 alphameric characters.

device-address
is the Local Communications Controller device subchannel 0 address, specified as two hexadecimal digits. The value must be from 00 to FC and a multiple of 4.

ring-address
is the 2-digit hexadecimal Local Communications Controller ring address. The address must have odd parity and a value, without the parity bit, of 16-125. To indicate that the node is not on the same Local Communications Controller ring as the local node, specify 00.

local-address
is a logical unit number from 1 to 32 defined in the EDX-SNA configuration, or 0 to allow EDX-SNA to assign the next available logical unit number.

X25-addr
is a 1- to 15-digit X.25 network address of this DTE line as defined in the X.25 network. This address is put into the calling address field of call request packets sent by circuits on this line. To delete an existing address, specify 0.

CP F ADDR Examples

```
CP F LCCDEV ADDR F8

CP F NODE2 ADDR 1C

CP F SNALU1 ADDR 2

CP F X25LN1 ADDR 220412348765
```

CP F APPLn—Change SNA Application Name

This command defines or changes the name of one of the three SNA applications that users can select when they log on. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified. The command must be addressed to an SNA physical unit station.

CP F	*station-name* **{APPL1 | APPL2 | APPL3}** [*application name*]

station-name
is the station name, 1 to 8 alphameric characters.

application name
is a 1- to 8-character SNA application name. It must match an application name defined in the SNA host. To delete an existing application name, omit the parameter.

CP F APPLn Example

```
CP F SNAPU APPL1 TSO
```

CP F BUFF—Change Buffer Size

This command changes the buffer size of a line, device, or SNA physical unit station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified. If you need information about I/O buffer sizes, see the *Design and Installation Guide.*

CP F	*station-name* **BUFF** *size*

station-name
is the station name, 1 to 8 alphameric characters.

size
is the size of the buffer required for the station. Specify a hexadecimal value.

CP F BUFF Example

CP F S1 BUFFER 600

CP F CALLID—Change SVC Circuit Station's Call ID

This command changes the call ID of an SVC circuit station. This command is not valid for stations defined with a contact type of USERINIT. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified; however, the new call ID is not in effect until the circuit is reopened.

CP F	*station-name* **CALLID** {*call-id* | **0**}

station-name
is the circuit station name, 1 to 8 alphameric characters.

call-id
is a 2-digit call ID that is defined in the $.SYSX25 data set. If you specified a contact type of WAIT, the station waits until it receives a call from the network address represented by this call ID. Specify 0 to accept calls from any network address. 0 can only be specified for a contact type of WAIT. If you specified a contact type of INIT, the IOCP sends a call request to the network address represented by this call ID when the station is started.

CP F CALLID Example

```
CP F CIRC1 CALLID 7
```

CP F DISKQ—Change Disk-Queuing Status

This command changes a station's disk-queuing status to active or inactive. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **DISKQ** {**AC** | **IN**}

station-name
is the station name, 1 to 8 alphameric characters.

AC | IN
specifies the new disk-queuing status as active (AC) or inactive (IN).

CP F DISKQ Example

```
CP F XYZ DISKQ IN
```

CP F FAC—Add or Change SVC Circuit Station's Facilities

This command adds or changes a circuit station's optional user facilities. Facilities are invalid for PVC circuits and SVC circuits that have a contact type of WAIT and a usage of CF or STD. If you use this circuit station to communicate through an X.25 network, you may choose only those facilities to which you have subscribed from your X.25 network. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified; however, the facilities are not in effect until the circuit is reopened.

CP F	*station-name* **FAC** *facilities-list*

station-name
is the circuit station name, 1 to 8 alphameric characters.

facilities-list
is one or more of the following facilities:

CUG *nn* Closed user group. *nn* is the 2-digit decimal ID of the closed user group of which this station is a member.

BCUG *nnnn* Bilateral closed user group. *nnnn* is the 4-digit decimal ID of the group of which this station is a member.

RPOA *nnnn* Recognized private operating agency. This facility allows you to specify a particular RPOA transit network through which the call is to be routed internationally, when more than one RPOA transit network exists at an international gateway. *nnnn* is the 4-digit decimal data network identification code for the requested RPOA transit network.

FS Fast select. Fast select allows up to 128 bytes of data to be carried on a call request, call accept, or call clear packet.

FSR Fast select restricted. This allows up to 128 bytes of data to be carried on a call request or a call clear packet; call accept is not allowed in response to the call request.

REV Reverse charging. The DTE accepting the call is charged for the call instead of the DTE that places the call.

Special Considerations

If the circuit has a contact type of USERINIT, any facilities you select with this command may be overridden in a call request control message.

CP F FAC Example

```
CP F CIRC1 FAC CUG 14 FS REV
```

This command adds a closed user group ID to CIRC1's station definition. Up to 128 bytes of data may be sent on the call establishment packets; and the DTE that accepts the call must pay for it.

CP F HDCOPY—Change Hard Copy Device

This command changes the hard copy device of a 3101 or 3101F device station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **HDCOPY** *device-name*

station-name
is the station name, 1 to 8 alphameric characters.

device-name
is the name of the EDX terminal to be used as the hard-copy device—the device on which the screen display is printed when you press the PA6 key.

CP F HDCOPY Example

CP F T3101 HDCOPY PRTR1

CP F HOLD—Control Message Traffic

This command prohibits or allows message traffic to or from a station. When message traffic to a station is prohibited, messages sent to it are retained, but the station cannot receive them. When message traffic from a station is prohibited, messages sent by it are discarded. The command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **HOLD** **{IN | OUT}** **{ON | OFF}**

station-name
is the station name, 1 to 8 alphameric characters.

IN | OUT
IN specifies message traffic from the station into the system.

OUT specifies message traffic to the station from the system.

ON | OFF
ON prohibits message traffic in the specified direction.

OFF allows message traffic in the specified direction.

CP F HOLD Examples

CP F STABC HOLD IN ON

CP F STXYZ HOLD OUT ON

CP F LCI—Change PVC Circuit Station's Logical Channel Identifier

This command changes a permanent virtual circuit (PVC) station's logical channel identifier (LCI). This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified; however, the new LCI is not in effect until the circuit is reopened.

CP F	*station-name* **LCI** *lci*

station-name
is the circuit station name, 1 to 8 alphameric characters.

lci
is the decimal number of the logical channel used by the PVC. Specify a value between 1 and 4095. If this circuit is used for sending messages through an X.25 network, you must have subscribed to this LCI from the network.

CP F LCI Example

```
CP F CIRC1 LCI 22
```

CP F LINE—Change Controlling Line

This command changes the controlling line of a terminal, circuit, or node station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **LINE** *line-name*

station-name
is the terminal, node, or circuit station name, 1 to 8 alphameric characters.

line-name
is, for terminal stations, the name of a 3271C, 3271E, or CA line station; for circuit stations, the name of a DCE or DTE line station; for node stations, the name of an LCC line station or 0 to specify that the node is not on the same Local Communications Controller ring as the local node.

Special Considerations

You should need to change an active station definition only when you have used CP F NA to change the controlling line's network address.

CP F LINE Example

```
CP F T3277 LINE TSO
```

CP F LOG—Change SNA Logon ID

This command changes the SNA logon ID of an SNA logical unit station. The logon ID is the identification of an SNA application to be connected to when the user of this station logs on. The command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **LOG** *id*

station-name
is the station name, 1 to 8 alphameric characters.

id
specify 1, 2, or 3 to select one of the SNA applications defined for the SNA physical unit station. Specify 0 to indicate that users are to be prompted to select the desired application when they log on. Specify HOST to indicate that a host-initiated session is to be started when the LU station is started.

CP F LOG Example

```
CP F SNALU2 LOG 3
```

CP F MODE—Change Mode of Operation

This command changes the mode of operation for various types of stations. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is started when the command is issued, its station control block is also modified. If you need additional information about the modes of operation, see the *Design and Installation Guide*.

CP F	*station-name* **MODE** *mode*

station-name
: is the station name, 1 to 8 alphanumeric characters.

mode
: specifies the mode of operation. Although a station type may support multiple modes, you can change only one mode at a time. You can abbreviate the mode, as shown below, or specify additional characters. For example, alphanumeric mode may be specified as ALPHA, ALPHANUMERIC, or any string of characters beginning with ALP. The modes that you can specify for various station types are:

REM | NON-R
: specifies whether or not the remove option is in effect. Valid only for I/O control programs (user station).

XON | XOF
: specifies whether or not transparent write is in effect. Valid only for point-to-point BSC lines (line station, subtype PTPT).

PRI | SEC
: specifies whether the Series/1 is the primary or secondary station. Valid only for point-to-point BSC lines (line station, subtype PTPT).

BAS | REC
: specifies that messages are processed in basic or record mode. Valid only for terminal stations, SNA LU stations, and point-to-point BSC lines (line station, subtype PTPT).

UBS[TOP] | UBR[ETRY]
: specifies whether to stop the station or retry the session when an auto-logon LU session is unbound. Valid only for SNA LU stations.

WAIT | NOW[AIT]
: specifies whether, on a NETINIT, the EDX SNA support waits for the LU-to-SSCP session to be established. Valid only for SNA LU stations.

NOP[C] | PC
: specifies whether messages from an LU station are sent through a Series/1 PC Connect Attachment or to other station types (NOPC). Valid only for SNA LU stations.

STOP | RETRY
specifies whether to stop (STOP) or reopen (RETRY) a circuit station with contact type of INIT when its circuit is closed. Valid only for circuit stations.

NON-D | TEX
specifies non-display or text mode. Valid only for device stations managed by $.IO0674 and 3271 terminal stations.

3270 | SCS
specifies 3270 data stream mode or SNA character string mode. Valid only for printers (device station, subtype PRINTER).

UC | MC
specifies uppercase mode or mixed case mode. Valid only for printers (device station, subtype PRINTER).

ALP | NUM
specifies alphanumeric or numeric mode. Valid only for 4978 device stations.

UL | LU | UU
specifies upper/lower, lower/upper, or upper/upper mode. Valid only for 4978 device stations.

AUT ON | OFF
specifies whether or not the IOCP needs to auto-answer. Specify ON for switched lines. Specify OFF for a local line or a switched line with manual answer or modem auto-answer. Do not specify ON for 7485 devices, since they must be connected with the local RS-422 lines.

BIT 0 | 300 | 600 | 1200 | 2400 | 4800 | 9600
specifies the bit rate for 3101s and 7485s. Valid only for 3101F stations. Specify 0 to indicate that the attachment card must get the bit rate from an external clocking device, such as the modem.

CP F MODE Examples

```
CP F S1CRT1 MODE REC
```

This command changes the mode of operation of S1CRT1 to allow for messages to be processed in record mode.

```
CP F F3101 MODE BIT 9600
```

This command changes the bit rate of the 3101F device, F3101.

```
CP F F3101 MODE AUT ON
```

This command specifies support for auto-answer is on.

CP F LU1PRTR MODE SCS

This command changes the mode of operation of a printer device, LU1PRTR, by specifying an SNA character string mode.

CP F MODEn—Change SNA Application Mode

This command defines or changes the mode of one of the three SNA applications that users can select when they log on. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **{MODE1 | MODE2 | MODE3}** [*mode*]

station-name
is the station name, 1 to 8 alphameric characters. The station must be an SNA physical unit station.

mode
is a 1- to 8-character SNA application mode. It must match a mode defined in the SNA host. To delete an existing application mode, omit the parameter.

CP F MODEn Example

```
CP F SNAPU MODE1 D3270
```

CP F NA—Change Network Address

This command changes the network address of a station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **NA** *address*

station-name
is the station name, 1 to 8 alphameric characters.

address
is the new network address, specified as four hexadecimal digits. For a node station, the rightmost two digits must be 00. For all other types of stations, the rightmost two digits must not be 00.

Special Considerations

Although links are specified by station name, they are recorded in $.SYSNET and station blocks as network addresses. Therefore when you change the network address of a station, you must redo any links to that station.

Use the CP LINK command to redo direct or alternate links. If you change the network address of a PU station, use the CP F PU command to put the new address in the associated LU stations. If you change the network address of a 3271C, 3271E, CA, DCE, DTE, or LCC line station, use the CP F LINE command to put the new address in the associated terminal, circuit, or node station.

CP F NA Example

```
CP F S1 NA 010A
```

CP F NAME—Change Station Name

This command changes the name of a station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **NAME** *new-name*

station-name
is the current station name, 1 to 8 alphameric characters.

new-name
is the new station name, 1 to 8 alphameric characters.

CP F NAME Example

CP F STABC NAME STXYZ

CP F NODE—Change Node System Type

This command changes the communication system type of a node station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **NODE** **{CF | CM}**

station-name
is the station name, 1 to 8 alphameric characters.

CF | CM
specifies whether the node is a Communications Facility (CF) or a Communications Manager (CM) system.

CP F NODE Example

```
CP F NODE2 NODE CM
```

CP F PKTSIZ—Change Line or Circuit Station's Packet Size

This command changes the packet size of a circuit or DTE line station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified. However, the new packet size is not in effect until the circuit is reopened.

CP F	*station-name* **PKTSIZ** *pktsize*

station-name
is the circuit or line station name, 1 to 8 alphameric characters.

pktsize
is the 1- to 4-digit decimal size of the maximum user data field required for a data packet. If the circuit or line is used to send messages through an X.25 network, the packet size must be one subscribed to from the X.25 network. Specify 16, 32, 64, 128, 256, 512, or 1024. To delete an existing packet size in a circuit station, omit the parameter. The circuit's packet size then defaults to the controlling line station's packet size.

CP F PKTSIZ Examples

```
CP F LINE1 PKTSIZ 512
```

This command defines a packet size of 512 bytes. 512 is now the default for any of this line's circuit stations that do not have packet sizes in their station definitions.

```
CP F CIRC2 PKTSIZ
```

This command removes a packet size from the circuit station's definition. This circuit now uses its controlling line station's packet size.

CP F POLL—Change Polling Address

This command changes the polling address of a terminal station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is started when the command is issued, its station control block is also modified.

CP F	*station-name* **POLL** *address*

station-name
is the station name, 1 to 8 alphanumeric characters.

address
is the 4-digit hexadecimal station polling address. Appendix A, "3270 Polling and Selection Address Table" on page 331 contains the values to be specified for terminal stations.

CP F POLL Example

```
CP F TERM1 POLL 40C6
```

This command changes the polling address to 40C6.

CP F PORT—Change Port Number

This command changes the port number of a terminal station controlled by a channel attachment line. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **PORT** *number*

station-name
is the station name, 1 to 8 alphameric characters.

number
is the decimal port number, 0-31.

CP F PORT Example

```
CP F S1CRT8 PORT 2
```

CP F PROTID—Change SVC Circuit Station's Protocol Identifier

This command changes an SVC circuit station's protocol identifier. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified; however, the new protocol ID does not take effect until the circuit is reopened.

CP F	*station-name* **PROTID** [*prot-id*]

station-name
is the circuit station name, 1 to 8 alphameric characters.

prot-id
is the optional 8-digit hexadecimal protocol identifier. It is valid only for contact types of WAIT and INIT. If the contact type is WAIT, the IOCP accepts only incoming calls that carry this protocol ID. If the contact type is INIT, the IOCP sends this protocol ID on the call request packet as the first 4 bytes of user data. (These 4 bytes are in addition to the user data field you define with the CP F USER command or with $.CONFIG.) To delete an existing protocol ID, omit the parameter.

The first 2 bits of the *prot-id* are significant to public data networks. Depending on their value, the protocol identifier and the user data field will be used in accordance with the specifications of the listed bodies:

00—Recommendation X.29
01—Network Administrations
10—International User Bodies
11—No constraints

A protocol identifier whose first 2 bits are other than 11 may cause a protocol to be implemented within public data networks.

CP F PROTID Example

CP F CIRC1 PROTID 0000145C

CP F PU—Change Controlling SNA Physical Unit Station

This command changes the controlling SNA physical unit station of an SNA logical unit station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is started when the command is issued, its station control block is also modified.

CP F	*station-name* **PU** *puname*

station-name
is the station name, 1 to 8 alphanumeric characters.

puname
is the name of an SNA physical unit station.

Special Considerations

If you use CP F NA to change the controlling PU station's network address, you must use the CP F PU command to reset the controlling PU of all LU stations that are controlled by that PU.

CP F PU Example

```
CP F SNALU1 PU SNAPU2
```

This command changes the physical unit station's controlling station SNALU1, to SNAPU2.

CP F PUNUM—Change SNA Physical Unit Number

This command is valid only if you have installed the Communications Facility EDX Secondary SNA2 support.

This command changes the number of the EDX-SNA physical units associated with a SNA physical unit station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is started when the command is issued, its station control block is also modified.

CP F	*station-name* **PUNUM** *pu-number*

station-name
is the station name, 1 to 8 alphanumeric characters.

pu-number
is the one-digit EDX-SNA PU number (1 to 4) associated with this station.

Special Considerations

If you use the CP F PUNUM command to change a physical unit station's number while the LU stations controlled by it are active, the LU stations will not use the new PU number until they are stopped and restarted.

CP F PUNUM Example

```
CP F SNAPU1 PUNUM 1
```

This command changes the EDX-SNA physical unit associated with the SNA physical unit station, SNAPU1, to 1.

CP F RETRY—Change Control Unit Station's Retry Maximum

This command specifies the maximum number of times a control unit's device station can time out before it becomes incactive. It modifies the station's definition in $.SYSNET, the network configuration data set. If the station is started when the command is issued, its station control block is also modified. However, the new retry maximum is not in effect until the associated line station is halted and restarted.

CP F	*station-name* **RETRY** *maximum*

station-name
is the control unit's station name, 1 to 8 alphanumeric characters.

maximum
is the maximum number of times that the IOCP, $.IO0AC0, allows a poll to the station to time out. Specify 0 if the control unit station definition has not been modified with the CP F RETRY command. A value of 0 keeps the control unit station active regardless of how many times a poll may time out. A value of 0 is used if the control unit station definition has not been modified with the CP F RETRY command.

CP F RETRY Example

CP F CU1 RETRY 10

This command sets the retry maximum to 10. The IOCP, $.IO0AC0, allows 11 consecutive time-outs before the station, CU1, becomes inactive.

CP F CU1 RETRY 0

This command sets the retry maximum to 0. The station, CU1, does not become inactive due to a time-out.

CP F SEL—Change Selection Address

This command changes the selection address of a 4978, 4980, or printer device station or of a terminal station controlled by a BSC line. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **SEL** *address*

station-name
is the station name, 1 to 8 alphameric characters.

address
is the 4-digit hexadecimal station selection address. Appendix A, "3270 Polling and Selection Address Table" on page 307 contains the values to be specified for terminal stations. For device stations, specify any hexadecimal value.

CP F SEL Example

```
CP F S1CRT1 SEL 60C6
```

CP F SEQ—Reset Message Sequence Numbers and Character Counts

This command resets input and output message sequence numbers and character counts. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

CP F	*station-name* **SEQ** **{I | O | B}**

station-name
is the station name, 1 to 8 alphameric characters.

{I | O | B}
I resets input sequence number and character count to 0.

O resets output sequence number and character count to 0.

B resets both sequence numbers and character counts to 0.

CP F SEQ Example

```
CP F T4978 SEQ B
```

CP F TIMEOUT

- See PTF in Msgs + Codes

CP F USER—Change SNA Logon or X.25 Call Request User Data Field

This command changes the SNA logon user data field of an SNA logical unit station or the call request user data field of an SVC CIRC station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified.

This command has no effect on SNA LU stations if the SNA logon ID is 0 or HOST; it is invalid for SVC CIRC stations whose contact type is USERINIT or WAIT. The command has no effect if the station has no SNA logon ID.

CP F	*station-name* **USER** [*user-field*]

station-name
is the station name, 1 to 8 alphameric characters.

user-field
is a 1- to 16-character logon user data field for a SNA LU or 1- to 12-character call request user data field for a circuit station. To delete an existing data field, omit the parameter.

CP F USER Examples

CP F X25CIRC USER

CP F SNALU3 USER TSOID/PASSWORD

CP F WINDOW—Change Line or Circuit Station's Window

This command changes the window size of a circuit, or DTE line station. This command modifies a station definition in $.SYSNET, the network configuration data set. If the station is active when the command is issued, its station control block is also modified. However, the new window is not in effect until the circuit is reopened.

CP F	*station-name* **WINDOW** *window*

station-name
: is the circuit or line station name, 1 to 8 alphameric characters.

window
: is the number of packets that can be sent before an acknowledgment is required. Specify a decimal number from 1 to 127. To delete an existing window size, omit the parameter.

CP F WINDOW Examples

    CP F LINE1 WINDOW 4

This command sets LINE1's window size to 4. 4 is now the window size for any of LINE1's circuit stations that do not have window sizes in their station definitions.

    CP F CIRC1

This command removes the window size from CIRC1's station definition. It will now use it's controlling line station's window size.

CP FILE NAME—Assign or Change Disk-Queue Data Set

This command assigns or changes a station's disk-queue data set. The station cannot be active. If the station already has a disk-queue data set that contains undelivered messages, those messages remain in the data set. You can receive them later by reassigning the data set to a station or by using the Communications Facility utility program $.UT2. You can also use it to redefine the station as storage-queued.

CP FILE	**NAME** *station-name* *dsname* [,*vol-id*] [{**AC** | **IN**}]

station-name
is the station name, 1 to 8 alphameric characters. This cannot be an active station.

dsname
is the name of the data set to be used as the station's disk queue. To specify that the station is to have no disk queue, enter 0000.

vol-id
is the volume name of the disk-queue data set. A null entry defaults to the volume where the control program resides.

AC | IN
specifies whether disk queuing is to be active (AC) or inactive (IN) when the station is started.

Special Considerations

You cannot use this command to change an inactive station whose disk-queue data set is in use. (This condition might result from an abnormal termination.) You can use the CP FILE RESET command to recover from this situation.

CP FILE NAME Example

CP FILE NAME T3101 0000

This command redefines station T3101 as storage-queued.

CP FILE PARMS—Display Disk-Queuing Parameters

This command displays disk-queuing parameters of stations.

CP FILE	**PARMS** {**NET** | *station-name*} [*destination*]

NET
displays disk-queuing parameters of all stations with disk queues.

station-name
displays disk-queuing parameters for this station if it has a disk queue.

destination
is the name of the EDX device on which the display is to appear. The default is the terminal at which you enter the command.

CP FILE PARMS Example

CP FILE PARMS NET $SYSPRTR

This command displays the disk-queuing parameters of all stations with disk queues on device $SYSPRTR:

```
*09:40:19  DISK-QUEUING PARAMETERS
STATION   DATA SET NAME STATUS SIZE-REC AC OL WARN DBND #DISKMSG %FULL HIGH%
S1        S1,CFVOL      CLOSED
HOSTCRT1  HCRT1,CFVOL   CLOSED
T3101     T3101,EDX002   OPEN     500    Y  Y  80   10       0       0    25
```

where:

STATION
is the name of a station defined to use disk queuing.

DATA SET NAME
is the name and volume of the station's disk-queue data set.

STATUS
OPEN indicates that the data set is currently in use by the station.

CLOSED indicates that the data set is not in use.

SIZE-REC
is the size of the data set in 256-byte records.

AC
(Y/N) indicates whether or not disk queuing is to be active when the station is started.

OL
(Y/N) indicates whether or not the data set is to be used in overlay mode.

WARN
is the specified percentage of capacity at which warning messages are issued.

DBND
is the specified percentage below the WARN percentage (deadband) to which the message capacity must fall, and then rise to the WARN percentage again, before another warning message is issued.

#DISKMSG
is the number of disk-queued messages.

%FULL
is the current percent of the data set capacity occupied by undelivered messages.

HIGH%
is the maximum percent of the data set capacity occupied by undelivered messages since the last time this value was reset by the $.DSINIT utility.

CP FILE RESET—Delete Disk-Queue Data Set

This command redefines a station as storage-queued. It deletes all references to the station's current disk-queue data set, even when that data set is in use. (This condition might result from an abnormal termination.)

CP FILE	**RESET** *station-name*

station-name
is the station name, 1 to 8 alphameric characters. This station must be stopped or halted.

CP FILE RESET Example

CP FILE RESET PROGA

This command redefines station PROGA as storage-queued.

CP H—Halt Station

This command halts a station. If the station uses disk queuing, the halt command closes the data set and updates the message sequence numbers and character counts in the $.SYSNET station definition. The station block is flagged inactive. The program that manages the station is requested to delete the station block from S$POOL. When a station is halted, any pending storage-queued messages are discarded.

CP H	*station-name* [*station-name1...station-namen*]

station-name
is the station to be halted. You may specify more than one name.

You can't use this command to halt:

- terminal stations
- node stations
- vector stations
- SNA logical unit stations
- alias stations
- circuit stations

You have to halt the station that controls them.

Halt Command Actions

Device station: The station block is deleted, its task in the I/O control program is detached, and control of the device is released.

Line station—CPU: Control of the line is released. The vector station blocks that represent stations in the remote node and the line station block are deleted. The station's task in the I/O control program is detached.

Line station—PTPT: Control of the line is released, the line station block is deleted, and its task in the I/O control program is detached.

Line station—3271, 3271E, and channel attach: Control of the line is released. The associated terminal stations are stopped. Their station blocks and the line station block are deleted. The station's task (or tasks) in the I/O control program is detached.

Line station—DTE and DCE lines: Control of the line is released. The associated circuit stations are stopped. The circuit station blocks, the remote node stations linked to the circuit stations, the vector station blocks that represent stations in those nodes, and the line station blocks are deleted. All open circuits are closed. The station's tasks in the I/O control program are detached.

Line station—Local Communications Controller: An off-line notification message is sent to each node with which communication was established. Each associated node station is stopped. The node station blocks, the vector station blocks that represent stations in those nodes, and the line station block are deleted. All open channels are closed. The station's tasks in the I/O control program are detached.

SNA physical unit station: Each associated SNA logical unit station is stopped, its station block is deleted, and its task in the I/O control program is detached. Then the SNA physical unit station block is deleted.

User station—I/O control program: All device, line, or SNA physical unit stations managed by the I/O control program are halted. Then the program deletes its station block and terminates execution. You cannot halt an I/O control program unless it is defined as a user station.

User station—application program: The halt actions for a user station depend on how the program is written. It may discard pending storage-queued messages or process them before halting. It is expected to delete its station block and terminate execution.

User station—program dispatcher: Each message station associated with an entry in the program dispatcher's transaction table is halted. The application program associated with the message station is expected to halt. Then the program dispatcher deletes its station block and terminates execution.

Message station: The halt actions for a message station depend on the program that manages the station. It may discard pending storage-queued messages or process them. It is expected to delete the message station block. You can't halt a message station that is already stopped.

System station—$.CF or $.DISP: The control program ($.CF) terminates execution when it has been requested to halt and there are no station blocks in S$POOL except aliases, the remote disk station ($.PD>IO<), the undeliverable message station ($.WASTE), and its own station ($.DISP). Before terminating, the program deletes these station blocks.

CP H Example

```
CP H $.CF $.IO0AE0 $.IO0630
```

This command halts the control program and two I/O control programs.

CP HELP—Display CP Commands

This command displays the syntax of individual command processor commands or lists all the CP commands.

CP HELP	[{*command* | **DEF** *type* [*subtype*] | **F** *type* [*subtype*]}]

command
is the command processor command to be displayed. A null entry displays a list of all commands and their functions.

DEF *type subtype*
displays all the parameters of the define command for the specified station type and subtype. If you do not specify a subtype, all Define commands for the specified station type are displayed.

F *type subtype*
displays all the Modify commands for the specified station type and subtype. If you do not specify a subtype, all the Modify commands for the specified station type are displayed.

CP HELP Examples

CP HELP

This command lists all the command processor commands.

```
COMMAND     FUNCTION
DEF         DEFINE STATION
F           MODIFY STATION ATTRIBUTES
FILE        DISPLAY OR MODIFY DISK QUEUING PARAMETERS
H           HALT STATIONS
HELP        DISPLAY COMMAND SYNTAX
LINK        LINK STATIONS
MAP3        DISPLAY OR MODIFY MAP APPLICATION SERVER INFORMATION
P           STOP STATIONS
Q           DISPLAY STATIONS AND OTHER SYSTEM INFORMATION
READ        READ A CP COMMAND DATA SET
S           START STATIONS
SET         SET SYSTEM PARAMETERS
ST          DISPLAY STATISTICS
V           SET LCC BYPASS
```

[3] The MAP command is supported by the *Manufacturing Automation Protocol Application Server* (Program Number 5719-XT1).

CP HELP DEF LINE

This command lists parameters and shows the syntax of the define commands that you can issue for a line station.

```
DEF NAME NA LINE   CPU|PTPT|3271C|3271E|CA BUFFSIZE <DSNAME<,VOLUME> Y|N>
DEF NAME NA LINE   LCC|PCC BUFFSIZE DEVICE-ADDR <DSNAME<,VOLUME> Y|N>
DEF NAME NA LINE   DCE|DTE BUFFSIZE PKTSIZ WINDOW <DSNAME<,VOLUME> Y|N>>
```

CP HELP F DEV 4978

This command lists the syntax and parameters of the Modify commands that you can issue for a 4978 device station.

```
*** 4978 DEVICE STATIONS:
F NAME BUFFER     XXXX                       BUFFER SIZE
F NAME CLASS      1|2                        CFBUF USAGE CLASS
F NAME DISKQ      AC|IN
F NAME HOLD       IN|OUT ON|OFF
F NAME MODE       ALPHANUMERIC|NUMERIC
F NAME MODE       BASIC|RECORD
F NAME MODE       NON-DISPLAY|TEXT|APL
F NAME MODE       UL|LU|UU
F NAME NA         XXXX                       NETWORK ADDRESS
F NAME NAME       CCCCCCCC                   STATION NAME
F NAME POLL       XXXX                       POLLING ADDRESS
F NAME SEL        XXXX                       SELECTION ADDRESS
F NAME SEQ        INPUT|OUTPUT|BOTH
```

The characters C, N (not shown), and X indicate that the parameter is variable information, as explained in the right-hand column. The maximum parameter length is indicated by the number of characters. C indicates that the parameter is characters, N indicates that it is a decimal numeric value, and X indicates that it is a hexadecimal value.

Less than and greater than signs (<, >) indicate that the parameter is optional. A vertical bar (|) indicates a choice of keyword parameters; specify one of them.

CP LINK—Define Message Destination

This command defines a default destination or an alternate destination for messages sent by a station. The message dispatcher uses the default destination to route messages for which no destination is specified. The program dispatcher uses the alternate destination to route undeliverable transactions; some I/O control programs send control messages to the alternate destination. User programs may use the alternate destination for different purposes. The specified link is set in the station definition in $.SYSNET. If the station is active, the link is also set in its station block.

CP LINK	*station-name1* *station-name2* [{**BOTH** | **ALT**}]

station-name1
: is the name of the origin station.

station-name2
: is the name of the destination station. To remove an existing link, specify 0000.

BOTH
: links the two stations to each other.

ALT
: specifies an alternate link from *station-name1* to *station-name2*.

CP LINK Examples

```
CP LINK S1CRT1 HOSTCRT1 BOTH
```

This command links stations S1CRT1 and HOSTCRT1 to each other. Both station definitions (and station blocks if they are active) are modified.

```
CP LINK T4978 0000
```

This command unlinks station T4978.

```
CP LINK T3101 $.PD
CP LINK T3101 HOSTCRT2 ALT
```

The first command defines $.PD as the default destination of messages sent by station T3101. The second command defines HOSTCRT2 as the station to receive undeliverable transactions sent to the program dispatcher by station T3101.

CP P—Stop Station

This command stops a station.

CP P	*station-name* [*station-name1...station-namen*]

station-name
is the station to be stopped. You may specify more than one name. You cannot stop a vector station.

Stop Command Actions

If the station uses disk queuing, the stop command closes the data set and updates the message sequence numbers and character counts in the $.SYSNET station definition. The station block is flagged inactive. Messages sent to stopped stations, other than storage-queued messages for MESSAGE stations, are considered undeliverable. Additional stop actions for other station types are as follows:

Circuit station: Any pending storage-queued messages for the station are discarded; its X.25 circuit is closed; and it is removed from the line's queue of circuits.

Device station: Any pending storage-queued messages for the station are discarded, its task in the IOCP is detached, and control of the device is released.

Line station—3271C, 3271E, CPU, and PTPT: Control of the line is released, and the station's task in the IOCP is detached.

Line station—channel attach: The associated terminal (port) stations are stopped, control of the channel is released, and the station's tasks in the IOCP are detached.

Line station—Local Communications Controller: An off-line notification message is sent to each node with which communication was established, the associated node stations are stopped, all open subchannels are closed, and the station's tasks in the IOCP are detached.

Line station—Series/1-PC Connect Attachment: An off-line notification message is sent to the PC node with which communication was established, the associated PC node station is stopped, all open subchannels are closed, and the station's tasks in the IOCP are detached.

Line station—DTE and DCE lines: The control of the line is released. The associated circuit stations are stopped, their open X.25 circuits are closed, and the station's tasks in the IOCP are detached.

Terminal station—3271C and 3271E: Any pending storage-queued messages are discarded.

Terminal station—port: There are no additional stop actions.

Node station: There are no additional stop actions.

SNA physical unit station: Each associated SNA logical unit station is stopped.

SNA logical unit station: The station's task in the IOCP is detached.

User station—I/O control program: All device, line, or SNA physical unit stations managed by the IOCP are halted. Then the program deletes its station block and terminates execution. You cannot stop an IOCP unless it is defined as a user station.

User station—application program: The stop actions for a user station depend on how the program is written. The program may discard pending storage-queued messages or process them before stopping. The Communications Facility expects it to delete its station block and terminate execution.

User station—program dispatcher: The program dispatcher deletes its station block and terminates execution. It does not stop the programs it manages.

Message station: The stop actions for a message station depend on the program that manages the station. It may discard pending storage-queued messages or process them. It is expected to delete the message station block.

System station—$.CF or $.DISP: The control program ($.CF) terminates execution when it has been requested to stop and there are no station blocks in S$POOL except aliases, the remote disk station ($.PD>IO<), the undeliverable message station ($.WASTE), and its own station ($.DISP). Before terminating, the program deletes these station blocks.

Volume station: Any pending storage-queued messages for the station are discarded and the volume task in the IOCP is detached.

CP P Example

```
CP P S1PRT CRTDEV CRTPTR
```

This command stops three stations: S1PRT, CRTDEV, and CRTPTR.

CP Q—Display $.SYSNET Station Definitions

This command displays information about all stations defined in $.SYSNET or all stations of the specified type.

CP Q	{**NET** | *type*} [*destination*]

NET
displays all station definitions.

type
specifies the station type: ALIAS, CIRCUIT, DEVICE, LINE, LU, MSG, NODE, PROGRAM, PU, TERM, or USER. Note that PROGRAM and USER are synonyms here.

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

CP Q Example

CP Q NET

This command displays the kind of information shown in Figure 12 on page 132.

In Figure 12 on page 132,

STATION
is the name of the station.

TYPE
is the type and subtype of the station, as described in Figure 13 on page 134.

NA
is the network address of the station.

LINK
is the direct link vector, the network address of the station to which this one is linked.

```
 08:08:33 QUEUE DISPLAY TYPE: NET
STATION   TYPE  NA   LINK STAT         S T A T I O N  D E P E N D E N T
$.WASTE   0C00 0C11 0000 0000
HOST      0AE0 0C09 0000 0000 BFSZ: 09C4
T4978     0678 0C24 0CC1 0000 BFSZ: 09C4  POLL: 4040  SEL: 6040
T770      04E2 0CC1 0C24 0000 LINE NAME: HOST       POLL: 4040  SEL: 6040
LCC1      0AB0 0C50 0000 0000 BFSZ: 09C4  DEVICE ADDR: 0050
LINK72    0AB8 0172 0000 0000 BFSZ: 0100  PKTSIZ: 128 WINDOW:  2
NODE0E    1040 0E00 0000 0000 LCC NAME: LCC1      RING ADDR: 001C
NODE0F    1041 0F00 0000 0000 LCC NAME: LCC1      RING ADDR: 002C
CHANNEL   0AD0 0C22 0000 0000 BFSZ: 09C4
PORT1     04D0 0CF1 0CFF 0000 LINE NAME: CHANNEL    PORT: 01
PRINTER1  0674 0C01 0000 0000 BFSZ: 09C4  POLL: 40C0  SEL: 60C0
T3101     0631 0CFF 0CF1 0000 BFSZ: 09C4  POLL: 40C1  $SYSPRTR
T3101F    06F3 0C59 0000 0000 BFSZ: 09C4  POLL: 40C2  $SYSPRTR  NOAUTO  2400 BPS
CIRC1     16BE 01D1 01E1 0000 LINE NAME: LINK72  CALLID: 40 USE: STD   CON: WAIT
                                 PROTID: 00000000 USER DATA:
CIRC2     16BE 01D2 01E1 0000 LINE NAME: LINK72  CALLID: 42 USE: STD+  CON: INIT
                                 PROTID: C00000F1 USER DATA:TSO/PASSWORD
PU1       14E8 0CA1 0000 0000 BFSZ: 09C4
LU1       12E2 0CA2 0000 0000 PU NAME: PU1        LOCAL ADDR: 13  LOGON ID: 2
WASTE     0E00 0CB1 0000 0000 ALIAS FOR $.WASTE
$.PD      0200 0CC4 0000 0000
$.IO14E8  0200 0CE8 0000 0000
$.IO0AB0  0200 0CB0 0000 0000
$.IO0AB8  0200 0CB8 0000 0000
$.IO0630  0200 0CB2 0000 0000
$.IO06F0  0200 0CB3 0000 0000
$.IO0AE0  0200 0CE0 0000 0000
$.IO0670  0200 0C70 0000 0000
$.IO0AD0  0200 0CB4 0000 0000
```

Figure 12. Display $.SYSNET Station Definitions

STAT

is a 16-bit value that contains information about the station's status and mode of operation. Note that modes of operation vary according to the station type, as described in the section "CP F MODE—Change Mode of Operation" on page 89. The significant bits in a station definition are:

```
0123 4567 89AB CDEF
X... .... .... ....  0 = Station not started, 1 = Station active
.X.. .... .... ....  0 = Secondary mode, 1 = Primary mode
..00 000. .... ....
.... ...X .... ....  0 = Basic mode, 1 = Record mode, or
                     0 = Non-remove mode, 1 = Remove mode
.... .... 00.. ....
.... .... ..XX ....  These bits have the following meanings:
                     00 = Non-display mode or upper/lower case mode
                     01 = Lower/upper case mode
                     10 = Text mode or upper/upper case mode
                     11 = APL mode
.... .... .... X...  1 = Input from station prohibited
.... .... .... .X..  1 = Output to station held
.... .... .... ..0.
.... .... .... ...X  0 = Alphameric mode, 1 = Numeric mode, or
                     0 = XOFF mode, 1 = XON mode
```

STATION DEPENDENT

shows different information for different types of stations, as follows:

327*x* terminal:
: name of controlling line station, polling address, selection address.

port terminal:
: name of controlling line station, port number.

4978 or printer device:
: buffer size, polling address, selection address.

3101 device:
: buffer size, polling address, name of hardcopy device.

3101F device:
: buffer size, polling address, name of hardcopy device, whether or not modem supports auto-answer, bit rate.

line:
: buffer size; for subtype LCC only, device address (address of subchannel 0); for subtypes DCE and DTE only, packet size and window size.

alias:
: name of station for which this one is alias.

node:
: name of controlling LCC line station, ring address.

SNA logical unit:
: name of associated SNA physical unit station, local address (logical unit number), logon ID.

SNA physical unit:
: buffer size.

circuit:
: name of the controlling DTE or DCE line station and the usage type. For PVC stations, the logical channel identifier; for SVC stations, the contact type, protocol identifier, and user data.

Type	Subtype	Symbols	Description
00	—		System[2]
02	00	USER[3]	Program
16	BD	CIRCUIT/PVC	X.25 permanent virtual circuit
16	BE	CIRCUIT/SVC	X.25 switched virtual circuit
04	C0	TERM/3271	3271 control unit
04	C2	TERM/3277	3277 terminal
04	C4	TERM/3286	3286 printer
04	D0	TERM/PORT	Emulated channel attach port
04	E0	TERM/3271	Emulated 3271 control unit
04	E2	TERM/3277	Emulate 3277 terminal
04	E4	TERM/3286	Emulated 3286 printer
06	31	DEVICE/3101	3101 terminal, teletypewriter adapter, managed as a 3277
06	74	DEVICE/PRINTER	4973/4974/4975/5219/5224/5225 printer, managed as a 3286
06	78	DEVICE/4978	4978/4980 terminal, managed as a 3277
06	80		Work session controller static screen device[2]
06	81		Work session controller roll screen device[2]
06	83		Work session controller output only device[2]
06	F3	DEVICE/3101F	3101 or 7485 terminal, feature programmable communications adapter or multifunction attachment, managed as a 3277
08	—		Vector[2]
0A	10	LINE/CPU	Series/1-to-Series/1 line
0A	BC	LINE/DCE	HDLC line—data circuit-terminating equipment
0A	B8	LINE/DTE	HDLC line—data terminal equipment
0A	20	LINE/PTPT	Point-to-point line
0A	B0	LINE/LCC	Local Communications Controller
0A	C0	LINE/3271C	3270 control line
0A	D0	LINE/CA	Channel attachment
0A	E0	LINE/3271E	3270 emulation line
0C	00	MSG	Message
0E	00	ALIAS	Alias
10	40	NODE/CF	Communications Facility node
10	41	NODE/CM	Communications Manager node
12	E2	LU/3277	SNA 3277 logical unit
12	EA	LU/3278	SNA 3278 logical unit
12	EB	LU/3279	SNA 3279 logical unit

Figure 13 (Part 1 of 2). Station Types

Type	Subtype	Symbols	Description
12	E4	LU/3286	SNA 3286 logical unit
12	EC	LU/3287	SNA 3287 logical unit
14	E8	PU/3274ES	SNA physical unit

Figure 13 (Part 2 of 2). Station Types

[2] For internal use only.

[3] The symbols are entered on the command that defines that station type and subtype. Not all subtypes have corresponding symbols.

CP Q *—Display Active Station Information

This command displays information about all active stations; that is, the station blocks in S$POOL, the system storage pool.

CP Q	[*] [*destination*]

*
: may be omitted if no destination is specified.

destination
: is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

*CP Q * Example*

CP Q * $SYSPRTR

This command displays information about all the stations in S$POOL at the device specified, $SYSPRTR.

```
 08:41:54 QUEUE DISPLAY TYPE: # (IN STORAGE)
STATION  TYPE  NA   LINK STAT LINE FIQ  LIQ  ISN  OSN  DISK MSG ACT
$.DISP   0000 0C00 0000 8000 1FAC 0000 0000 0000 0000   N/A
$.IOOAEO 0200 0CE0 0000 8000 0000 0000 0000 0000 0001   N/A
$.IO0670 0200 0C70 0000 8000 0000 0000 0000 0000 0003   N/A
$.IO14E8 0200 0CE8 0000 8000 0000 0000 0000 0000 0005   N/A
CU71     04E0 0CC0 0000 8000 0C09 0000 0000 0000 0000   N/A
T770     04E2 0CC1 0C24 8000 0C09 0000 0000 0000 0000   N/A
T4978    0678 0C24 0CC1 8000 08BE 0000 0000 0000 0000   N/A
HOST     0AE0 0C09 0000 8002 0000 0000 0000 0000 0000   N/A
$.WASTE  0C00 0C11 0000 8000 0000 3268 32A0 0000 0000   N/A
PU1      14E8 0CA1 0000 8000 0000 0000 0000 0000 0000   N/A
```

where:

STATION
: is the name of the station.

TYPE
: is the type and subtype of the station.

NA
: is the network address of the station.

LINK
: is the direct link vector, the network address of the station to which this one is linked.

STAT

is a 16-bit value that contains information about the station's status and mode of operation. The significant bits in a station block are:

```
0123 4567 89AB CDEF
X... .... .... ....  0 = Stopped, 1 = Active
.XXX XXXX .... ....  Station type dependent *
.... .... X... ....  1 = Waiting for acknowledgement of SEND
.... .... .XXX ....  Station type dependent
.... .... .... X...  1 = Input from station prohibited
.... .... .... .X..  1 = Output to station held
.... .... .... ..XX  Station type dependent *
```

* The station type dependent information is either the station's mode of operation, as described in section "CP Q—Display $.SYSNET Station Definitions" on page 131 or other status information, as described in the *Debugging Guide*.

LINE

shows different information for different types of stations, as follows:

terminal, node, or circuit:
network address of controlling line station.

device:
address of I/O control program station block.

SNA logical unit:
network address of associated SNA physical unit station.

FIQ

is the address, within the system message pool, of the first message on the station's queue.

LIQ

is the address, within the system message pool, of the last message on the station's queue.

ISN

is the input sequence number, which indicates the number of messages sent by the station.

OSN

is the output sequence number, which indicates the number of messages received by the station.

DISK MSG

indicates whether or not (YES or NO) there are any messages on the station's disk queue. N/A indicates either that the station does not have a disk queue or that the control program being used does not support disk queuing.

ACT

indicates, for stations with disk queues, whether or not (YES or NO) disk queuing is active.

CP Q BSC—Display EDX BSC Line Information

This command displays information about the BSC lines defined to the Event Driven Executive.

CP Q	**BSC** [*destination*]

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

CP Q BSC Example

CP Q BSC

This command displays the following kind of information:

```
 08:24:32 QUEUE DISPLAY TYPE: BSC
TYPE   DVAD   DVID   DDBA
SM     0019   1006   1E3E
MC     0029   1006   1E3E
SM     0070   2206   1F4E
SM     0071   2206   1FF0
SM     0072   2206   207C
SM     0073   2206   2108
PT     0039   1006   2194
```

where:

TYPE
is the line type as defined to EDX.

DVAD
is the line address.

DVID
is the hardware device ID of the BSC adapter.

DDBA
is the address of the EDX device data block (BSCDDB).

CP Q NET—Display $.SYSNET Station Definitions

This command displays information about all stations defined in $.SYSNET or all stations of the specified type.

CP Q	{NE[T] | *type*} [*destination*]

NET
displays all station definitions.

type
specifies the station type: ALIAS, CIRCUIT, DEVICE, LINE, LU, MSG, NODE, PROGRAM, PU, TERM, USER, or VOLUME. Note that PROGRAM and USER are synonyms here.

destination
is the name of the EDX device on which the display is to appear or the Communications Facility user or message station to which the data is to be sent. If you do not specify a destination or if you specify an unknown device, the destination defaults to the EDX terminal at which you enter the command.

CP Q NET Example

```
CP Q NET
```

This command displays the kind of information shown in Figure 12 on page 154.

In Figure 12 on page 154,

STATION
is the name of the station.

TYPE
is the type and subtype of the station, as described in Figure 13 on page 157.

NA
is the network address of the station.

LINK
is the direct link vector, the network address of the station to which this one is linked.

```
 08:08:33 QUEUE DISPLAY TYPE: NET
STATION  TYPE  NA  LINK STAT C       S T A T I O N  D E P E N D E N T
-------- ---- ---- ---- ---- - -------------------------------------------
LINECPU  0A10 0151 0000 0000 1 BFSZ: 09C4
LINEPTPT 0A20 0152 0000 0000 1 BFSZ: 09C4
LINEPCC  0AA8 0153 0000 0000 1 BFSZ: 09C4  DEVICE ADDR: 0058
LINELCC  0AB0 0154 0000 0000 1 BFSZ: 09C4  DEVICE ADDR: 0050
LINEDTE  0AB8 0155 0000 0000 1 BFSZ: 09C4 PKTSIZ: 128 WINDOW:   2
LINEDCE  0ABC 0156 0000 0000 1 BFSZ: 09C4 PKTSIZ: 128 WINDOW:   2
LINE327C 0AC0 0157 0000 0000 1 BFSZ: 09C4
LINECHAN 0AD0 0158 0000 0000 1 BFSZ: 09C4
LINE327E 0AE0 0159 0000 0000 1 BFSZ: 09C4
VOLLAN   1CC8 01E1 0000 0000 1
USER     0200 0111 0000 0000 1
T3271C   04C0 0121 0000 0000 1 LINE: LINE327C  POLL: 407F SEL: 0000
T3277C   04C2 0122 0000 0000 1 LINE: LINE327C  POLL: 4040 SEL: 6040
T3286C   04C4 0123 0000 0000 1 LINE: LINE327C  POLL: 40C1 SEL: 60C1
TERMPORT 04D0 0124 0000 0000 1 LINE: LINECHAN PORT:   8
T3271E   04E0 0125 0000 0000 1 LINE: LINE327E  POLL: 407F SEL: 0000
T3277E   04E2 0126 0000 0000 1 LINE: LINE327E  POLL: 4040 SEL: 6040
T3286E   04E4 0127 0000 0000 1 LINE: LINE327E  POLL: 40C1 SEL: 60C1
DVCE4974 0674 0132 0000 0000 1 BFSZ: 09C4  POLL: 4040 SEL: 6040
DVCE4978 0678 0133 0000 0000 1 BFSZ: 09C4  POLL: 4040 SEL: 6040
DVCE310F 06F3 0134 0000 0000 1 BFSZ: 09C4  POLL: 4040 $SYSPRTR  9600 BPS NOAUTO
VECTOR   1040 0200 0000 0000 1 LINE: ........ RING ADDR: 0000
MSG      0C00 0161 0000 0000 1
ALIAS    0E00 0171 0000 0000 1 ALIAS FOR MSG
NODECF   1040 0300 0000 0000 1 LINE: LINELCC  RING ADDR: 001F
NODECM   1041 0400 0000 0000 1 LINE: LINELCC  RING ADDR: 002F
NODEPC   1042 C100 0000 0000 1 LINE: LINEPCC  VOLUME NAME: VOLLAN
PU3274ES 14E8 01A1 0000 0000 1 BFSZ: 09C4     PU NUMBER: 1 [3]
LU3277E  12E2 0191 0000 0000 1 PU NAME: PU3274ES LOCAL ADDR:  0  LOGON ID: 0
LU3286E  12E4 0192 0000 0000 1 PU NAME: PU3274ES LOCAL ADDR:  0  LOGON ID: 0
LU3278E  12EA 0193 0000 0000 1 PU NAME: PU3274ES LOCAL ADDR:  0  LOGON ID: 0
LU3279E  12EB 0194 0000 0000 1 PU NAME: PU3274ES LOCAL ADDR:  0  LOGON ID: 0
LU3287E  12EC 0195 0000 0000 1 PU NAME: PU3274ES LOCAL ADDR:  0  LOGON ID: 0
CIRCPVC  16BD 01B1 0000 0000 1 LINE: LINEDTE  LCI:     6 USE: CF
CIRCSVC  16BE 01B2 0000 0000 1 LINE: LINEDTE  CALLID:  0 USE: CF   CON: WAIT
                                 PROTID: 00000000 USER DATA:
```

Figure 12. Display $.SYSNET Station Definitions

[3] This field is present only if you've installed the EDX Secondary SNA2 support.

STAT

is a 16-bit value that contains information about the station's status and mode of operation. Note that modes of operation vary according to the station type, as described in the section "CP F MODE—Change Mode of Operation" on page 99. The significant bits in a station definition are:

```
0123 4567 89AB CDEF
X... .... .... ....  0 = Station not started, 1 = Station active
.X.. .... .... ....  0 = Secondary mode, 1 = Primary mode
..00 000. .... ....
.... ...X .... ....  0 = Basic mode, 1 = Record mode, or
                     0 = Non-remove mode, 1 = Remove mode
.... .... 00.. ....
.... .... ..XX ....  These bits have the following meanings:
                     00 = Non-display mode or upper/lower case mode
                     01 = Lower/upper case mode
                     10 = Text mode or upper/upper case mode
                     11 = APL mode
.... .... .... X...  1 = Input from station prohibited
.... .... .... .X..  1 = Output to station held
.... .... .... ..0.
.... .... .... ...X  0 = Alphanumeric mode, 1 = Numeric mode, or
                     0 = XOFF mode, 1 = XON mode
```

C

is the CFBUF usage class of the station.

STATION DEPENDENT

shows different information for different types of stations, as follows:

327*x* terminal:
name of controlling line station, polling address, selection address.

port terminal:
name of controlling line station, port number.

4978 or printer device:
buffer size, polling address, selection address.

3101 device:
buffer size, polling address, name of hardcopy device.

3101F device:
buffer size, polling address, name of hardcopy device, whether or not modem supports auto-answer, bit rate.

line:
buffer size; for subtype LCC and PCC only, device address (address of subchannel 0); for subtypes DCE and DTE only, packet size and window size.

alias:
name of station for which this one is alias.

node:
subtype CF or CM—name of controlling LCC line station and ring address;

subtype PC—name of controlling PCC line station and the volume station name.

SNA logical unit:
name of associated SNA physical unit station, local address (logical unit number), logon ID.

SNA physical unit:
buffer size, EDX-SNA PU number associated with this station.[4]

circuit:
name of the controlling DTE or DCE line station and the usage type. For PVC stations, the logical channel identifier; for SVC stations, the contact type, protocol identifier, and user data.

[4] This field is present only if you've installed the EDX Secondary SNA2 support.

CP Q PARM—Display System Parameters of Active Station

This command displays information about started stations for diagnosing system problems.

CP Q	PA[RM] [*destination*]

destination
is the name of the EDX device on which the display is to appear or the Communications Facility user or message station to which the data is to be sent. If you do not specify a destination or if you specify an unknown device, the destination defaults to the EDX terminal at which you enter the command.

CP Q PARM Example

CP Q PARM $SYSLOGA

This command displays the following kind of information:

```
 00:27:22 QUEUE DISPLAY TYPE: PARM (IN STORAGE)
STATION  ADDR TCBA BFSZ WORK ADR1 ADR2 ECB  FLAG DVD  QALV QACT
-------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
$.DISP   0576 .... .... .... .... 36A0 0000 .... .... .... ..
$.IO0AA8 06E6 .... .... .... .... .... 0000 .... .... .... ..
$.IO1CC8 07A6 .... .... .... .... .... 0000 8000 .... .... 00
$.IO0AB0 095E .... .... .... .... .... 0000 .... .... .... ..
$.IO0674 0C06 .... .... .... .... .... 0000 .... .... .... ..
$.IO06F0 0CFE .... .... .... .... .... 0000 .... .... .... ..
$.IO14E8 0DE6 .... .... .... .... .... 0000 .... .... .... 00
PRINTER  0AFE 0B54 09C4 .... 4040 6040 0000 .... 8000 .... ..
T3101A   0C5E C110 09C4 3236 0CE6 0000 0000 .... 0721 .... 0A
VECTOR1  0A0E .... .... 4040 0A06 0050 0808 0856 .... 4040 ..
VECTOR2  0ADE .... .... E3C5 0AD6 0100 0808 0DE6 0B54 D940 ..
PCCLINE  05DE 0634 2400 .... 0092 0054 0000 .... .... .... ..
LCCLINEI 0856 08AC 09C4 .... 004F 0050 0000 .... .... .... ..
MSG1     0A2E .... .... .... .... .... FFFF .... .... .... ..
MSG2     0A86 .... .... .... .... .... FFFF .... .... .... ..
PCNODEA  07FE .... .... .... 0092 .... FFFF .... 0211 .... ..
CFNODEK  09B6 .... .... .... 0049 .... FFFF .... .... .... ..
LU3277E  0E3E 0E94 09C4 0F24 .... 0000 0000 .... .... .... ..
PU3274ES 0D66 .... 09C4 .... .... .... FFFF .... .... .... ..
PCVOLA   073E 8610 .... .... 7B18 7F80 0000 C000 8000 .... 0A
```

If you've installed the EDX Secondary SNA2 support, the information for PU3274ES and LU3277E would look like this:

```
LU3277E  0E3E 6078 09C4 6DFA 0100 0000 0000 .... .... .... 0A
PU3274ES 0D66 .... 09C4 .... 0100 .... FFFF .... .... .... ..
```

where:

STATION
is the name of the station.

ADDR
is the address of the station block.

TCBA
is the address of the station's task control block.

BFSZ
is the buffer size defined for the station.

WORK
is the maximum transaction size if the station is the program dispatcher ($.PD). For all other stations, WORK is the address of the work area associated with the station block. If a CP command created the station block, nothing is displayed in the WORK field.

ADR1
shows different information for different types of stations, as follows:

327*x* terminal or device:
polling address.

port terminal:
port number.

Local Communications Controller line:
ring address.

Series/1-PC Connect Attachment line:
ring address.

node:
ring address.

SNA logical unit:
local address (logical unit number). If you have installed the EDX Secondary SNA2 support, the first byte is the EDX-SNA physical unit number; the second byte is the local address (logical unit number).

SNA physical unit:
If you have installed the EDX Secondary SNA2 support, the EDX-SNA physical unit number appears here.

DTE or DCE line:
packet size in hexadecimal.

Circuit:
packet size in hexadecimal.

Volume station:
return address of internal subroutine calls in volume task.

ADR2
shows different information for different types of stations, as follows:

327*x* terminal or device:
selection address.

Local Communications Controller line:
device address (address of subchannel 0).

Series/1-PC Connect Attachment line:
device address (address of subchannel 0).

SNA logical unit:
logon ID.

DTE or DCE line:
window size in hexadecimal.

Circuit:
window size in hexadecimal.

Volume station:
return address of internal subroutine calls in volume task.

ECB
is the event control block. Zero means the station is waiting for an event. Nonzero means the ECB has been posted, and the station is not waiting.

FLAG
is device-dependent information described in the *Debugging Guide*.

DVD
is device-dependent information described in the *Debugging Guide*.

QALV
is the alternate link vector, the network address of the station to which invalid transactions sent by this station are routed.

QACT
is an 8-bit value that contains information about the station's disk queue status, as follows:

```
0123 4567
X... ....   1 = Disk queuing active
.X.. ....   1 = Disk message retrieval in progress
..X. ....   1 = Start as disk-queued station failed
...X ....   1 = Shut down in progress
.... 1...   Always 1
.... .0..   Reserved
.... ..X.   1 = TCB is outside station control block
.... ...X   1 = Disk messages pending
```

...
indicates that the value is 0 or that the information is not applicable to this type of station.

CP Q SNA—Display EDX-SNA PU and LU Status

This command is not supported for EDX Secondary SNA2. To display the EDX Secondary SNA2 control block information, use the EDX-SNA utility $SNADISP.

This command displays EDX-SNA control block status information for a PU and the LUs defined to it, including LUs that are not used by the Communications Facility. Refer to the appropriate control block description in the EDX *Systems Network Architecture Internal Design* book for interpretations of the data displayed in the numeric fields.

CP Q	**SN[A]** [*destination*]

destination
is the name of the EDX device on which the display is to appear or the Communications Facility user or message station to which the data is to be sent. If you do not specify a destination or if you specify an unknown device, the destination defaults to the EDX terminal at which you enter the command.

CP Q SNA Example

CP Q SNA

This command displays the following kind of information:

```
 18:48:47 QUEUE DISPLAY TYPE: SNA
 PU   ADDR RTRY FLG1
PUOA 258A 0003 6010

 LU   ADDR EOPR FLG1 FLG2 FLG3      FLAG7          TERM    LU STATE
 LU01 25CE 0000 0000 0000 2000 0000 0000 0000 00 C000 *** UNBOUND ***
 LU02 2C5C 492C 2004 0100 A000 0100 1620 004A 50 0000 CONT.BETB
 LU03 32EA                                            *** UNBOUND ***
```

where:

PU
is the device address of the SDLC attachment card supporting the physical unit, prefixed by *PU.*

ADDR (PU description)
is the address in the $SNA partition where the PU's control block is located.

RTRY
is the field DDBRTLM in the EDX-SNA device descriptor block (DDB). This field corresponds to the RETRY parameter defined for the PU.

FLG1 (PU description)
is the field PUSFLG1 in the EDX-SNA physical unit services control block (PUS).

LU
is the LU number, prefixed by *LU.*

ADDR (LU description)
is the LU's control block address in the $SNA partition.

EOPR
is the field LUEEOPR in the LUE control block, which indicates the SNA operation in progress.

FLG1
is the field LUEFLG1 in the LUE control block.

FLG2
is the field LUEFLG2 in the LUE control block.

FLG3
is the field LUSFLG1 in the LUS control block.

FLAG7
shows the finite state machine (FSM) states defined by the fields LULFLG1 through LULFLG7 in the LUL control block.

TERM
is the field LULTERM in the LUL control block.

LU STATE
is a detailed description of the state of the LU. If the LU is not in session, the field may contain one of the following descriptions:

*** UNBOUND ***	LU is not in session
*** BIND PEND.RESET ***	LU session being terminated
*** BIND PEND.ACTIVE ***	LU session being established

If the LU is in session, the state of the half-duplex flip-flop FSM (who has the right to send) is shown by the first descriptor of two descriptors separated by a period. The second descriptor shows the LU's bracket state.

In the example, LU02 shows an LU STATE value of CONT.BETB, which indicates that the session is in half-duplex contention and between brackets. The following states may be displayed for half-duplex sessions:

SEND	Series/1 has right to send
RCV	Host has right to send
CONT	Either may send
ERP1	Series/1 error recovery
PEND1	Awaiting MSG or RSP
PEND2	Awaiting MSG or RSP
PEND.R	Awaiting MSG or RSP

The following are possible bracket states:

BETB	Between brackets
PEND.INB	In brackets after RSP
PEND.BB	Between brackets after RSP
PEND.BTB.RSP.S	Between brackets after RSP to SEND
PEND.BTB.RSP.R	Between brackets after RSP to RCV
PEND.BTB.EC.R	Between brackets after end chain
PEND.BTB.PURGE.R	Between brackets after purge RCV
PEND.BTB.EC.R	Between brackets after end chain
INB	In brackets
PEND.BTB.EC.S	Between brackets after end chain
PEND.BTB.PRG.S	Between brackets after purge sent
PEND.BTB.S	Between brackets after RCV
PEND.BTB.S	Between brackets after SEND

CP Q SYS—Display EDX Terminal Information

This command displays information about the terminals defined to the Event Driven Executive.

CP Q	**SYS** [*destination*]

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

CP Q SYS Example

CP Q SYS

This command displays the following kind of information:

```
 08:07:51 QUEUE DISPLAY TYPE: SYS
TERMNAME  DVAD DVID CCBA
 $SYSLOG  0004 0406 1944
*$TERM1   0005 040E 1AF4
 $TERM2   0006 040E 1CA4
 $SYSLOGA 0000 0010 1E3E
 $SYSPRTR 0001 0306 1FF8
```

where:

TERMNAME
is the label of the TERMINAL statement in the Event Driven Executive configuration definitions (* denotes the name of the terminal that originated the query).

DVAD
is the hardware device address to which the terminal is assigned.

DVID
is the hardware device ID of the adapter to which the terminal is connected.

CCBA
is the address of the terminal's CCB (console control block).

CP Q X25—Display Status Information for Active XHCS Lines

This command displays status information for EDX XHCS lines and all the logical channels defined for each line. If a circuit station is using a logical channel, its name is also displayed.

The command processor gets this information from $XHCS control blocks, line device descriptor modules (DDMs), and the X.25 IOCP work area. Information is displayed only for lines whose DDMs are loaded in storage.

CP Q	**X25** [*destination*]

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

CP Q X25 Example

CP Q X25

This command displays the following kind of information:

```
 00:34:06 QUEUE DISPLAY TYPE: X25
LNAME     TDEV TDID LTYP L N K S T A T  LCI  LCTYP STATION  L C S T A T E
LINK1A    001B 101E DTE  LINK DOWN      001  PVC   ........ DATA TRANSFER
                                        002  PVC   ........ DATA TRANSFER
                                        003  PVC   ........ DATA TRANSFER
                                        004  PVC   ........ DATA TRANSFER
                                        005  PVC   ........ READY
                                        006  PVC   ........ READY
                                        007  PVC   ........ READY
                                        008  PVC   ........ READY
                                        009  PVC   ........ READY
                                        00A  PVC   ........ READY
                                        00B  PVC   ........ READY
                                        00C  PVC   ........ READY
LINK72    0072 5042 DTE  LINK ACTIVE    001  PVC   ........ DATA TRANSFER
                                        002  PVC   ........ DATA TRANSFER
                                        003  PVC   ........ DATA TRANSFER
                                        004  PVC   ........ DATA TRANSFER
                                        005  SVC   LC1      DATA TRANSFER
                                        006  SVC   ........ READY
                                        007  SVC   ........ READY
                                        008  SVC   ........ READY
                                        009  SVC   ........ READY
                                        00A  SVC   ........ READY
                                        00B  SVC   ........ READY
                                        00C  SVC   ........ READY
LINK70    0070 5042 DCE  LINK ACTIVE    001  PVC   ........ DATA TRANSFER
                                        002  PVC   ........ DATA TRANSFER
                                        003  PVC   ........ DATA TRANSFER
                                        004  PVC   ........ DATA TRANSFER
                                        005  SVC   C1       DATA TRANSFER
                                        006  SVC   ........ READY
                                        007  SVC   ........ READY
                                        008  SVC   ........ READY
                                        009  SVC   ........ READY
                                        00A  SVC   ........ READY
                                        00B  SVC   ........ READY
                                        00C  SVC   ........ READY
```

where:

LNAME
: is the device name (DDMNAME) of the adapter used for receiving, as specified using the XHCS utility $XHCUT1.

TDEV
: is the device address of the adapter used for transmitting.

TDID
: is the hardware device ID of the transmit adapter.

LTYP
: is the line type as defined to XHCS. This may be either DTE or DCE.

LNKSTAT

is the state of the link according to the DDM. The following states may be displayed (refer to CCITT Recommendation X.25 for further information):

LINK ACTIVATION
: Establishing the link. The X.25 IOCP has issued an INIT command. For DTE, SAMB was sent, waiting for UA. For DCE, DM was sent, waiting for SABM.

LINK ACTIVE
: Link established. For DTE, UA was received. For DCE, SABM was received and UA was sent.

DTE FRAME REJECT
: (DTE only.) FRMR sent, expecting DM from DCE.

DTE RESETTING
: (DTE only.) Received DM, N2 exceeded, or FRMR.

DCE FRAME REJECT
: (DCE only.) FRMR sent.

DCE RESETTING
: (DCE only.) Received FRMR or N2 exceeded.

DISCONNECTING
: DISC sent.

LINK DOWN
: Received DISC, or DM(F) in response to SABM, or UA in response to DISC.

LCI

is the logical channel number.

LCTYP

is the type of the logical channel. The different types of logical channels are PVC, SVC, SVCI (incoming calls only), and SVCO (outgoing calls only).

STATION

is the name of the circuit station using the logical channel. If no name is displayed, the logical channel is free.

LCSTATE

is the state of the logical channel according to the DDM. The following states may be displayed (corresponding CCITT Recommendation X.25 states are shown in parentheses):

READY
: The logical channel is free, no call exists (p1).

DTE WAITING
: The calling DTE has sent a call request packet across the DTE/DCE interface (p2).

DCE WAITING
: DCE has indicated that there is an incoming call by sending an incoming call packet across the DTE/DCE interface (p3).

DATA TRANSFER
: The calling DTE has received a call connected packet specifying the same logical channel as on the call request, indicating that the call has been accepted by the called DTE (p4). This state is also the initial state for logical channels defined as PVCs.

CALL COLLISION
: A DTE and DCE have simultaneously sent a call request and an incoming call packet specifying the same logical channel (p5).

DTE CLR REQ
: DTE has sent a clear request packet across the DTE/DCE interface (p6).

DCE CLR IND
: The DCE has sent a clear indication packet across the DTE/DCE interface (p7).

DTE RESET REQ
: DTE has sent a reset request packet specifying the logical channel (d2).

DCE RESET IND
: DCE has sent a reset indication packet specifying the logical channel and the reason for resetting (d3).

CP READ—Read Command Data Set

This command reads a data set containing command processor and program dispatcher commands and processes them.

CP READ	*dsname* [,*vol-id*]

dsname
is the name of the data set to be read.

vol-id
is the volume name of the data set. A null entry defaults to the volume where the message dispatcher resides.

CP READ Example

```
CP READ DSNAME
```

This command reads data set DSNAME and processes the commands it contains.

CP S—Start Station

This command starts a station. The start command creates a station block in S$POOL, the system storage pool, using information from the station definition in $.SYSNET.

CP S	*station-name*[,[*vol-id*],[*partition*] [,**COLD**]] [*station-name1*...*station-namen*]

station-name
is the station to be started. You may specify more than one name.

vol-id
is the volume where the program resides. The default is the volume where the control program resides. This parameter is ignored unless the station type is USER.

partition
is the partition where the program is to be loaded. Specify S if the program is to be loaded in a static partition. The partition must be mapped. The default is any available mapped partition. This parameter is ignored unless the station type is USER.

COLD
discards messages in the station's disk-queue data set and allows the station to receive new messages.

If the station uses disk queuing, the start command opens the station's disk-queue data set, and retrieves the message sequence numbers and character counts from the $.SYSNET station definition. If you specified a cold start, messages in the data set are discarded. If the station uses disk queuing and errors occur while opening the data set, the station is started using storage queuing only.

Additional start actions or requirements for particular station types are as follows:

Circuit station: You must start the controlling line station before you start a circuit station.

Device station: The appropriate IOCP, according to the station type and subtype, is loaded if it is not already active. The program activates a task for the station and initiates communication with the device.

Line station—3271C, 3271E, CPU, PTPT, and channel attach: The appropriate IOCP, according to the station type and subtype, is loaded if it is not already active. The program activates a task for the station and acquires control of the line.

Line station—DxE: The X.25 IOCP and EDX XHCS are loaded if they are not already active. The IOCP activates a task for the station and acquires control of the line.

Line station—Local Communications Controller: The Local Communications Controller IOCP is loaded if it is not already active. The program activates several tasks for the station and sends an on-line broadcast message that goes to all other active nodes on the ring.

Line station—Series/1-PC Connect Attachment: The PC Connect IOCP is loaded if it is not already active. The program activates several tasks for the station and sends an on-line broadcast message that goes to the gateway PC on the same Series/1-PC Connect Attachment.

Terminal station: You must start the controlling line station before you start a terminal station.

Node station: Communication is established between the local node and the remote node represented by the station. You must start the associated line station before you start a node station.

SNA physical unit station: The SNA IOCP and EDX-SNA ($SNA) are loaded if they are not already active.

SNA logical unit station: The SNA IOCP activates a task for the SNA logical unit. It establishes communication with a host SNA application. You must start the associated SNA physical unit station before you start a logical unit station.

User station: The program that has the same name as the station is loaded. You need not start an IOCP unless you want to specify its volume name or the partition into which it is to be loaded.

Message station: A message station is usually started by a program—a user application program or the program dispatcher. A message station cannot be removed from the active network unless there is a program managing the station.

Remote station: A remote station is a station of any type that is not in this node, except a node type station on the Local Communications Controller ring or Series/1-PC Connect Attachment. For a remote station, the start command creates a vector station block.

Volume station: The PC Connect disk-server IOCP is loaded if it is not already started. The IOCP activates a task for the station. You must start the volume station before you start the associated PC node station.

CP S Examples

```
CP S HOST,,,COLD CU71 T770 T4978
```

This command starts four stations: HOST, CU71, T770 and T4978. Any disk-queued messages for station HOST are discarded. Disk-queued messages for other stations are retained and can be received.

```
CP S $.CFMENU,CFLIB,6
```

This command starts a station. If it is a user station, program $.CFMENU on volume CFLIB will be loaded into partition 6, if it is a mapped partition and if there is sufficient space. If partition 6 is not a mapped partition, or if there is insufficient space, the program will be loaded into any available mapped partition.

CP SET LOG—Assign System Log Device

This command assigns the specified device or station to receive all Communications Facility system log messages. The default log device is the EDX terminal from which the control program is loaded.

CP SET	**LOG** {*destination* | ***** | **0000**}

destination
is the name of the device or station to receive the system log messages. You may specify one of the following:

- The name of an EDX terminal; it must not be a virtual terminal.
- *, to indicate the terminal from which you are entering CP SET LOG.
- The name of a station.
- 0000, to indicate that system messages are not to be logged.

CP SET LOG Example

```
CP SET LOG $SYSPRTR
```

This command assigns the printer, $SYSPRTR, to receive the system log messages.

CP SET NODE—Set Node Address

This command sets the node address of the Series/1 in which the Communications Facility is running. The default node address is 0000, which implies a single-node system.

CP SET	**NODE** *address*

address
is the 4-digit hexadecimal node address to be assigned to a Series/1. The first two digits can be from 01 to FF, and the last two digits must be 00.

If this node communicates with the Communications Manager over the Local Communications Controller, the first two digits must be the hexadecimal representation of the alphabetic characters the Communications Manager uses to address this node. For example, if this node is known to the Communications Manager as BB, the node address must be C200.

CP SET NODE Example

CP SET NODE 0200

This command assigns a node address of 0200 to a Series/1.

CP ST—Display Message Activity Statistics

This command displays message activity statistics of stations. Message activity statistics are maintained in station blocks while stations are active. When a station that has a disk queue is stopped or halted, its statistics are copied into its $.SYSNET definition. You can use this command to obtain station block statistics and $.SYSNET statistics. For a station with no disk queue, the $.SYSNET statistics always show zero message activity.

CP ST	{*station-name* | * | **NET**} [*destination*]

station-name
: is the name of the station whose statistics are to be displayed. Its $.SYSNET statistics are displayed. If the station is active its station block statistics, preceded by an asterisk, are also displayed.

*
: requests a display of station block statistics of all active stations.

NET
: requests a display of statistics of all defined stations. For each station, its $.SYSNET statistics are displayed. If the station is active its station block statistics, preceded by an asterisk, are also displayed.

destination
: is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

CP ST Example

CP ST T4987

This command displays message activity statistics of station T4978, as shown below. The first line is its station block statistics, the second line is its $.SYSNET statistics.

```
09:05:34 STATION STATISTICS DISPLAY FOR T4978
  STA-NAME    ICNT      ICHR      OCNT      OCHR    %   #DISKMSG  HI%
* T4978        10        286        10      2374    0         0     5
  T4978        10        286         9      2353
```

where:

STA-NAME
: is the name of the station.

ICNT
: is the input message sequence number, the number of messages sent by the station.

ICHR
: is the input message character count, the number of characters sent by the station.

OCNT
is the output message sequence number, the number of messages received by the station.

OCHR
is the output message character count, the number of characters received by the station.

%
is the percentage of the station's disk-queue data set capacity currently in use.

#DISKMSG
is the number of messages in the station's disk-queue data set.

HI%
is the maximum percentage of the station's disk-queue data set capacity ever used.

CP ST H—Display Local Communications Controller Statistics of Local Devices

This command displays the hardware error statistics of a Local Communications Controller local device (defined as type LINE, subtype LCC). The line station must be active.

CP ST	H *ring-address*

ring-address
is a 2-digit hexadecimal Local Communications Controller ring address.

CP ST H Example

CP ST H 4F

This command displays hardware error statistics of the Local Communications Controller local device at ring address 4F:

```
08:07:20 STATION STATISTICS DISPLAY FOR RALNODED-HARDWARE ERROR COUNTS
 STA-NAME RINGAD     TP      RCRC     FCRC      PT       OA      DIR     EC
 RALNODED 4F         0        0        0        0        0      C000    EC02
```

where:

STA-NAME
is the name of the station.

RINGAD
is the ring address of the station.

TP
is the number of transmit random access memory (RAM) parity errors.

RCRC
is the number of cyclic redundancy check character (CRC) errors in frames received for the station.

FCRC
is the first detection CRC errors in frames not belonging to the station.

PT
is the number of pass-through errors.

OA
is the number of origin address parity errors.

DIR
is the directory bytes.

EC
is the hardware engineering change level.

CP ST H—Display Local Communications Controller Statistics of Local Devices and Remote Nodes

This command displays the hardware error statistics of Local Communications Controller local devices (defined as type LINE, subtype LCC) and remote nodes (defined as type NODE). The line station for which statistics are requested or the line station that provides access to the node for which statistics are requested must be active.

CP ST	{*station-name* | * | **NET**} [*destination*] **H** [*ring-address*]

station-name
is the name of the station whose statistics are to be displayed. If *ring-address* is specified, it must be the address of this station.

*
requests a display of statistics of all active Local Communications Controller line and node stations. If *ring-address* is specified, statistics are displayed only for stations with that address.

NET
requests a display of statistics of all active Local Communications Controller line stations and all node stations defined in $.SYSNET. If *ring-address* is specified, statistics are displayed only for stations with that address.

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

ring-address
is a 2-digit hexadecimal Local Communications Controller ring address.

CP ST H Examples

```
CP ST LCCDEV H
```

This command displays statistics for the Local Communications Controller line or node whose station name is LCCDEV.

```
CP ST * $SYSPRTR H
```

This command displays statistics on the system printer for all active Local Communications Controller line and node stations.

```
CP ST * H 10
```

This command displays statistics for active Local Communications Controller line and node stations whose ring address is 10.

CP V OFF—Remove Node from Local Communications Controller Device

This command removes the local node from its Local Communications Controller attachment which removes it from the ring. It also halts the station.

The CP V OFF command is intended to be used only for diagnostic purposes. After a node has been removed from the Local Communications Controller ring, it will no longer respond to any communication on the ring, including a remote IPL request. To clear this condition, the node must be manually IPLed, or the unit containing the Local Communications Controller attachment must be powered off and then powered on.

CP V	**OFF** *station-name*

station-name
is the name of the Local Communications Controller local device station.

CP V OFF Example

CP V OFF LCCDEV

CPRSTART—Restart the Communications Facility

This command restarts the Communications Facility when shutdown is pending because there are active stations. The command can only be entered from a terminal assigned to the partition in which the shutdown program, $.CFSHUT, is running.

CPRSTART	

GOTEST—Execute Transaction Program in Test Mode

This command causes transaction-processing programs in test mode to begin execution under control of $DEBUG. (You must load $DEBUG before issuing this command.) The use of GOTEST for testing a transaction processing program is described in the *Programmer's Guide*. The command can be entered only from a terminal.

GOTEST	

Special Considerations

You can run several programs under control of $DEBUG at the same time, but you must load and activate the programs one at a time. See the *Programmer's Guide* for an example.

PD C—Change Cell Identifier

This command changes the cell identifier.

PD C	*cell-id*

cell-id
is the new cell identifier, 2 alphameric characters. A cell identifier of ?? or ** is invalid.

PD C Example

PD C S3

This command changes the cell identifier to S3.

PD CP—Send CP Command

This command sends a command processor command to a cell. You can send CP commands to any cell in the network with this command.

PD CP	*cell-id* *command*

cell-id
is the cell where the command is to be sent, 2 alphameric characters.

command
is the command processor command to be sent.

PD CP Example

PD CP S2 S TU692

This command sends a CP command to cell S2 to start station TU692.

PD F P—Modify Path Definition Entry

This command modifies an entry in the path definition table. It does not modify $.SYSPD, the program dispatcher data set. See the insert command (PD I P) for additional information on the items that can be modified.

PD F	**P[ATH]** *cell-id* **{H {ON | OFF} |** **C {ON | OFF} |** **S {ON | OFF} |** **PREFE {ON | OFF} |** **PREFI** *pppp*}

cell-id
is a 2-character cell identifier.

H ON | OFF
specifies whether the hold mode for the path is to be set ON or OFF.

C ON | OFF
specifies whether the convert mode for the path is to be set ON or OFF.

S ON | OFF
specifies whether the stream mode for the path is to be set ON or OFF.

PREFE ON | OFF
specifies whether or not this path is to be the preferred path.

PREFI *pppp*
sets the prefix to the specified four characters.

PD F P Examples

```
PD F PATH HS PREFE ON
```

This command specifies that the path to cell HS is the preferred path.

```
PD F PATH CI PREFI CSSN
```

This command specifies that all transactions sent along path CI are to be prefixed by CSSN.

PD F T—Modify Transaction Identifier Entry

This command modifies an entry in the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set. See the insert (PD I T) command for additional information on the items that can be modified.

PD F	**T[ID]** *tran-id* **{TE {ON | OFF} |** **PU {ON | OFF} |** **PR {ON | OFF} |** **H {ON | OFF} |** **S {ON | OFF} |** **RE {ON | OFF} |** **PA** *partition* **|** **TY** *tran-type***}**

tran-id
is a 4-character transaction identifier.

TE ON | OFF
specifies whether the test mode for the transaction identifier is to be set ON or OFF. The use of test mode for testing a transaction processing program is described in the *Programmer's Guide.*

PU ON | OFF
specifies whether the purge mode for the transaction identifier is to be set ON or OFF.

PR ON | OFF
specifies whether the prefind mode for the transaction identifier is to be set ON or OFF.

H ON | OFF
specifies whether the hold mode for the transaction identifier is to be set ON or OFF.

S ON | OFF
specifies whether the stream mode for the transaction identifier is to be set ON or OFF.

RE ON | OFF
specifies the retry mode for the transaction identifier. ON means that when storage is not available for the program, the load is to be retried until it succeeds or until the program dispatcher is shut down. OFF means that the retry counts set by the PD RC command are to be observed.

PA *partition*
sets the partition number for the transaction identifier. The partition must be mapped.

TY *tran-type*
sets the transaction type (2-digit program type code) for the transaction identifier.

PD F T Examples

PD F TID MENU PU ON

This command specifies that the program that processes transaction MENU may be stopped to make its storage available for other programs.

PD F TID LGIT H OFF

This command specifies that LGIT transactions are not to be held.

PD F TID RPT PA 2

This command specifies that the program that processes transaction RPT is to be loaded into partition 2.

PD H—Display PD Commands

This command displays the syntax of individual program dispatcher commands or lists all the PD commands.

PD H	[*command*]

command
is the program dispatcher command to be displayed. A null entry displays a list of all commands and their functions.

PD H Example

PD H

The following information will be displayed:

```
PD HELP PROCESSOR
COMMAND      FUNCTION
Q(UERY)      DISPLAY PROGRAM DISPATCHER TABLES
H(ELP) CMD   ENTER HELP MODE (CMD=COMMAND)
S            START PATH OR TID
P            STOP PATH OR TID
F            MODIFY PATH OR TID TABLE ENTRY
R(EMOVE)     REMOVE PATH OR TID TABLE ENTRY
I(NSERT)     INSERT PATH OR TID TABLE ENTRY
TRAN(S)      SEND A TRANSACTION INTO SYSTEM
C(ELL)       CHANGE CURRENT CELL ID
TRAC(E)      SET TRANSACTION TRACE MODE
M(ESSAGE)    SEND A MESSAGE TO AN EDX TERMINAL
ID           SEND ID CHECK MESSAGE TO A CELL
CP           SEND A CP COMMAND TO A CELL
T(IME)       SEND TIME TO A CELL
RC           SET RETRY COUNTS FOR PROGRAM LOAD
UP           SET USER PGM ($.UPXXXX) LOAD MODE
```

PD I PATH—Insert Path Definition Entry

This command inserts an entry into the path definition table. It does not modify $.SYSPD, the program dispatcher data set. You cannot use this command unless space for additional entries was allocated by a RESERVE statement in $.SYSPD.

PD I	**PATH** *station-name* *,cell-id* **[{,C,***prefix* **| ,S}]** **[P,H]**

station
is a 1- to 8-character name of the station that represents a path to another cell.

cell-id
is a 2-character identifier of the destination cell.

C
indicates that messages sent along this path are to be converted using the pseudo binary algorithm described in the *Design and Installation Guide*. This is used primarily for communication with IMS or CICS on a non-Series/1 host.

prefix
is a 4-character prefix added to all transactions along this path.

S
indicates that this path definition is in 3270 data stream mode. Output goes to this path without modification.

P
indicates that this is the preferred path—the path to use for a transaction with an unknown destination cell identifier.

H
puts the path on a hold queue. You must use a PD Modify command or transaction to release it for use.

PD I PATH Examples

```
PD I PATH REMOTE1,R1
```

This command defines a path to cell R1. It specifies that transactions with cell identifier R1 are to be sent to station REMOTE1.

```
PD I PATH HOSTCRT1,CI,,CSSN P
```

This command defines a path to cell CI. It specifies that transactions with cell identifier CI or with an unknown cell identifier are to be prefixed with CSSN and sent to station HOSTCRT1.

Parameter Requirements

You must leave a space between the two groups of optional parameters:

,C,*prefix* P,H
,S P,H

In each group, you must use commas as placeholders for the parameters you don't use before the last parameter you specify. For example:

,,*prefix* ,H

PD I RTE—Insert Reroute Entry

This command inserts a reroute entry into the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set. You cannot use this command unless space for additional entries was allocated by a RESERVE statement in $.SYSPD. The reroute entry defines, for a specific transaction identifier, a cell identifier that overrides the transaction's primary cell identifier.

PD I	**RTE** *tran-id* *cell-id* **[H]**

tran-id
is a 1- to 4-character transaction identifier.

cell-id
is a 2-character identifier of the cell where the transaction is to be routed.

H
puts the transaction on a hold queue. You must use a PD Modify command or transaction to release it for use.

PD I RTE Example

```
PD I RTE RPT S2
```

This command specifies that transaction RPT is to be routed to cell S2.

PD I TID—Insert Transaction Identifier Entry

This command inserts an entry into the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set. You cannot use this command unless space for additional entries was allocated by a RESERVE statement in $.SYSPD.

PD I	**TID** *tran-id* *pgm* **[**,*vol-id*,{*part* | **0**},*prefind*] [*type*,**P**,**S**,**R**] **[H]**

tran-id
is a 1- to 4-character transaction identifier.

pgm
is a 1- to 8-character name of the program that is to process the transaction.

vol-id
is a 1- to 6-character volume name of the program. The default is the IPL volume.

part
is the number of the partition where the program is to be loaded. You can specify one of the following:

1 to 8
means that the specified partition is used. The partition must be mapped.

0
means that any available mapped partition is used.

-1 to -8
means that any available mapped partition is used except the one specified.

CF
means that the $.CF partition is used.

NCF
means that any partition is used except the $.CF partition.

prefind
indicates whether or not a prefind of the program's data sets and overlays is to be performed when this TID entry is inserted. Omit the parameter to enable prefind; specify N to disable prefind.

type
is a code identifying the type of program. Valid codes are:

10

- is the default
- is a single transaction program
- the transaction is not sent to the program
- a station is not created for the program

11

- is a single transaction program
- the transaction is not sent to the program
- a station is created for the program

12

- is a single transaction program
- the transaction is sent to the program
- a station is created for the program

13

- is a single transaction program
- the transaction is sent to the program
- a station is created for the program
- the program is not loaded

20

- is a multiple transaction program
- the transaction is not sent to the program
- a station is not created for the program

21

- is a multiple transaction program
- the transaction is not sent to the program
- a station is created for the program

22

- is a multiple transaction program
- the transaction is sent to the program
- a station is created for the program

23

- is a multiple transaction program
- the transaction is sent to the program
- a station is created for the program
- the program is not loaded

30

- is a never ending program
- the transaction is not sent to the program
- a station is not created for the program

31

- is a never ending program
- the transaction is not sent to the program
- a station is created for the program

32

- is a never ending program
- the transaction is sent to the program
- a station is created for the program

33

- is a never ending program
- the transaction is sent to the program
- a station is created for the program
- the program is not loaded

40

- is a never ending, reentrant program
- the transaction is not sent to the program
- a station is not created for the program

41

- is a never ending, reentrant program
- the transaction is not sent to the program
- a station is created for the program

42

- is a never ending, reentrant program
- the transaction is sent to the program
- a station is created for the program

43

- is a never ending, reentrant program
- the transaction is sent to a program
- a station is created for the program
- the program is not loaded

P

indicates that the program may be stopped to make its storage available for other transaction-processing programs. This is only valid for program types 30 to 43.

S

indicates that this transaction is in 3270 data stream mode. Output goes to the transaction program station without modification.

R

indicates that the program load retry counts set by the PD RC command are to be ignored for this transaction. When there is no storage available for the program, the load is to be retried until it succeeds or until the program dispatcher is shut down.

H

puts the transaction on a hold queue. You must use a PD Modify command or transaction to release it.

PD I TID Examples

PD I TID MENU $.WSMENU 22

This command specifies that transactions with identifier MENU are to be sent to $.WSMENU, which is a type 22 program. The program is to be loaded into any available partition. When the insert command is processed, a prefind of $.WSMENU's data sets and overlays is performed.

PD I TID RPT REPORT,RPTLIB,,N H

This command specifies that transactions with identifier RPT are to be held; and that when the hold is released, program REPORT is to be loaded. REPORT is a type 10 program on volume RPTLIB. No prefind of its data sets and overlays is performed.

PD I TID T327 T327PGM 42,,S

This command specifies that transactions with identifier T327 are 3270 data streams that are to be sent without modification to program T327PGM.

Parameter Requirements

You must leave a space between the groups of optional parameters:

,vol,part,prefind type,P,S,R H

In each group, you must use commas as placeholders for the parameters you don't use before the last parameter you specify. For example:

,vol,,prefind ,,S

PD ID—Check Active Cell

This command sends a transaction to a cell to check whether it is active.

An active cell sends back a message to the EDX terminal where the command was issued.

PD ID	[*cell-id*]

cell-id
is a 2-character identifier of the cell to be checked. If it is not specified, all cells will be checked.

PD ID Example

PD ID S2

This command sends an ID check transaction to cell S2. If cell S2 is active, you will get the following response where TERM3 is the name of the EDX terminal:

TERM3 CELL S2 ACTIVE

PD M—Send Message

This command sends a message to an EDX terminal or printer.

PD M	{*cell-id* | * | **} {*terminal-name* | *} *message*

cell-id
is a 2-character identifier of the cell where the message is to be sent. One asterisk indicates this cell. Two asterisks indicate all cells.

terminal-name
is a 1- to 8-character name of the EDX terminal or printer where the message is to be displayed. An asterisk indicates all EDX terminals and printers.

message
is the data to be sent, 1 to 57 characters.

Special Considerations

If you send a message to:

- A printer that is powered-off or disabled, the program dispatcher waits until the printer is available.
- To an unidentified terminal, the message will be sent to the terminal that started $.PD.
- To the system log device, the message may be lost.
- To a specific EDX terminal or printer in a cell that cannot be found, the message will be sent to the terminal where $.PD was loaded.

PD M Examples

```
PD M S4 $SYSLOG ARE YOU THERE?
```

This command sends the message 'ARE YOU THERE?' to the terminal named $SYSLOG in cell S4.

```
PD M ** * A BROADCAST
```

This command sends the message 'A BROADCAST' to every terminal and printer in every cell known to the program dispatcher.

PD P P—Stop Path Definition Entry

This command deactivates an entry in the path definition table. It does not modify $.SYSPD, the program dispatcher data set.

PD P	**{P[ATH]}** *cell-id* [*cell-id1...cell-idn*]

cell-id
is a 2-character cell identifier. You may specify more than one cell identifier.

PD P P Example

PD P PATH R1 R2

This command deactivates the paths to cells R1 and R2.

PD P T—Stop Transaction Identifier Entry

This command deactivates an entry in the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set.

PD P	**T[ID]** *tran-id* [*tran-id1...tra -idn*]

tran-id
is a 1- to 4-character transaction identifier. You may specify more than one transaction identifier.

PD P T Example

PD P TID MENU

This command deactivates transaction identifier MENU.

PD Q—Display Program Dispatcher Tables

This command displays both the transaction identifier table and the path definition table used by the program dispatcher, or the table specified.

PD Q	**{ALL | PATH | TID}** [*destination*]

ALL
displays the transaction identifier table and the path definition table. This is the default.

PATH
displays the path definition table.

TID
displays the transaction identifier table.

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

PD Q Examples

PD Q PATH

This command displays the path definition table:

```
16:05:42 PROGRAM DISPATCHER DISPLAY FOR CELL:  SJ TYPE:  PATH
CELL STATION MSGS MSGR #ERS STAT PFIX
 HT  T771    0000 0000 0000 9202
 S2  C770A   0000 0000 0000 8201
 S3  C770B   0000 0000 0000 8201
  4  OPEN TABLE ENTRIES
```

where:

CELL
is a cell identifier.

STATION
is the name of the station that is the path to the cell.

MSGS
is the number of transactions sent over the path.

MSGR
is the number of transactions received over the path.

#ERS
is the number of errors.

STAT
is a 16-bit value, the status of the path:

```
0123 4567 89AB CDEF
X... .... .... ....  1 = Path active
.X.. .... .... ....  1 = Path in hold mode
..X. .... .... ....  1 = Convert data
...X .... .... ....  1 = Preferred path
.... 00.. .... ....  Reserved
.... ..X. .... ....  1 = Path station defined in $.SYSNET
.... ...X .... ....  1 = Hold station defined in $.SYSNET
.... .... 0... ....  Reserved
.... .... .X.. ....  1 = 3270 data stream mode
.... .... ..00 00..  Reserved
.... .... .... ..X.  1 = Emulated 3270 control unit path station
.... .... .... ...X  1 = Real 3270 control unit path station
```

PFIX
is the transaction prefix.

PD Q TID $SYSPRTR

This command displays the transaction table on $SYSPRTR:

```
16:05:45 PROGRAM DISPATCHER DISPLAY FOR CELL:  SJ TYPE:  TID
 ID     PGM NAME VOLUME PT TY PGM ORGN LN VDEA #RLD RS PS OS AS STAT #MSG #ERS
HMU     $.HMU    EDX002  0 22 00002251 0E 26F4 00E8 2A 0C 1A 04 8000 0000 0000
WSC     $.WSC    EDX002  0 42 000043B1 29 26F4 02F8 2C 23 00 09 8000 0000 0000
MENU    $.WSMENU EDX002  0 22 0000223F 0F 26F4 0160 0C 0C 00 00 8000 0000 0000
IPL     $.PDIPL  EDX002  0 12 000030FD 14 26F4 02C4 0E 0E 00 00 8000 0000 0000
<IO>    $.PD<IO> EDX002  0 22 0000124E 26 26F4 034E 12 12 00 00 8000 0000
  10    OPEN TABLE ENTRIES
```

where:

ID
is the transaction identifier.

PGM NAME
is the name of the program that processes the transaction.

VOLUME
is the volume where the program resides.

PT
is the partition into which the program is to be loaded.

TY
is the program type code.

PGM ORGN
is the first record number of the program.

LN
is the length of the program in records.

VDEA
is the address of the program's volume descriptor entry.

#RLD
is the number of RLD (relocatable dictionary) entries in the program.

RS
is the total storage size of the program in pages, the sum of PS, OS, and AS.

PS
is the program size in pages.

OS
is the program's overlay size in pages.

AS
is the program's auxiliary storage size in pages (as defined by the STORAGE parameter of PROGRAM statement).

STAT
is a 16-bit value, the status of the transaction identifier:

```
0123 4567 89AB CDEF
X... .... .... .... 1 = TID active
.X.. .... .... .... 1 = TID in hold mode
..X. .... .... .... 1 = TID in test mode
...X .... .... .... 1 = Program is purgable
.... X... .... .... 1 = TID is an alias
.... .X.. .... .... 1 = Do not prefind
.... ..X. .... .... 1 = Program station defined in $.SYSNET
.... ...X .... .... 1 = Hold station defined in $.SYSNET
.... .... X... .... 1 = Reroute entry
.... .... .X.. .... 1 = 3270 data stream mode
.... .... ..X. .... 1 = Program load retry sent to $.PD<RI>
.... .... ...X .... 1 = Retry program load forever
.... .... .... 000. Reserved
.... .... .... ...X 1 = Program has been requested to stop
```

#MSG
is the number of transactions processed.

#ERS
is the number of errors.

PD Q PSEUDO—Display Remote Disk Definitions

This command displays the remote disk definitions.

PD Q	**PSEUDO** [*destination*]

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

PD Q PSEUDO Example

PD Q PSEUDO

This command displays the remote disk definitions:

```
 09:36:45 PROGRAM DISPATCHER DISPLAY FOR CELL : 73  TYPE : PSEUDO
PSEUDO    REAL   CELL
 NAME     NAME    ID   VDEA
CB002    EDX002   CB   1738
CB003    EDX003   CB   1766
CBASM    ASMLIB   CB   1794
CD002    EDX002   CD   18A8
CD003    EDX003   CD   18D6
CDASM    ASMLIB   CD   1904
```

where:

PSEUDO NAME
is a pseudo volume name defined in this cell.

REAL NAME
is the corresponding real volume name.

CELL ID
is the identifier of the cell where the real volume exists.

VDEA
is the address of the pseudo volume descriptor entry.

PD Q SCHD—Display Scheduler Entries

This command displays information about transactions that have been scheduled for later processing.

PD Q	SCHD [*destination*]

destination
is the name of the EDX device or the Communications Facility device or terminal station on which the display is to appear. The default is the terminal at which you enter the command.

PD Q SCHD Example

PD Q SCHD $SYSPRTR

This command displays scheduled transactions on $SYSPRTR:

```
 11:18:12 PROGRAM DISPATCHER DISPLAY FOR CELL : CC  TYPE : SCHD
TID   CC T  TIME       DATA
SYST SJ 3 000000 CP READ SHUTNET
SYST SJ 3 000500 CP READ SHUTSJ
```

where:

TID
is the identifier of the scheduled transaction.

CC
is the identifier of the cell where the transaction is to be processed.

T
is the type of scheduler request:

1 = single execution, elapsed time
2 = multiple execution, elapsed time
3 = single execution, actual time
4 = multiple execution, actual time

TIME
is the scheduled time.

DATA
is the transaction data.

PD R P—Remove Path Definition Entry

This command removes an entry from the path definition table. It does not modify $.SYSPD, the program dispatcher data set.

PD R	**P[ATH]** *cell-id* [*cell-id1...cell-idn*]

cell-id
is a 2-character cell identifier. You may specify more than one cell identifier.

PD R P Example

```
PD R P R1 R2
```

This command removes R1 and R2 from the path definition table.

PD R T—Remove Transaction Identifier Entry

This command removes an entry from the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set.

PD R	**T[ID]** *tran-id* [*tran-id1...tran-idn*]

tran-id
is a 1- to 4-character transaction identifier. You may specify more than one transaction identifier.

PD R T Example

```
PD R T MENU
```

This command removes MENU from the transaction identifier table.

PD RC—Set Retry Counts

This command sets the number of times the program dispatcher will attempt to load a program when storage is not available. When storage is not available, the program dispatcher requests programs defined as purgable to stop; waits 200 milliseconds; and retries the load. If you do not use this command to set the retry counts, the program dispatcher tries loading the program 10 times, processes other pending transactions, tries loading the program 10 more times, and so on for a total of 30 attempts.

PD RC	*retries* *maximum*

retries
specify a value from 1 to 99 to indicate the number of times the program dispatcher is to try loading a program before processing other pending transactions. Specify 0 to indicate no retries.

maximum
specify a value from 0 to 999 to indicate the maximum number of times the program dispatcher is to try loading a program.

PD RC Example

PD RC 5 20

This command specifies that when storage for a program is not available, the program dispatcher is to try loading the program 5 times, process other pending transactions, try loading the program 5 more times, and so on for a total of 20 attempts.

PD S P—Start Path Definition Entry

This command activates an entry in the path definition table. It does not modify $.SYSPD, the program dispatcher data set.

PD S	**P[ATH]** *cell-id* [*cell-id1...cell-idn*]

cell-id
is a 2-character cell identifier. You may specify more than one cell identifier.

PD S P Example

PD S P R1 R2

This command activates the paths to cells R1 and R2.

PD S T—Start Transaction Identifier Entry

This command activates an entry in the transaction identifier table. It does not modify $.SYSPD, the program dispatcher data set. It performs a prefind of the transaction program, if the transaction identifier entry so specifies.

PD S	**T[ID]** *tran-id* [*tran-id1...tran-idn*]

tran-id
is a 1- to 4-character transaction identifier. You may specify more than one transaction identifier.

PD S T Example

PD S TID MENU

This command activates transaction identifier MENU.

PD T—Send Time and Date

This command sends the time and date to a cell. The cell uses it as the system time and date.

PD T	*cell-id*

cell-id
is a 2-character identifier of the cell where the current time and date are to be sent.

PD T Example

PD T S3

This command sends the current time and date to cell S3.

PD TRAC—Trace Transaction

This command starts or stops the program dispatcher transaction trace. See the *Programmer's Guide* for additional information on use of the trace command.

PD TRAC	**{ON | OFF}** [{*destination* | *}] **[X]**

ON | OFF
turns the trace on or off.

destination
is the name of the EDX device or the station to receive the trace output.

*
displays the trace on the terminal where the PD TRAC command was entered; this is the default.

X
indicates that the trace output is to be hexadecimal; character is the default.

Special Considerations

If the trace destination is the system log device, you may lose some of the trace output.

PD TRAC Example

```
PD TRAC ON $SYSPRTR
```

This command sets the transaction trace on device $SYSPRTR in character form.

PD TRAN—Send Transaction

This command sends a transaction to the program dispatcher. When you enter the command from a terminal, you may omit the prefix PD.

PD TRAN	*trans*

trans
is the transaction to be sent.

PD TRAN Examples

PD TRAN SCHDHSLGITHS3024500

> TRAN SCHDHSLGITHS3024500

Each of these commands sends a scheduler transaction (SCHD) to cell HS. The transaction specifies that transaction LGIT in cell HS is to be processed once (code 3) at 02:45:00.

PD UP—Set User Program Option

This command specifies whether or not the program dispatcher is to load program $.UP*xxxx* when it has no entry in its transaction identifier table for *xxxx*. The system default is to load $.UP*xxxx* for undefined transactions. If you have no such programs, you should specify that this not be done to avoid wasting time on unsuccessful load attempts.

PD UP	**{ON | OFF}**

ON | OFF
enables (ON) or disables (OFF) load of $.UP*xxxx*.

PD UP Example

PD UP OFF

This command specifies that the program dispatcher is not to load program $.UP*xxxx* when it receives undefined transaction *xxxx*.

TRAN—Send a Transaction

This command sends a transaction to the program dispatcher. It provides the same function as the PD TRAN command, but it can be entered only from a terminal.

TRAN	*trans*

trans
is the transaction to be sent.

TRAN Example

> TRAN SCHDHSLCITHS3024500

This command sends a scheduler transaction (SCHD) to cell HS. The transaction specifies that transaction LCIT in cell HS is to be processed once (code 3) at 02:45:00.

WSC—Start a Work Session Controller Terminal

This command initiates communication between the terminal where it is entered and a local work session controller application program. It generates an SD (start device) transaction and sends it to the work session controller in the local cell. The SD transaction is described in the *Programmer's Guide*. The command can be entered only from a terminal.

WSC	[*tran-id* [*workarea-size* [*terminal-name*]]]

tran-id
is the transaction identifier defined for the program with which you want to communicate. The default is MENU.

workarea-size
is the size of the user work area to be used in the session, 0-128 bytes. The default is 0.

terminal-name
is the terminal name to be used in the session. The default is the reverse of the EDX name of the terminal where the command is entered.

WSC Examples

```
> WSC PRG3 0 TERM48C
```

This command initiates communication with the work session controller application program whose transaction identifier is PRG3. The program does not need a user work area. It will use the name TERM48C to communicate with the terminal.

```
> WSC
```

This command initiates communication with the work session controller application program whose transaction identifier is MENU. The program does not need a user work area. If the command is entered from EDX terminal $SYSLOGA, the program will use the name AGOLSYS$ to communicate with the terminal.

IBM-Supplied Transactions Reference

The Communications Facility provides certain functions through transactions. This chapter defines the format of these transactions and gives an example of each.

The transactions may be read from a data set, sent by a program, or entered from a terminal using either form of the TRAN command:

> TRAN transaction
> PD TRAN transaction

Transaction Functions

The remote IPL transaction (IPL) performs an IPL of a Series/1.

The scheduler transaction (SCHD) schedules a transaction to be performed at a later time.

The system transaction (SYST) performs a variety of functions. The program dispatcher uses system transactions to process program dispatcher PD commands and to control the execution of transaction processing programs. The functions available for your use are described in this chapter.

Transaction Format

The first part of a transaction message is a fixed-format header that identifies the transaction and the cell where it is to be processed.

The format of a transaction message is:

1-4	5-6	7-10	11-12	13-*n*
tid1	*c1*	*tid2*	*c2*	*transaction data*

where:

tid1
is the primary transaction identifier, a 4-character code that identifies the transaction.

c1
is the primary cell identifier, a 2-character code that identifies the cell where the transaction is to be processed. Two blanks or 00 means that the transaction is to be processed in the cell where it originated. Two asterisks mean that the transaction is a broadcast transaction, which is to be processed in all cells known to the program dispatcher.

tid2
is the secondary transaction identifier, whose meaning is defined by the program that processes the transaction.

c2
is the secondary cell identifier, whose meaning is defined by the program that processes the transaction. When *c2* is ??, the program dispatcher replaces it with the ID of the cell where the transaction originated.

transaction data
: is whatever data is sent to the program that processes the transaction.

The minimum length of a transaction message is 6 bytes. The data portion is optional. If a transaction contains no data, the secondary transaction identifier and secondary cell identifier are optional.

The letter *b* represents a blank in the transaction format definitions and examples in this chapter.

IPL—Remote IPL Transaction

This transaction IPLs an EDX supervisor at a remote node; the Communications Facility doesn't support IPL of a Realtime Programming System supervisor. The secondary transaction is an optional acknowledgment that reports whether or not the IPL was successful.

You can send a remote IPL transaction from any Series/1. Only cells (Series/1s in which the program dispatcher is running) can process IPL transactions. The cell that processes an IPL transaction must have a transaction identifier entry in $.SYSPD that causes the remote IPL program, $.PDIPL, to be loaded when an IPL transaction is received. The required entry is:

    TID IPL $.PDIPL 22

The line connected to the remote Series/1 can be either a BSC line or Local Communications Controller line. Cells with a BSC line require either of the following I/O control programs for remote IPL support:

- $.IO0AC0
- $.IO0A10

Cells using the Local Communications Controller ring require the following I/O control program for remote IPL support:

- $.IO0AB0

When Series/1s are connected on a Local Communications Controller ring, you can initiate a remote IPL by pressing the load button at a Series/1 where the Local Communications Controller is the IPL device. This causes a hardware IPL request to be sent to each Series/1 on the ring. In those nodes where the remote node (the one where the load button was pressed) is an active station, program $.IO0AB0 sends a transaction which IPLs the remote node.

When the communications path used by the remote disk support is the Local Communications Controller ring and the Communications Facility program dispatcher is not active in the remote node, the node with the physical disk must be IPLed first. The Communications Facility program dispatcher and the Local Communications Controller IOCP, $.IO0AB0, must be active to access the remote disk from the remote node.

If a node containing remote disk supervisor support is IPLed while a remote disk operation is in progress, unpredictable results can occur.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-16	17-22	23-37
IPLb	*c1*	*tid2*	*c2*	*c3*	*tp*	*vol-id*	*progname,vol-id*

where:

c1
: is a 2-character identifier of the cell where the IPL transaction is to be processed.

tid2
: is a 4-character identifier used as the primary transaction identifier in the acknowledgment transaction. Specify blanks if an acknowledgment is not wanted.

c2
: is a 2-character identifier used as the primary cell identifier in the acknowledgment transaction.

c3
: is a 2-character identifier of the cell to be IPLed.

tp
: is a code that indicates the type and address of the IPL device. Specify one of the following:

- D2 is device 4962 at hexadecimal address 03.
- D3 is device 4963 at hexadecimal address 48.
- D4 is device 4964 at hexadecimal address 02.
- D5 is device 4965 at hexadecimal address 45.
- D6 is device 4966 at hexadecimal address 22.
- D7 is device 4967 at hexadecimal address C0.
- D8 is the 4950 or 5170-495 disk at hexadecimal address 03.
- D9 is the IDSK disk device at hexadecimal address 60.
- B9 is the BSC or Local Communications Controller attachment at any hexadecimal address less than C4. You cannot specify this type of IPL for a 4950 or a 5170-495 Series/1 system unit.
- L0 is an indirect reference to an IPL device.
- 30 is device DDSK-30 or DDSK-60 at hexadecimal address 44.

Code types D2-D9 and 30 are for disk or diskette devices. An IPL bootstrap sent across the line for these devices causes the nucleus to be loaded from disk or diskette. You must use $INITDSK to store an IPL bootstrap in the normal IPL location.

An IPL bootstrap sent across the line for code type B9 allows the nucleus to be loaded from another Series/1 in the network. The BSC attachment card in the remote Series/1 must be strapped for remote IPL. The Local Communications Controller must be strapped to allow IPL also. See the *Design and Installation Guide* for additional information about strapping for remote IPL.

Code type L0 is used by program $.IO0AB0 when it responds to a hardware IPL request. In a transaction with code type L0, positions 17-24 contain the remote node station name. When $.PDIPL receives such a transaction, it searches the IPL transaction data set, $.SYSLCC, for an entry for the remote node station. If it finds the entry, it performs the remote IPL using the information from the entry. See the *Design and Installation Guide* for additional information about data set $.SYSLCC.

vol-id
is the 6-character name of the volume in the remote Series/1 to be used as the default IPL volume for code type B9 only.

progname,vol-id
is the program name and volume of the copy of the EDX supervisor to be transmitted to the remote Series/1 for code type B9 only.

IPL Examples

IPLbS2ABCDbbS3D2

This transaction IPLs cell S3 from cell S2 using device 4962 at cell S3. The acknowledgment transaction is:

ABCDbbIPLbS2S3D2*xx*

where *xx* is one of the following return codes:

- Y0 means the IPL was successful.
- E0 means invalid code type was specified.
- E1 means unknown cell identifier was specified.
- E2 means the send of a bootstrap failed.
- E3 means the EDX supervisor to be transmitted was not found (code type B9 only).
- E5 means that $.PDIPL could not locate the station representing the path to the remote cell.
- E6 means error detected in dialog with remote Series/1.
- E7 means the EDX supervisor to be sent is larger than X'FE00' in partition 1 (code type B9 only). Use the EDX utility $DISKUT2 to determine the partition 1 size of the nucleus.

- E8 means there is no entry for the remote node station in $.SYSLCC (code type L0 only).

- E9 means error while attempting to read a disk data set.

- EA means $.PDIPL abended with a program or machine check.

- EB means there is no more space in dynamic storage to buffer an additional pending IPL request.

- EC means the remote Series/1 is a Communications Manager node.

SCHD—Scheduler Transaction

This transaction schedules transaction processing for a specific time of day or after a given period of time. Processing begins within 11 seconds of the specified time. The secondary transaction is the one to be scheduled.

Do not change the system time between scheduling a transaction and its execution. If you do, the transaction may not be processed when you intended, or it may never be processed.

The cell that processes an SCHD transaction must have a transaction identifier entry in $.SYSPD that causes the scheduler program, $.PDSCHD, to be loaded when an SCHD transaction is received. The required entry is:

TID SCHD $.PDSCHD,,,N 32

The transaction format is:

1-4	5-6	7-10	11-12	13	14-15	16-17	18-19	20-35
SCHD	*c1*	*tid2*	*c2*	*t*	*hh*	*mm*	*ss*	*data*

where:

c1
: is a 2-character identifier of the cell where the scheduler program is to be executed.

tid2
: is a 4-character transaction identifier used as the primary transaction identifier when the scheduler request has been satisfied.

c2
: is a 2-character identifier of the cell where the transaction is to be sent when the scheduler request has been satisfied.

t
: is a 1-character code that indicates the type of scheduler request:

0 cancel scheduler request. The rest of the transaction must be identical to the one that you are canceling.

1 single execution, elapsed time

2 multiple execution, elapsed time

3 single execution, actual time

4 multiple execution, actual time.

hh
: is a 2-character representation of hours.

mm
: is a 2-character representation of minutes.

ss
is a 2-character representation of seconds.

data
is any 16 bytes of user supplied data.

SCHD Example

SCHDC1ABCDbb1000030DATADATADATADATA

After a delay of 30 seconds, the following transaction is sent to this cell:

ABCDbbSCHDC1DATADATADATADATA

SYST CP—CP Command Transaction

This transaction sends a command to the specified cell.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-87
SYST	*cl*	bbbb	bb	**CP**	*command*

where:

cl
is the 2-character identifier of the cell where the transaction is to be processed.

command
is any valid command processor CP command, 1 to 73 characters.

SYST CP Example

SYSTS3bbbbbbCPSbE32776

This transaction starts station E32776 in cell S3.

SYST HL—High Speed Loader Transaction

This transaction causes the high speed loader to load a program. It can be issued only by a program. When this transaction arrives at the program dispatcher, the data in the transaction is passed directly to the high speed loader, which loads the specified program. No station is created, and no data is sent to the program.

1-4	5-6	7-10	11-12	13-14	15-62
SYST	*cl*	bbbb	bb	**HL**	*tid entry*

where:

cl
is the 2-character identifier of the cell where the transaction is to be processed.

tid entry
is a single entry in the transaction identifier table. See the *Debugging Guide* for a description of the format and content of a transaction identifier entry.

SYST ID—ID Check Transaction

This transaction causes the program dispatcher in the specified cell to send a cell active message to a terminal. The secondary transaction is a SYST MS (message) transaction.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-22
SYST	*c1*	**SYST**	*c2*	**ID**	*terminal-name*

where:

c1
is the 2-character identifier of the cell where the transaction is to be processed.

c2
is the 2-character identifier of the cell where the message is to be sent.

terminal-name
is the name of the EDX terminal to receive the message, 1 to 8 characters.

SYST ID Examples

```
SYSTSJSYSTS2ID$SYSLOGA
```

This transaction causes cell SJ to send a cell active message to terminal $SYSLOGA in cell S2. The resulting secondary transaction is:

```
SYSTS2SYSTSJMS$SYSLOGACELL SJ ACTIVE
```

The secondary transaction sends the message 'CELL SJ ACTIVE' to $SYSLOGA in cell S2.

SYST MS—Message Transaction

This transaction sends a message to EDX terminals and printers.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-22	23-74
SYST	*cl*	bbbb	bb	**MS**	*terminal-name*	*message*

where:

cl
is the 2-character identifier of the cell where the transaction is to be processed. You may enter ** to indicate all cells.

terminal-name
is the 8-character name of the EDX terminal or printer to receive the message. You may enter ******** to display the message on all EDX terminals and printers.

message
is the message to be displayed, 1 to 52 characters.

Special Considerations

If you send a message to a printer that is powered off or disabled, the program dispatcher waits until the printer is available.

If you send a message to the system log device, the message may be lost.

SYST MS Example

SYSTS3bbbbbbMS$SYSLOGbTHIS IS A TEST MESSAGE

This transaction sends the message 'THIS IS A TEST MESSAGE' to $SYSLOG in cell S3.

SYST RC—Retry Counts Transaction

This transaction sets the number of times the program dispatcher will attempt to load a program when storage is not available. When storage is not available, the program dispatcher requests programs defined as purgable to stop, waits 200 milliseconds, and retries the load.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-16	17-19
SYST	*cl*	bbbb	bb	**RC**	*rc*	*max*

where:

cl
: is the 2-character identifier of the cell where the transaction is to be processed.

rc
: is a value from 00 to 99 to indicate the number of times the program dispatcher is to try loading a program before processing other pending transactions. The system default is 10.

max
: is a value from 000 to 999 to indicate the maximum number of times the program dispatcher is to try loading a program. The system default is 30.

SYST RC Example

SYSTbbbbbbbbRC05020

This transaction specifies that when storage for a program is not available, the program dispatcher is to try loading the program 5 times, process other pending transactions, try loading the program 5 more times, and so on for a total of 20 attempts.

SYST RH—Release Hold Transaction

This transaction releases the hold on a path or transaction.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-22
SYST	*c1*	bbbb	bb	**RH**	*station-name*

where:

c1
: is the 2-character identifier of the cell where the transaction is to be processed.

station-name
: is the name of the station where transactions are held. For a held path, the station name is $.PH*xx*, where *xx* is the identifier of the remote cell. For a held transaction, the station name is $.PH followed by the transaction identifier.

SYST RH Examples

SYSTbbbbbbbbRH$.PHABCD

This transaction releases transactions with identifier ABCD.

SYSTbbbbbbbbRH$.PHS3

This transaction releases transactions destined for cell S3.

SYST SP—Start Remote Disk Transaction

This transaction causes remote disk support to release control of the BSC or Local Communications Controller connection over which remote disks can be accessed before the program dispatcher is active. Thereafter, remote disk support sends remote disk I/O transactions to the program dispatcher.

You issue this transaction in a cell that accesses remote disks after the program dispatcher has started. If remote disk is defined in your configuration, you'll need to send this transaction as part of your regular startup routine. You can include it in the $.SYSPD data set.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-16	17-18
SYST	*cl*	bbbb	bb	**SP**	*address*	*cell-id*

where:

cl
: is the 2-character identifier of the cell where the transaction is to be processed.

address
: is the 2-character device address of a remote disk, as defined in the Series/1 that accesses remote disks. If you have defined more than one remote disk, specify the address of any one of them.

cell-id
: is the 2-character identifier of the cell where the real disks exist.

SYST SP Example

```
SYSTbbbbbbbbSPE3SJ
```

This transaction causes remote disk support to start using the program dispatcher to access remote disks located in cell SJ. The device address of the remote disk (or of one of them) is E3.

SYST ST—Station Detection Transaction

This transaction determines whether or not a station exists in the specified cell. The secondary transaction is an acknowledgement that reports the result.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-22
SYST	*c1*	*tid2*	*c2*	**ST**	*station-name*

where:

c1
: is the 2-character identifier of the cell where the transaction is to be processed.

tid2
: is the 4-character identifier used as the primary transaction identifier in the acknowledgement transaction.

c2
: is the 2-character identifier used as the primary cell identifier in the acknowledgement transaction.

station-name
: is the name of the station, 1 to 8 characters.

Special Considerations

The program dispatcher accepts LU in place of ST in positions 13-14 of the transaction.

SYST ST Examples

SYSTS9HMUbS2ST$.RMU

This transaction checks if station $.RMU exists in cell S9. The acknowledgement transaction sent to cell S2 is:

HMUbS2SYST??ST$.RMUbbb*x*

where *x* is N if the station is not there or Y if the station is there.

SYST TI—Time Transaction

This transaction sets the system time and date in a cell.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-16	17-18	19-20	21-22	23-24	25-26
SYST	*cl*	*bbbb*	*bb*	TI	*hh*	*mm*	*ss*	*mo*	*dy*	*yr*

where:

cl
is the 2-character identifier of the cell where the transaction is to be processed.

hh
is the 2-character representation of hours.

mm
is the 2-character representation of minutes.

ss
is the 2-character representation of seconds.

mo
is the 2-character representation of the month.

dy
is the 2-character representation of the day.

yr
is the 2-character representation of the year.

SYST TI Example

SYSTS4bbbbbbTI161232022682

This transaction sets the system time to 16:12:32 and date to 02/26/82 in cell S4.

SYST TR—Trace Transaction

This transaction sets transaction trace mode on or off.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15	16-25	26
SYST	*c1*	bbbb	bb	**TR**	*x*	*terminal*	*t*

where:

c1
: is the 2-character identifier of the cell where the transaction is to be processed.

x
: specifies O for trace on or F for trace off.

terminal
: is the name of the EDX terminal where the trace is to be displayed, 8 characters followed by two blanks. You may also specify a station where the trace output is to be sent by entering D=station-name.

t
: specifies C or blank for a character trace or X for a hexadecimal trace.

SYST TR Example

SYSTS3bbbbbbTRO$SYSLOGAbbC

This transaction sets transaction trace on in cell S3. The trace data is to be displayed on terminal $SYSLOGA as characters.

SYST UP—User Program Transaction

This transaction specifies whether or not the program dispatcher is to load program $.UP*xxxx* when it has no entry in its transaction table for *xxxx*.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15
SYST	*cl*	bbbb	bb	**UP**	*x*

where:

cl
: is the 2-character identifier of the cell where the transaction is to be processed.

x
: specify O to enable load of $.UP*xxxx* or F to disable load of $.UP*xxxx*. O is the system default.

SYST UP Example

SYSTbbbbbbbbUPF

This transaction specifies that the program dispatcher is not to load program $.UP*xxxx* when it receives undefined transaction *xxxx*.

SYST WH—Program Detection Transaction

This transaction determines whether or not a program is loaded in the specified cell. The secondary transaction is an acknowledgement that reports the result.

The transaction format is:

1-4	5-6	7-10	11-12	13-14	15-22
SYST	*c1*	*tid2*	*c2*	**WH**	*program-name*

where:

c1
: is the 2-character identifier of the cell where the transaction is to be processed.

tid2
: is the 4-character identifier used as the primary transaction identifier in the acknowledgement transaction.

c2
: is the 2-character identifier used as the primary cell identifier in the acknowledgement transaction.

program-name
: is the name of the program, 1 to 8 characters.

SYST WH Examples

```
SYSTS9PGM1S2WH$.WSC
```

This transaction checks if program $.WSC is loaded in cell S9. The acknowledgement transaction sent to cell S2 is:

PGM1S2SYST??WH$.WSCbbb*x*

where *x* is N if the program is not there or Y if the program is there.

The Communications Facility provides utility programs to help you configure and operate your Communications Facility system:

- $.CONFIG—Configuration Processor. You can use $.CONFIG to define, display, change, and link station definitions in $.SYSNET. You can add or change messages in $.SYSMSG, and you can display the contents of $.SYSMSG and $.SYSNET.

- $.DSINIT—Disk-Queue Data Set Initialization. If you have stations that use the disk-queuing facility, you must use $.DSINIT to format EDX data sets for use as disk-queue data sets.

- $.HMU/$.RMU—Host Management and Remote Management Utilities. You can use $.HMU and $.RMU to control a remote Series/1 from a host Series/1. The host Series/1 operator can allocate and delete data sets on the remote Series/1; send data sets to and receive data sets from the remote system; cause programs to be executed at the remote site; dump the remote processor's storage; and connect a host EDX terminal to the remote system in pass-through mode.

- $.PDBSTS—Volume Protection. You can use this utility to interactively serialize the use of a real disk volume defined as a remote disk in your Series/1.

- $.SETSTG—Set $.CFD Storage. You can use this utility to change the total size of $.CFD's dynamic storage and to specify how it is to be divided between disk-queue file control blocks and the message buffer pool (CFBUF).

- $.UT1—Diagnostic Aid Utility. You can use this utility to find storage addresses of control blocks and tables; display storage; list, allocate, and delete stations; send, receive, and display a station's messages.

- $.UT2—Message Management Utility. You can use this utility to manage message queues by purging storage and disk message queues. You can use $.UT2 to print a report of disk-queued messages; storage-queued messages can be reported only when you purge them.

Messages issued by $.CONFIG, $.DSINIT, $.SETSTG, and $.UT2 are listed in *Messages and Codes*.

$.CONFIG—Configuration Processor Utility Program

The configuration processor, $.CONFIG, is an interactive utility program supplied with the Communications Facility. You can use $.CONFIG to define and link stations, change and display existing station definitions, store and change message text in $.SYSMSG, and display the contents of $.SYSMSG and $.SYSNET.

You can use $.CONFIG from a 4978, 4979, 4980, or 3101 display station. (If your installation does not have one of these terminals, you must use the command processor (CP) commands, described in the "Operator Command Reference" on page 31 to define your network.) The utility will prompt you through the process of entering all the information required to complete the function you select. Error and informational messages appear at the bottom of the screen.

You can use $.CONFIG even when the Communications Facility program ($.CF) has not yet been loaded for execution. You can't use it, however, until you've generated a supervisor that includes support for the Communications Facility language extension instructions.

You must load $.CONFIG into a mapped partition.

Using $.CONFIG

To gain access to $.CONFIG, enter this command:

```
> $L $.CONFIG
```

The utility will then prompt you for the names and volumes of the $.SYSNET and $.SYSMSG data sets to be used. $.SYSNET is the data set that contains all the station definitions for a node. $.SYSMSG is the data set that contains the text of log messages. When the Communications Facility is running, the data sets must have those names and reside on the same volume as the control program. You can, at your option, use $.CONFIG to prepare copies of these data sets with other names or on other volumes.

$.CONFIG then waits for you to enter any of the following commands:

- ASSIST—Display or suppress extra prompts
- CHANGE—Change an existing station definition
- COPY—Create a station definition from an existing station
- DEFINE—Define a station
- DISPLAY—Display a station definition
- EDIT—Store or alter message text in $.SYSMSG
- HELP or ?—Display $.CONFIG commands
- LIST—Display the contents of $.SYSMSG and $.SYSNET
- LINK—Link stations
- SET EXIT—Set the exit code
- END—Terminate utility

If you enter HELP or ?, $.CONFIG displays the screen image shown in Figure 14 on page 266.

```
COMMAND            DESCRIPTION
-----------        ----------------------------------------------
ASSIST    (AS)     SUPPRESS/DISPLAY EXTRA PROMPTS
CHANGE    (CH)     MODIFY A STATION DEFINITION
COPY      (CO)     COPY A STATION DEFINITION
DEFINE    (DE)     DEFINE A NEW STATION
DISPLAY   (DI)     DISPLAY A STATION DEFINITION
EDIT      (ED)     CREATE OR EDIT LOG MESSAGE DATA SET
LINK      (LIN)    DEFINE DESTINATION OF MESSAGES
LIST      (LIS)    LIST LOG MESSAGES OR STATION DEFINITIONS
SET EXIT  (SE)     SET COMMAND EXIT CODE
END       (EN)     END PROGRAM


PRESS ENTER TO EXIT HELP MODE
```

Figure 14. Screen Layout after Entering ? to $.CONFIG Initial Screen

Observe the following conventions when you use $.CONFIG:

- You can abbreviate any command word to its first two characters, or as many as are necessary to uniquely identify it.
- Complete each entry by pressing the ENTER key.
- You can terminate any procedure except LIST and HELP by entering the exit code. (EN or whatever you have set it to.)

ASSIST—Display or Suppress Extra Prompts

You can use the ASSIST command to control the amount of prompting that $.CONFIG does. When $.CONFIG is loaded, the assist option is always off; command processors, such as DEFINE or LIST, will do only minimal prompting. If you want additional assistance during command processing, you may enter a question mark (?) in response to any prompt that ends with '(?)'. Information specific to that one prompt is displayed.

The ASSIST command allows you to receive this additional information automatically for the entire $.CONFIG session or any part of the session; you may set assist on or off as you wish. Figure 15 on page 267 shows the ASSIST command prompts. Figure 16 on page 267 shows the LINK prompts after setting ASSIST on. Compare these additional prompts with the information provided in Figure 21 on page 273, when ASSIST is off.

```
$.CONFIG:  DISPLAY OR SUPPRESS EXTRA PROMPTS
------------------------------------------------------------
DO YOU WANT TO SET ASSIST ON? (Y/N): Y

OP17 I ASSIST OPTION: ON
PRESS ENTER TO CONTINUE
```

Figure 15. $.CONFIG ASSIST Command Example

```
$.CONFIG: DEFINE MESSAGE DESTINATIONS          ENTER EN TO EXIT
--------------------------------------------------------------------------
A DIRECT LINK SPECIFIES THE DESTINATION OF A STATION'S MESSAGES.
AN ALTERNATE LINK SPECIFIES THE DESTINATION OF A STATION'S UNDELIVERABLE
TRANSACTIONS, OR THE DESTINATION OF X.25 CONTROL MESSAGES.
DIRECT OR ALTERNATE LINK (D/A) (?): D

ENTER THE NAME OF THE STATION FROM WHICH THE MESSAGES ARE SENT.
FROM STATION NAME (?): PRTDEV1

ENTER THE NAME OF THE STATION THAT IS TO RECEIVE THE MESSAGES.
TO REMOVE AN EXISTING LINK, PRESS ENTER OR ENTER 0000.
TO STATION NAME (?): PRTDEV2

ENTER Y TO LINK THE "FROM" AND "TO" STATIONS TO EACH OTHER.
ENTER N TO JUST LINK THE "FROM" STATION TO THE "TO" STATION.
LINK BOTH WAYS (Y/N) (?): Y

OP45 I STATION: PRTDEV2  LINKED TO STATION: PRTDEV1
OP45 I STATION: PRTDEV1  LINKED TO STATION: PRTDEV2
PRESS ENTER TO CONTINUE
```

Figure 16. $.CONFIG LINK Command with Assist

CHANGE—Change Station Definitions

You can use the CHANGE command to modify a station definition in $.SYSNET. If you need to change the station block (the station definition in storage) you must use a CP Modify command. The CHANGE command won't change the station name nor will it change the station definition in $.SYSNET if the station is active in storage and it has an active disk queue. Use the appropriate CP Modify command to make these changes.

When you enter the CHANGE command, you are prompted for the station name. $.CONFIG then displays the station's definition one line at a time. The format of the display is the same as that for the DEFINE command.

To change a field, type in the new value and press ENTER. To retain the current value of a field, just press ENTER. $.CONFIG displays a new line of information each time you press ENTER.

You may use an asterisk (*) to clear certain optimal parameters from a station definition, such as the disk-queue data set name and SNA LU user data. In the case of a disk-queue name, you may remove disk queuing by entering '0000' in response to the DSNAME(?) prompt.

COPY—Copy Station Definitions

You can use the COPY command to create a station from an existing station definition. COPY requires you to enter the new station name, the network address, and the disk-queue parameters. If these are the only differences between the two stations, you are finished.

If parameters other than the name, network address, and disk-queue parameters differ from the original station, you must then use the $.CONFIG CHANGE command or one of the CP Modify commands to alter those parameters in the new station definition.

Figure 17 on page 269 shows how the screen would appear after you create a new station definition by copying an existing definition.

```
$.CONFIG: COPY STATION DEFINITION                    ENTER EN TO EXIT
.........................................................................................
ENTER 1 TO 8 CHARACTER NAME OF STATION TO BE COPIED
FROM STATION: SAMPLE1

ENTER 1 TO 8 CHARACTER NAME OF STATION TO BE CREATED.
TO STATION: SAMPLE3

ENTER A NETWORK ADDRESS FOR THE NEW STATION, OR PRESS ENTER AND THE
SYSTEM WILL PROMPT FOR INFORMATION TO GENERATE ONE FOR YOU.
ENTER 4-DIGIT HEX NETWORK ADDRESS, OR PRESS ENTER FOR GUIDANCE
NETWORK ADDRESS(?): 0119

IF DISK QUEUING OF MESSAGES WILL BE USED, ENTER THE DATA SET TO BE
USED.  OTHERWISE, PRESS ENTER.
ENTER DISK QUEUE DATASET,VOLUME OR PRESS ENTER
NAME,VOLUME(?):

OP50 I STATION: SAMPLE1 COPIED TO STATION: SAMPLE3
PRESS ENTER TO CONTINUE
```

Figure 17. $.CONFIG COPY Command Example

DEFINE—Define Stations

The DEFINE command creates a station definition entry in the $.SYSNET data set. When you enter the DEFINE command, you are prompted for the station's name, network address, and type. Then you are prompted for more information depending on the station's type.

Figure 18 on page 270 shows how the screen would appear after you define a station that represents a DTE line station.

```
$.CONFIG: STATION DEFINITION                    ENTER EN TO EXIT
----------------------------------------------------------------------------
ENTER 1 TO 8 CHARACTER NAME
NAME: SAMPLE1
ENTER (LINE/DEVICE/USER/TERM/ALIAS/MSG/NODE/PU/LU/CIRC)
TYPE: LI
ENTER (PTPT/CPU/3271C/3271E/LCC/CA/DTE/DCE)
LINE TYPE: DTE
ENTER 4-DIGIT HEX NETWORK ADDRESS, OR PRESS ENTER FOR GUIDANCE
NETWORK ADDRESS(?): 0121
ENTER SIZE (80-32734) OF LINE BUFFER (DEFAULT = 256)
BUFFER SIZE(?):
ENTER SIZE (16/32/64/128/256/512/1024) OF DATA PACKET (DEFAULT IS 128)
PKTSIZ(?):
ENTER VALUE FROM 1 TO 7 AS WINDOW (DEFAULT IS 2)
WINDOW(?):
ENTER 1 TO 15 DIGIT DECIMAL VALUE, OR PRESS ENTER

X.25 LINE ADDRESS(?): 222111444555666
ENTER DISK QUEUE DATASET,VOLUME OR PRESS ENTER
NAME,VOLUME(?): DISKQUE,SAMPLE
IS DISK QUEUING TO BE ACTIVE WHEN STATION IS STARTED? (Y/N): Y
OP10 I STATION: SAMPLE1 DEFINITION COMPLETE
PRESS ENTER KEY TO CONTINUE
```

Figure 18. $.CONFIG DEFINE Command Example

The chapter "Defining Stations" in the *Design and Installation Guide* describes the various types of stations and the information you specify when you define them. Figure 13 on page 134 lists the valid types and subtypes.

DISPLAY—Display Station Definitions

You can use the DISPLAY command to verify the definitions of your stations. This command displays information about the station whose name you enter when you are prompted for a station name.

To review the definitions of multiple stations, enter '*' in response to the station name prompt, and $.CONFIG will display the definition of another station each time you press the ENTER key. To terminate the DISPLAY session, enter 'EN' or the exit code in the command line at the bottom of your screen.

Figure 19 on page 271 shows an example of how the screen might appear after you entered the DISPLAY command and the station name SAMPLE1.

```
$.CONFIG: STATION DISPLAY                          ENTER EN TO EXIT
-------------------------------------------------------------------
                ENTER 1 TO 8 CHARACTER NAME
                NAME: SAMPLE1

                TYPE: LINE
           LINE TYPE: DTE
     NETWORK ADDRESS: 0121
         BUFFER SIZE:   256
              PKTSIZ: 128
              WINDOW: 2
  X.25 LINE ADDRESS: 222111444555666
      DISK BUFFERING: ACTIVE
               USING: DISKQUE,SAMPLE

OP12 I DISPLAY COMPLETE

PRESS ENTER KEY TO CONTINUE
```

Figure 19. Screen Layout after Displaying a Station

EDIT—Create Messages in $.SYSMSG

You can use the EDIT command to create a file of log messages for your application program or to edit a file that already exists—one you previously created, or one of those supplied by the Communications Facility.

The initial screen gives you the option of creating a new file or editing an existing one. You may also list the message IDs. To create a new file, specify C in the ACTION field, and give the file a 2-character ID in the FILE ID field. $.CONFIG creates a file, with the ID you specified, containing 99 messages that all have UNDEFINED MESSAGE as their text.

Note that the Communications Facility itself uses eight file IDs: CA, CF, CP, IO, I1, PD, PN, and SN. Give your file a name different from any of those. When you send a log message, the default file ID is UM. You may want to name your file UM so you can use that default.

To change message text, enter E in the ACTION field, the ID in the FILE ID field, and the number of the message (1-99) you want to change in the MSG# field. The screen will look like Figure 20 on page 272. The current text is displayed in the TEXT NOW IS field. Enter the text you want (1-48 characters) in the NEW TEXT field. To replace the current text, you must enter at least one character (which may be a blank). If you don't enter any characters, the text is not changed. $.CONFIG writes the text to the file when you press ENTER.

```
$.CONFIG: MESSAGE FILE MAINTENANCE                  ENTER EN TO EXIT
------------------------------------------------------------------------
ENTER C = CREATE, E = EDIT, OR L = LIST IDS
ACTION: E

ENTER 2-CHARACTER MESSAGE FILE ID
MSGID: JW

ENTER MESSAGE NUMBER (1-99)
THEN ENTER NEW TEXT, OR PRESS ENTER FOR NO CHANGE

MESSAGE NUMBER: 8
   TEXT NOW IS: UNDEFINED MESSAGE
      NEW TEXT: CALL YOUR SYSTEMS PROGRAMMER

MESSAGE NUMBER: EN
```

Figure 20. $.CONFIG EDIT Command Example

LINK—Link Stations

You can use the LINK command to link one station to another.

$.CONFIG lets you define direct links and alternate links. If you select a direct link, any message sent from the FROM station that has no destination specified will be sent to the TO station. If you select an alternate link, any transaction from the FROM station that cannot be delivered or any X.25 control message is sent to the TO station.

If you want an automatic path in both directions between two stations, specify BOTH WAYS.

Figure 21 on page 273 shows an example of how to use the LINK command to link one station to another.

```
$.CONFIG: DEFINE MESSAGE DESTINATIONS          ENTER EN TO EXIT
--------------------------------------------------------------------------------
DIRECT OR ALTERNATE LINK (D/A) (?): D

FROM STATION NAME (?): SAMPLE1

TO STATION NAME (?): USER

LINK BOTH WAYS (Y/N) (?): Y

OP45 I STATION: USER      LINKED TO STATION: SAMPLE1
OP45 I STATION: SAMPLE1   LINKED TO STATION: USER
PRESS ENTER TO CONTINUE
```

Figure 21. $.CONFIG LINK Command Example

You can also use the LINK command to remove a link by entering '0000' in the 'TO STATION' field.

The LINK command has no effect on station control blocks. To change the links of started stations, use the CP LINK command.

LIST—Display $.SYSMSG and $.SYSNET

You can use the LIST command to display all or selected members of $.SYSMSG or $.SYSNET. To list the contents of a member of $.SYSMSG, enter the member ID (CP, CF, CA, IO, I1, PD, PN, SN, or an ID you have created) after the prompting message. To list all the member IDs or the contents of all members of $.SYSMSG, enter '*' in response to the initial prompt.

To display the contents of $.SYSNET, enter 'NET' in response to the prompt. You then have a choice between listing all station definitions, all stations in a particular node, all stations of a particular type (DEVICE or CIRCUIT, for example), or all stations of a type-subtype combination such as CIRCUIT PVC or TERM 3271.

In response to the prompt LIST ON PRINTER?, enter Y to direct the output to a printer; enter N to direct the output to your terminal. If you enter Y, you will get a prompt requesting the name of the printer.

Figure 22 on page 274 shows an example of a display of all the definitions for circuit stations with a subtype of SVC. The terminal user entered CIRC,SVC in response to the prompt for a NET subset. See the section "CP Q—Display $.SYSNET Station Definitions" on page 131 for an explanation of the data displayed.

```
LISTING OF $.SYSNET ON $.SYSNET,CF2VOL      00/00/00      00:26:42
STATION   TYPE  NA   LINK STAT          S T A T I O N  D E P E N D E N T
-------   ----  ---- ---- ----  ------------------------------------------------
SVC4      16BE AAAA 0000 0000 LINE NAME: DTE      CALLID:  0 USE: STD  CON: WAIT
                                 PROTID: 00000000 USER DATA:
SVC5      16BE AAAA 0000 0000 LINE NAME: DTE      CALLID:  0 USE: STD  CON: USER
                                 PROTID: 00000000 USER DATA:
SVC6      16BE AAAA 0000 0000 LINE NAME: DTE      CALLID: 99 USE: STD+ CON: INIT
                                 PROTID: 00000000 USER DATA:
 .

END LISTING OF $.SYSNET
PRESS ENTER TO CONTINUE
```

Figure 22. $.CONFIG LIST NET Command Example

SET EXIT—Define an Exit Character

You can use the SET EXIT command to define one or two characters, except ? or *, to be used as an exit code. If you enter this code in response to a prompt, $.CONFIG terminates the current command.

If you don't define an exit code, $.CONFIG uses the characters EN. Should you want to enter a text string that begins with the characters EN, $.CONFIG interprets it as a request to end the current command. You must change the exit code to use such a string.

END—Terminate $.CONFIG

The END command terminates the $.CONFIG session.

$.DSINIT—Disk-Queue Data Set Initialization Utility Program

The $.DSINIT program initializes data sets to contain the control information required for disk queuing. $.CF need not be active to use $.DSINIT, but the Communications Facility language extensions must have been included in the EDX supervisor. You must load $.DSINIT into a mapped partition. Data sets to be used for disk queuing must be allocated by the EDX utility program $DISKUT1 and must not be allocated on fixed-head volumes. The data sets must be completely initialized by the $.DSINIT utility program. The size of data sets for disk queuing is discussed in the *Design and Installation Guide*.

If multiple data sets are to be initialized with identical sizes and specifications, it may be more convenient to initialize one of them with $.DSINIT and then make copies, with different names, by using the EDX utility program $COPYUT1.

You can also use $.DSINIT to reinitialize existing disk-queue data sets, either completely or partially.

After you load the $.DSINIT utility program, you'll be prompted to specify:

- The data set name and volume. If you don't specify volume, the one from which $.DSINIT was loaded is used.
- The warning level, expressed as a percentage of total data set capacity, at which you want a message to be sent to the system log when the data set contents have exceeded that level. If you want no warning messages, specify the level as 0.
- The deadband level, expressed as a percentage of total data set capacity, which controls the number of warning messages issued. After a warning message is issued, no more are issued until the data set contents drop below the warning level by the amount of the deadband specified and then rise to the warning level again. The default deadband value is 10%.
- Overlay or nonoverlay mode of operation. In overlay mode, the oldest messages in the data set are overlaid by a new message when the data set has insufficient free space. If you specify a warning level other than 0 for the data set and a message is overlaid, an error message is sent to the system log. In nonoverlay mode of operation, a message is not placed in the data set unless sufficient free space exists. When there is insufficient space for the message, a completion code is returned to the sender indicating that the message was not sent.

A data set that has previously been initialized may be completely reinitialized when it is not being used by a station. Because the reinitialization destroys undelivered messages, the program warns you if the data set contains any of these messages.

Previously initialized data sets may be partially initialized without destroying any messages that may be in the data set.

Parameters that may be reinitialized include capacity warning level, warning deadband value, and overlay/nonoverlay mode of operation. These items may be modified while the data set is in use by a station.

The high water mark (maximum data set level reached during operation) may also be reset if the data set is not in use by a station.

Complete Initialization Example

An example of complete data set initialization follows:

```
> $L $.DSINIT

$.DSINIT      22P,01:52:06, LP= 4000

THIS PROGRAM INITIALIZES PREVIOUSLY ALLOCATED
DATA SETS FOR USE AS MESSAGE QUEUES

ENTER DATA SET NAME,VOLUME: QUEFILE

INITIALIZE QUEFILE,EDX002 ? Y

IS THE DATA SET TO BE COMPLETELY INITIALIZED ? Y

CAPACITY WARNING LEVEL TO BE SPECIFIED ? Y 80

THE DEFAULT DEADBAND VALUE FOR WARNING MESSAGES IS 10%, OK ? Y

IS THE DATA SET TO OPERATE IN OVERLAY MODE ? N

QUEFILE,EDX002  HAS BEEN INITIALIZED FOR MESSAGE QUEUING

MORE DATA SETS TO INITIALIZE ? N

$.DSINIT ENDED AT 01:52:54
```

Partial Initialization Example

An example of partial data set initialization follows:

```
> $L $.DSINIT

$.DSINIT      22P,17:21:47, LP= 4000

THIS PROGRAM INITIALIZES PREVIOUSLY ALLOCATED
DATA SETS FOR USE AS MESSAGE QUEUES

ENTER DATA SET NAME,VOLUME: QUEFILE

INITIALIZE QUEFILE,EDX002 ? Y

IS THE DATA SET TO BE COMPLETELY INITIALIZED ? N

THE CURRENT CAPACITY WARNING LEVEL IS 80%, OK ? N 90

THE CURRENT WARNING MSG DEADBAND VALUE IS 10%,  OK ? N 12

THE CURRENT MODE IS NON-OVERLAY, OK? N

MODE HAS BEEN CHANGED TO OVERLAY

THE CURRENT HIGH WATER MARK IS 12%

SHOULD IT BE RESET TO ZERO? N

NO STATION CURRENTLY USING THIS DATA SET

QUEFILE,EDX002  HAS BEEN INITIALIZED FOR MESSAGE QUEUING
```

```
MORE DATA SETS TO INITIALIZE ? N

$.DSINIT ENDED AT 17:23:26
```

$.HMU AND $.RMU—Host Management and Remote Management Utility Programs

The Communications Facility provides two utility programs, $.HMU and $.RMU, to help you manage the remote Series/1s in your network.

With these utilities you can allocate or delete a data set, execute a program, send and receive data sets, dump storage to disk, and communicate with remote programs from your terminal. These functions are essentially those performed by the EDX utility $RMU, as described in the EDX *Communications Guide*. You should review the information in that book before you use the Communications Facility remote management functions.

You request remote management functions by issuing commands to $.HMU, which is running in your cell. It exchanges transactions with $.RMU, which is running in a remote cell.

$.HMU and $.RMU Requirements

The program dispatcher must be operational in both the host and remote cells. The host cell must have the following entry in its transaction table for $.HMU:

TID HMU $.HMU 22

The remote cell must have the following entry in its transaction table for $.RMU:

TID RMU $.RMU 22

There must also be a valid path defined between the cells. The path can be used by application programs while remote management functions are being performed.

If you want to use the pass-through function, virtual terminal support must be included in the EDX supervisor for the remote system.

$.HMU and $.RMU will run only in a partition that includes the common area.

Starting $.HMU and $.RMU

You start $.HMU in the cell you designate as the host cell. You may start it by:

- issuing the EDX $L command from a partition that includes the common area.
- sending a transaction. The format of the transaction that starts $.HMU is as follows:

  HMUbbbbbbbbb[*name*]

  where *name* is an optional EDX terminal name and b represents a blank. If you specify *name*, that terminal is used for the HMU session. If you omit *name*, the terminal at which you enter the transaction is used.
- entering a 3 in the menu provided by program $.WSMENU.

After starting $.HMU, you'll be prompted for the remote cell designation. The program dispatcher then sends a transaction that causes $.RMU to be started in the remote cell. If no response is received within two minutes, these messages are displayed:

```
NO RESPONSE FROM REMOTE WITHIN 2 MINUTES

DO YOU WISH TO CONTINUE WAITING?
```

You can answer NO to terminate the session or YES to wait another two minutes. When communication between the two programs is established, you'll be prompted to enter one of the following $.HMU commands:

- ?—Display All Commands
- AL—Allocate Data Set
- CR—Change Remote Cell Number
- DE—Delete Data Set
- DU—Dump Storage to Disk(ette) Data Set
- EX—Execute Program at Remote Site
- ID—Check Remote Cell ID
- PT—Pass-Through Processing
- RM—Receive Data Set Member
- SD—Shutdown Remote and Start Optional Program
- SM—Send Data Set Member
- WR—Wrap Message
- EN—Terminate Utility

To interrupt the function you selected, you may enter one of these commands:

- > HMUABORT—to stop the current function
- > HMUSTOP—to stop RM or SM function

A description and example of each command is given in the following sections.

AL—Allocate Data Set

This command allocates a data set on the remote cell.

```
ENTER REMOTE DATA SET NAME (NAME,VOLUME): A,EDX002

ENTER NUMBER OF RECORDS TO ALLOCATE: 100

ENTER DATA SET TYPE: 0=USER; 1=DATA; 3=PROGRAM: 1

REQUESTED FUNCTION COMPLETED
```

CR—Change Remote Cell Number

This command ends the $.RMU utility in the remote cell and starts communication with a different remote cell.

```
SHUTDOWN COMPLETED SUCCESSFULLY

ENTER THE REMOTE CELL NUMBER: S2

CONTACTING $.RMU IN CELL S2
```

DE—Delete Data Set

This command deletes a data set on the remote cell.

```
ENTER REMOTE DATA SET NAME (NAME,VOLUME): A,EDX002

REQUESTED FUNCTION COMPLETED
```

DU—Dump Storage

This command dumps a partition to a data set on the remote cell.

```
ENTER REMOTE DATA SET NAME (NAME,VOLUME): DUMP,EDX003

ENTER PARTITION # TO DUMP: 2

REQUESTED FUNCTION COMPLETED
```

EN—Terminate Utility

This command stops $.HMU in the host cell and $.RMU in the remote cell.

EX—Execute Program

This command causes an EDX program to be executed in the remote cell.

```
WHICH PARTITION NUMBER?

    -1    RMU PARTITION
     0    ANY PARTITION
   1-8    SPECIFIC PARTITION

ENTER NUMBER: 2

DO YOU WANT THE LOAD MSG LOGGED? Y

DO YOU WANT TO WAIT FOR REMOTE PROGRAM? Y

ENTER PGM NAME (NAME,VOLUME): PROG3,EDX003

DO YOU WANT TO PASS ANY FREE SPACE AMOUNT? Y

ENTER FREE SPACE AMOUNT IN BYTES: 512

DO YOU WANT TO PASS PARAMETERS TO THE PGM? Y

ENTER NUMBER OF WORDS (MAX=40): 2

ENTER CHARACTER STRING: ABCD

DO YOU WANT TO PASS DATA SET NAMES? Y

ENTER NUMBER OF DATA SETS (MAX=9): 1

ENTER DATA SET NAME (NAME,VOLUME): WORK,EDX003

EXECUTE PROGRAM COMPLETED SUCCESSFULLY
```

ID—ID Check

This command verifies which remote cell you are communicating with.

`ENTER HOST ID:` **HMUb*xx***

where *xx* is the cell identifier of the cell where $.HMU is running and b represents a blank.

`REMOTE ID RECEIVED WAS:` **RMUb*yy***

where *yy* is the cell identifier of the cell where $.RMU is running and where b represents a blank.

PT—Pass-through

This command logically connects the EDX terminal used for $.HMU to the EDX system where $.RMU resides. This allows your terminal to become a terminal on the remote cell. Pass-through requires that the remote EDX system be defined properly. Refer to the chapter "Remote Management Utility" of the *EDX Communications Guide* for restrictions.

If you issue query type commands, such as $A ALL or CP Q NET, be sure that $.PD's transaction buffer is large enough for the all the data that will be returned in response to the command. This is necessary because the entire response is sent as a single transaction. (See "Planning Storage Requirements" in the *Design and Installation Guide.*)

Use the pass-through assist program when issuing CP commands during a pass-through session. Otherwise, a deadlock situation may occur, as described in the "Considerations for Using PASSTHRU" section of the chapter "Remote Management Utility" in the EDX *Communications Guide.* If you do not use the pass-through assist program and a deadlock occurs, you can use the HMUABORT command to terminate the pass-through session.

`ENTER COMMAND (?):` **PT**

`START PASSTHRU ASSIST PROGRAM (?):` **Y**

`>`

When your terminal is ready to accept input, you'll receive this message:

`A(TTN), R(EAD), Q(UIT)`

Enter A to simulate pressing the attention key.
Enter Q to terminate the pass-through session.
Enter R to display terminal output such as status information that you requested by issuing a CP command.

The pass-through assist program , $RMUPA, remains in storage until you terminate it.

You should terminate the pass-through assist program before you terminate the pass-through session, as follows:

`A(TTN), R(EAD), Q(UIT)` **A**

> $RMUPA

Then you may terminate the pass-through session:

```
A(TTN), R(EAD), Q(UIT) Q

PASSTHRU SESSION COMPLETED
```

The pass-through assist program is loaded into any available partition. If it is loaded into the $.CF partition, the remaining space may not be sufficient to execute CP commands.

To avoid this problem, you can load the pass-through assist program when the system is IPLed and leave it running until the system is shut down. Include the following instructions in your $INITIAL program:

```
         ENQT   RTERM
         LOAD   $RMUPA,LOGMSG=NO,PART=n
                 •
                 •
                 •
RTERM    IOCB   CDRVTB
```

where *n* is a partition number. If you also load $.CFS or $.CFD in $INITIAL, load $RMUPA first or specify different partitions for the two programs.

RM—Receive Data Set

This command allows the host cell to receive a data set from the remote cell.

Data set type may be DATA or PROGRAM.

```
HOST DATA SET (NAME,VOLUME): FILE,EDX003

REMOTE DATA SET (NAME,VOLUME): FILE1,EDX004

TRANSFER SOURCE DATA SET? (Y/N): N

START WITH RECORD 1 OF REMOTE MEMBER? (Y/N): Y

TRANSFER MEMBER:
FROM REMOTE FILE,EDX003 TO HOST FILE1,EDX004
CONTINUE? (Y/N): Y

RECEIVE FUNCTION COMPLETED
20 RECORDS RECEIVED
```

SD—Shut Down $.RMU

This command shuts down $.RMU in the remote cell without ending $.HMU in the host cell and optionally starts a program in the remote cell.

```
DO YOU WISH TO LOAD A PGM AT THE REMOTE? Y

WHICH PARTITION NUMBER?
    -1   RMU PARTITION
     0   ANY PARTITION
    1-8  SPECIFIC PARTITION
```

```
ENTER NUMBER: 2

DO YOU WANT THE LOAD MSG LOGGED ? Y

DO YOU WANT TO WAIT FOR REMOTE PROGRAM ? Y

ENTER PGM NAME (NAME,VOLUME): PROG3,EDX003

DO YOU WANT TO PASS ANY FREE SPACE AMOUNT ? Y

ENTER FREE SPACE AMOUNT IN BYTES: 512

DO YOU WANT TO PASS PARAMETERS TO THE PGM ? Y

ENTER NUMBER OF WORDS (MAX=40): 2

ENTER CHARACTER STRING: ABCD

DO YOU WANT TO PASS DATA SET NAMES ? Y

ENTER NUMBER OF DATA SETS (MAX=9): 1

ENTER DATA SET NAME (NAME,VOLUME): WORK,EDX003

SHUTDOWN COMPLETED SUCCESSFULLY
```

SM—Send Data Set

This command sends a data set from the host cell to the remote cell.

Data set type may be DATA or PROGRAM.

```
HOST DATA SET (NAME,VOLUME): FILE,EDX003

REMOTE DATA SET (NAME,VOLUME): FILE1,EDX004

TRANSFER SOURCE DATA SET ? (Y/N):4 N

START WITH RECORD 1 OF REMOTE MEMBER ? (Y/N):4 Y

TRANSFER MEMBER:
FROM HOST FILE,EDX003 TO REMOTE FILE1,EDX004
CONTINUE? (Y/N): Y

SEND FUNCTION COMPLETED
20 RECORDS TRANSMITTED
```

[4] If data set type is PROGRAM, you will not be prompted for this information. Processing will start with record 1 of the remote member.

WR—Wrap Message

This command sends a message from the host cell to the remote cell and $.RMU sends the same message back to the host. You can use this to verify that the communications link is active and operating.

```
ENTER WRAP TEXT
```

HELLO THERE

```
WRAP DATA RECEIVED WAS
HELLO THERE
```

> *HMUABORT—Stop Current Function*

This command immediately ends any function in progress. It has no effect if you enter it before establishing communication with the remote cell or when there is no function in progress. Some functions, especially those waiting for a response from your terminal, will not respond to HMUABORT until you have entered all the required parameters.

If you use HMUABORT to terminate an RM or SM function, the data set being transmitted may be left in an incomplete state. You should use the HMUSTOP command instead.

> *HMUSTOP—Stop RM or SM Function*

This command stops the data set transmission you requested by an RM or SM command. It has no effect if you enter it when an RM or SM function is not in progress.

$.PDBSTS—Volume Protection Utility Program

This utility program allows you to acquire exclusive control of a disk volume directory in a remote Series/1 before you load a program that may cause the directory to be updated. The program prevents a potential problem with remote disk support—two programs could enqueue on the same directory, one in the Series/1 where the volume actually exists and the other in a different Series/1 where the volume is defined as a pseudo volume.

Use the EDX $L command to load the program. Then enter one of the following commands:

- ?—Display valid commands
- DQ—DEQ volume
- EQ—ENQ for volume
- LD—Load a program
- EN—Terminate utility

A description and example of each command is given in the following sections.

DQ—Release Control of Remote Volume

This command releases exclusive control of a real disk volume in a remote cell that corresponds to the pseudo volume defined in your cell.

```
ENTER PSEUDO VOLUME NAME- PSU002

PSEUDO VOLUME PSU002 RELEASED
```

In this example, you no longer control the remote disk volume that corresponds to your volume, PSU002.

EN—End

This command terminates the utility program.

EQ—Acquire Control of Remote Volume

This command gives you exclusive control of a real disk volume in a remote cell that corresponds to the pseudo volume defined in your cell.

```
ENTER PSEUDO VOLUME NAME- PSU002

DO YOU WISH TO WAIT IF VOLUME IS NOT AVAILABLE (Y/N): Y

DO YOU WISH TO LOAD A PROGRAM? (Y/N): N
```

In this example, you now have control of the remote disk volume that corresponds to your volume, PSU002.

LD—Load Program

This command allows you to load a program.

```
PGM(NAME,VOLUME): $EDIT1N

ENTER # OF DATASETS FOR THIS PROGRAM- 1

DS (NAME,VOLUME): EDITWORK
```

In this example, the EDX utility program $EDIT1N would be loaded and executed.

$.SETSTG—Set $.CFD Storage Utility Program

$.SETSTG is a program that alters the size and/or allocation of dynamic storage for $.CFD, the disk-queuing version of the control program. This storage is used for disk-queue file control blocks (FCBs) and the message buffer pool (CFBUF). Use this utility only if your Communications Facility system has disk queuing of messages.

A file control block is required for each station defined and started with disk queuing active. You may need to increase the space reserved for file control blocks if your system configuration includes many such stations.

The disk-queuing version of the control program ($.CFD) contains 7680 bytes of dynamic storage, with 7168 bytes used for message queues and 512 bytes used for file control blocks. You can alter the total dynamic storage size and the space reserved for file control blocks. The remaining portion of the dynamic storage is used for CFBUF. Storage is allocated in increments of 256 bytes, based on your responses to $.SETSTG prompts.

$.SETSTG updates the $.CFD program header on disk; it does not affect processor storage. Therefore, your change will not take effect until you restart the Communications Facility. $.SETSTG and $.CFD must reside on the same disk volume.

$.SETSTG Example

```
> $L $.SETSTG

$.SETSTG      13P,10:48:14, LP= 4000

THE CURRENT SIZES OF THE FCB POOL AND CFBUF ARE:
     FCB =     512 (= SPACE FOR     10 FCBS)
   CFBUF =    7168
   TOTAL =    7680

IS TOTAL SIZE OK ? N

ENTER THE TOTAL BYTES OF STORAGE DESIRED: 9216

THE CURRENT SIZES OF THE FCB POOL AND CFBUF ARE:
     FCB =     512 (= SPACE FOR     10 FCBS)
   CFBUF =    8704
   TOTAL =    9216

IS TOTAL SIZE OK ? Y

IS THE NUMBER OF FCBS TO BE CHANGED? Y

ENTER THE MAX. NUMBER OF ENTRIES TO ALLOW
IN THE FCB POOL: 40

THE STORAGE RESERVED FOR FCBS WILL BE: 2048

THE REMAINING SPACE FOR CFBUF WILL BE: 7168

OK ? Y

MODIFICATION COMPLETED

$.SETSTG ENDED AT 10:55:34
```

$.UT1—Diagnostic Aid Utility

$.UT1, the diagnostic aid utility, has various functions to assist you in determining the source of a Communications Facility system problem. You load $.UT1 from an EDX terminal using the $L command. You can load and run $.UT1 even when the Communications Facility is not operational; however, some of the $.UT1 commands require that the Communications Facility be operational. It must be loaded into a mapped partition.

Since $.UT1 is a diagnostic tool, its storage requirements have been minimized. Thus, even though there is some overlap with other Communications Facility utilities, it may be preferable to use $.UT1 instead of some other utility during Communications Facility execution.

$.UT1 functions fall into four categories: station management, storage display, system facility display, and message management.

Station Management

$.UT1 has four station management commands:

- LA—List all stations by name
- LU—Display the starting storage address of a station's control block
- AL—Allocate a message station or an alias station
- DE—Delete a station

For example, if the Communications Facility were unexpectedly canceled, or if CFBUF had become saturated with messages, you could use the $.UT1 LA command (in lieu of the CP Q command, which would be unavailable) to get a list of active stations and determine whether there are messages waiting for them.

The DE command is useful when, for example, CFBUF has been filled with messages for a specific station, to the extent that all message traffic in the system stops. In this case, deleting the station purges all its messages from the system message pool and thus frees CFBUF.

When testing a Communications Facility application program, it may be useful to be able to allocate a message station to receive messages from the application. You would use the AL function, for example, if the application had caused the command processor to go into a busy condition, which could be terminated only by the application sending a message to some other station.

You can use the LU command to point you to a station control block for the purpose of further examining its contents.

Storage Display

Two $.UT1 commands allow you to display the contents of system storage areas:

- DB—Display the first 13 words of each element currently in the message buffer pool.

- DQ—Display the first 13 words of each element currently in the system storage pool.

If these two commands are issued from a 3101 terminal in block mode, use the $TERMUT1 utility to specify OVERFLOW LINES=Y. Otherwise the last character of each line will not be displayed.

During debugging of a Communications Facility application, you can use DQ to confirm that work space addresses contained in the application are correct, and that the workspace is in use by the correct task. You can use DB to confirm that a message sent by the application was actually stored in CFBUF.

Facility Management

The $.UT1 GA command allows you to get the addresses of various system control blocks and tables.

By using GA, you can get the addresses of these system facilities:

- Task control block
- Console control block
- System station pool control block
- System buffer pool control block
- System queue control block
- $SYSCOM
- Language extension command table

Having found the address of the language extension command table, you can locate five other system facilities by their negative displacements from the table:

-24	S$SMECB	Storage manager event control block
-18	SMRCB	Storage manager resource control block
-8	S$SYSQ	System station queue control block
-4	S$CFAKR	Address key of $.CF
-2	S$CFPOOL	Communications Facility message buffer

For example, if your application uses the common area of the supervisor ($SYSCOM), you can use the GA command to confirm that the data has been correctly moved to $SYSCOM.

Message Management

Four $.UT1 commands allow you to operate on Communications Facility messages:

- SE—Send a message to a station
- RE—Receive a message from a station
- DM—Display, on the current log device, a member of the $.SYSMSG data set
- CP—Send a CP command to a node

To use these commands, the Communications Facility must be operational. If, for example, you're debugging a program that sends messages, you might use $.UT1 to receive them. If you want to start or stop a station in a remote Series/1, you can use the utility to issue a CP command. You can also use the DM command in lieu of $.CONFIG to list, for example, all the command processor messages.

The rest of this chapter gives the formats of the $.UT1 commands. They appear in alphabetical order.

AL—Allocate Station

To allocate a message station in the system message pool, the command is:

AL *station-name*

station-name
: is the name of the message station to be allocated.

To allocate an alias station, the command is:

AL ALIAS *alias-name* FOR *station-name*

alias-name
: is the name of the alias station.

station-name
: is the name of the station to which the alias is to be assigned.

CP—Send a CP Command

When you enter the $.UT1 CP command, you'll be prompted for the CP command you want to send and for its destination. The CP command you enter can consist of up to 70 characters.

This command requires that the Communications Facility be operational.

DB—Display Buffer Pool

The DB command displays the message buffer pool in the $.CF partition. Each element in the pool is displayed; the display looks like this:

```
ADDR NIQ  PIQ  OWNR SIZE MNIQ MPIQ MPRI STCB MALV FLAG MOAF MDAF MSNF MDCF MRH
5708 5780 0000 5552 0070 0060 0007 571A 5721 6060 D740 D4E2 C7D3 E4C3 40D6 C6C6
5780 57A8 5708 0000 0020 0000 0000 7F03 4F54 0000 0000 28B4 0000 0001 0007 2000
57A8 57D0 5780 4F54 0020 0000 0000 7F03 4F54 0000 0000 28B4 285C 0000 0003 0000
57D0 57F8 57A8 4F54 0020 0000 57B0 7F03 4F54 0000 0000 0000 285C 0000 0004 0000
57F8 0000 57D0 0000 1D00 0000 0000 0007 7DD0 0000 0000 0000 0000 00C6 001C 1800
```

where:

ADDR
: is the address of the storage resource block of this element.

NIQ
: is the address of the next storage resource block in the queue. If this is the last or only element in the chain, NIQ is 0000.

PIQ
: is the address of the previous storage resource block in the queue. If this is the first or only element in the chain, PIQ is 0000.

OWNR
: is the address of the requestor's TCB if this space is in use. If it isn't in use, OWNR is 0000.

SIZE
is the size, in hexadecimal, of the buffer space that follows. To calculate the total amount of space required, add 8 (for the storage resource block) to SIZE.

MNIQ
is the address of the next message for this station. If this element is 0, this is the last message on the station's queue.

MPIQ
is the address of the previous message for this station. If this element is 0, this is the first message in the station's queue.

MPRI
is the priority of the message.

STCB
is the sender's task control block (TCB) address.

MALV
is the network address of the sender's alternate link; 0 if it has no alternate link.

FLAG
Three bits are of interest: bit 8 is 0 if the origin field (MOAF) is a network address, and 1 if it is the address of an EDX terminal control block; bit E (the seventh bit of the second byte) is 1 if the message was disk-queued, 0 if it wasn't; and bit F (the last bit) is 1 if the destination is $.WASTE, 0 otherwise.

MOAF
is the origin address, in the form indicated by bit 8 of M$FLAG. If that bit is 0, this field is the originating station's network address. If that bit is 1, this field is the address of the EDX terminal control block associated with the terminal from which a command was issued.

MDAF
is the destination station's network address; 0 if the destination was specified by linkage rather than expressly.

MSNF
is a binary number representing the sequence number of the message, assigned at its origin.

MDCF
is a binary number representing the number of bytes in the message.

MRH
is an indicator of the message type: bit 1=status message, bit 2=data or transaction message, bit 3=command message, bit 4=log message. Bits 6-7 are always 1, and the other bits are always 0.

DE—Delete Station

The DE command deletes the definition of a station from processor storage. $.SYSNET definitions are not affected. The format of the command is:

DE *station-name*

station-name
: is the name of the station to be deleted.

If you issue a DE command for an active station, you receive a prompt asking if you still want to delete it. If you answer yes, pending storage-queued messages for the station are purged and the station is deleted. Alternatively, you can use the RE command to receive the messages and then delete the station.

DM—Display Log Messages

This command displays, on the log device, one member of the $.SYSMSG data set. The command format is:

DM *id*

id
: is the ID of the member to be displayed.

All 99 messages are displayed. This command requires that the Communications Facility be operational.

DQ—Display System Pool

The DQ command displays the first 13 words of each element in the system storage pool, S$POOL. Each element in the pool is displayed; the display looks like this:

```
ADDR NIQ  PIQ  OWNR SIZE QNIQ QPIQ TYPE STAT QDLV QNAU QTCF NAME
2224 228C 0000 5526 0060 2294 0000 0000 8000 0000 0000 0808 5B4B C4C9 E2D7 4040
228C 22E4 2224 8DF4 0050 26FC 222C 0020 8000 0000 01DE 0808 5B4B D7C4 4040 4040
22E4 25EC 228C 5694 0300 0000 5B4B C8E2 D340 4040 0000 245E 24DE 0000 08CE 24E2
25EC 26F4 22E4 8DF4 0100 2804 27AC 0AC0 A012 0000 0129 0808 D7D6 D3D3 F2F9 4040
26F4 274C 25EC 8DF4 0050 2754 2294 0200 8000 0000 01DA 0808 5B4B C9D6 F0C1 C3F0
274C 27A4 26F4 8DF4 0050 27AC 26FC 04C0 8002 0000 01C0 0808 C3F3 F2F7 F1F2 F9E3
27A4 27FC 274C 8DF4 0050 25F4 2754 04C2 8022 01DE 01C1 0808 C3F3 F2F7 F7F2 F9E3
27FC 2854 27A4 5824 0050 0000 25F4 0C00 8000 0000 2804 0808 5B4B E6E2 C340 4040
2854 0000 27FC 0000 0EE0 0000 0000 0680 8000 0000 285C 0808 E3C5 D9D4 F4F9 F7F8
```

where:

ADDR
: is the address of the storage resource block of this element.

NIQ
: is the address of the next storage resource block in S$POOL. If this is the last or only element in the pool, NIQ is 0000.

PIQ
: is the address of the previous storage resource block in S$POOL. If this is the first or only element in the pool, PIQ is 0000.

OWNR
is the address of the requestor's TCB if this space is in use. If it isn't in use, OWNR is 0000.

SIZE
is the size, in hexadecimal, of the buffer space that follows. To calculate the total amount of space required, add 8 (for the storage resource block) to SIZE.

If the entry represents a station block, the remaining 11 fields are the first 11 words of that station block. If it represents user-acquired workspace, the 11 fields are the contents of the first 11 words of the work buffer for the user whose TCB address is displayed under OWNR.

GA—Get Address of System Facilities

The GA command allows you to get the addresses of various system control blocks and tables.

When you enter the GA command, $.UT1 displays a menu of facilities whose addresses you can request. You choose the facility you want by number:

0	The current task control block
1	The current console control block
4	System station pool control block
5	System buffer pool control block
6	System queue control block
7	$SYSCOM
8	Language extension command table

LA—List All Stations

The LA command lists all the stations in storage. For each station, this command lists its name, its network address, and the storage address of its station control block. If there are messages on the queue for a station, it is flagged with 'S' to indicate storage-queued messages pending, 'D' to indicate disk-queued messages pending, or 'B' to indicate both.

LU—Display Station Control Block Address

The LU command displays the storage address of a station's control block. Its format is:

LU *station-name*

station-name
is the name of the station whose control block address is to be displayed.

RE—Receive a Message

The RE command receives the next message on a station's queue. Its format is:

RE *station-name*

station-name
is the station from whose queue a message is to be received.

This command requires that the Communications Facility be operational.

SE—Send a Message

The SE command sends a text message to a station. The message can consist of up to 70 characters. Once you enter SE, you'll be prompted for the name of the destination station and the text of the message to be sent.

This command requires that the Communications Facility be operational.

$.UT2—Message Management Utility Program

Utility program $.UT2 allows you to manage message queues. You can obtain a report of messages in a disk queue, and you can purge messages from a disk or storage queue. When you purge messages, you can also obtain a report of them, and you can send each message on to the destination you specify.

Most of the functions of $.UT2 can be used with any station's message queue. A few of the functions can be used only with the undeliverable message queue, $.WASTE. Unless you specify otherwise, the message queue processed is that for $.WASTE.

Each undeliverable message in a $.WASTE queue is preceded by a reason message that includes the date, time, and a reason code that indicates why the accompanying data message could not be delivered.

The reason codes are listed in Figure 23.

Reason Code	Reason
WA01	Message sent to unknown station.
WA02	Message sent to null destination from unknown station.
WA03	Message sent to remote station and path to remote node unknown.
WA04	Message sent to null destination from station with unknown direct link vector.
WA05	Message sent to message dispatcher.
WA06	Message sent to inactive (stopped) station whose type is other than message.
WA07	Message sent directly to $.WASTE by SEND instruction using OPTION=WASTE.

Figure 23. Reason Codes

When you load $.UT2 with the EDX $L command, you must specify the name and volume of the system configuration data set, $.SYSNET. You must load the program into a mapped partition. Only one copy of the program is allowed in storage.

The $.UT2 commands are:

- ?—Display list of commands
- DQ—Select a disk queue
- PU—Purge messages
- RE—Report messages
- ST—Select a station
- EN—Terminate utility

Unless you are obtaining a message report on the terminal from which you loaded $.UT2, you can cancel a purge or report operation by entering:

> CA

A description of each command is given in the following sections.

DQ—Select a Disk Queue

You use this command to identify which message queue you want to process by specifying the name and volume of a disk-queue data set. The data set must not be assigned to an active station. You can use this command, for example, to process a disk-queue data set that is no longer assigned to a station.

EN—Terminate Utility Program

This command ends the utility program.

PU—Purge Messages

This command purges messages from either the message queue you specified with the DQ or ST command or the default message queue, $.WASTE.

After entering the PU command, you'll be prompted to specify:

- whether you want a report of the purged messages
- if you want any of the purged messages to be sent on
- which messages are to be purged.

There is an example of purging messages, obtaining a report of purged messages, and sending messages on in the "Purge, Report, and Send Messages Example" on page 302.

If you choose to have a report of purged messages, you'll be prompted for the name of the terminal on which the report is to be displayed. If you are going to send messages on, you should specify the terminal from which you loaded $.UT2 so that you can see each message before you specify its disposition. The report is described in the "RE—Report Messages" on page 301.

If you choose to send any messages on, you'll be prompted during the purge operation for the disposition of each message (whether or not you want it sent on). If you send it on, you'll be prompted for the name of the destination station. If you are purging messages from a $.WASTE queue, you'll be prompted separately for the disposition of each reason message and each accompanying undeliverable data message.

A message cannot be sent on if it is larger than $.UT2's buffer (the default size is 512 bytes). In that case, you can end $.UT2 and use the SS command of EDX utility program $DISKUT2 to increase the buffer size.

You specify which messages you want purged by selecting one of the following options:

AL
: purges all messages from a queue, both those in storage and on disk. The station to which the message queue belongs must be active. Use this option only to purge a queue whose messages are not received by some other program such as the undeliverable message queue, $.WASTE.

DA
purges disk-queued messages sent within the range of dates you specify. This option is effective only for $.WASTE queues—the date is not recorded in other message queues.

TI
purges disk-queued messages sent within the range of times you specify. The time range applies to all dates. This option is effective only for $.WASTE queues—the date is not recorded in other message queues.

DE
purges disk-queued messages sent to the station you specify. You identify the destination station by giving its network address, four hexadecimal digits.

OR
purges disk-queued messages sent by the station you specify. You identify the origin station by giving its network address, four hexadecimal digits.

EN
terminates the purge function.

Only disk-queued messages are purged for all options except AL. If the disk queue is assigned to an active station, the station is stopped during the purge operation and then restarted.

RE—Report Messages

This command produces a report of the messages in either the disk queue you specified with the DQ or ST command or the default disk queue, $.WASTE. There is an example of obtaining a report of messages in "Report Messages Example" on page 305.

The format of the report varies depending on whether or not the disk queue is (or was) assigned to $.WASTE. In the report of a $.WASTE queue, the display of each data message is preceded by a display of the accompanying reason message. In either case, the report includes the following for each data message:

- Origin station name
- Origin station type/subtype
- Destination station name
- Disk-queue data set relative record number
- Message header, displayed in hexadecimal (see the *Programmer's Guide* for a description of message headers)
- Message contents

The origin and destination of a message are recorded in the message header as network addresses. The first three items above are displayed only if a station block containing the network address is in storage. They identify the station to which the network address is currently assigned.

After entering an RE command, you'll be prompted to specify the terminal on which the report is to appear. You may specify any EDX terminal, including the one from which you loaded $.UT2.

You specify which messages you want reported by selecting one of the following options:

AL
: reports all messages.

DA
: reports the messages sent within the range of dates you specify. This option is effective only for $.WASTE queues—the date is not recorded in other message queues.

TI
: reports the messages sent within the range of times you specify. The time range applies to all dates. This option is effective only for $.WASTE queues—the date is not recorded in other message queues.

DE
: reports the messages sent to the station you specify. You identify the destination station by giving its network address, four hexadecimal digits.

OR
: reports the messages sent by the station you specify. You identify the origin station by giving its network address, four hexadecimal digits.

EN
: terminates the report function.

ST—Select a Station

You use this command to identify which message queue you want to process by specifying the name of a station. The default station is $.WASTE. You may specify LU instead of ST.

Purge, Report, and Send Messages Example

This example shows you how to purge messages within a range of dates from a disk queue identified by its name and volume. It shows you how to obtain a report of purged messages and how to send messages on. A sample report is shown in Figure 24 on page 304. The sample report is for $.WASTE's disk queue. Notice that the report lists reason messages as well as data messages.

```
>$L $.UT2 $.SYSNET,EDX002

$.UT2         32P,09:23:38, LP= 2800

USING STATION NAME $.WASTE

COMMAND (?): DQ

ENTER DATASET (NAME,VOLUME): WASTFILE,EDX002

USING DISK QUEUE WASTFILE,EDX002

COMMAND (?): PU
```

```
REPORT MESSAGES? Y

ENTER TERMINAL NAME (CR OR * = THIS ONE): $SYSPRTR

SEND ANY MESSAGES ON? Y

PURGE OPTION (?): ?
AL-ALL
DA-DATE (VALID FOR $.WASTE ONLY)
TI-TIME (VALID FOR $.WASTE ONLY)
DE-DESTINATION
OR-ORIGIN
EN-END

PURGE OPTION (?): DA

START (MM/DD/YY): 12/16/81

END (MM/DD/YY): 12/20/81

SEND MESSAGE ON? Y

ENTER DESTINATION NAME: PROGA

MESSAGE SENT TO PROGA
SEND MESSAGE ON? N

SEND MESSAGE ON? N

SEND MESSAGE ON? N

PURGE OPTION (?): EN

USING DISK QUEUE WASTFILE,EDX002

COMMAND (?): EN

$.UT2    ENDED AT 09:43:58
```

```
                        M E S S A G E   R E P O R T    12/20/81

    DISK QUEUE WASTFILE,EDX002   PREVIOUS STATION NAME $.WASTE

    PURGING BY DATE   12/16/81    12/20/81

  DATE TIME REASON: 12/16/81 09:09:33 WA06
    ORIGIN     TYPE     DESTINATION                  RECORD          3
               0000     HOST
  MESSAGE HEADER:  0000 0004 0105 3754 0000 0000 0000 0229 0000 0033 0000 0000
  MESSAGE CONTENTS:
MESSAGE SENT TO STATION HOST WHICH IS INACTIVE

  DATE TIME REASON: 12/16/81 09:18:02 WA05
    ORIGIN     TYPE     DESTINATION                  RECORD          5
               0000
  MESSAGE HEADER:  0000 0006 0105 3754 0000 0000 0000 0000 0000 001A 0000 0000
  MESSAGE CONTENTS:
MESSAGE SENT TO DISPATCHER


   E N D   O F   R E P O R T
```

Figure 24. $.WASTE Message Report

Report Messages Example

This example shows you how to report all messages in a disk queue identified by its station name. A sample report is shown in Figure 25 on page 306.

```
>$L $.UT2 $.SYSNET,EDX002

$.UT2         32P,09:53:22, LP= 2800

USING STATION NAME $.WASTE

COMMAND (?): ?

PU-PURGE MESSAGES
RE-REPORT MESSAGES
ST-SELECT STATION
DQ-SELECT DISK QUEUE
EN-END

COMMAND (?): ST

ENTER STATION NAME (CR = $.WASTE): MSGLU

USING STATION NAME MSGLU

COMMAND (?): RE

ENTER TERMINAL NAME (CR OR * = THIS ONE): $SYSPRTR

REPORT OPTION (?): ?

AL-ALL
DA-DATE (VALID FOR $.WASTE ONLY)
TI-TIME (VALID FOR $.WASTE ONLY)
DE-DESTINATION
OR-ORIGIN
EN-END

REPORT OPTION (?): AL

REPORT OPTION (?): EN

USING STATION NAME MSGLU

COMMAND (?): EN

$.UT2    ENDED AT 09:55:04
```

```
                    M E S S A G E   R E P O R T    12/20/81

      STATION NAME MSGLU   DISK QUEUE MSGFILE

    REPORTING ALL MESSAGES

    ORIGIN     TYPE     DESTINATION                        RECORD          2
               0000     MSGLU
  MESSAGE HEADER:  0000 0003 0005 3754 0000 0000 0000 02FF 0000 0011 0000 0000
  MESSAGE CONTENTS:
CP S HOST

    ORIGIN     TYPE     DESTINATION                        RECORD          3
               0000     MSGLU
  MESSAGE HEADER:  0000 0004 0105 3754 0000 0000 0000 02FF 0000 0013 0000 0000
  MESSAGE CONTENTS:
CP SET LOG $SYSPRTR

    ORIGIN     TYPE     DESTINATION                        RECORD          4
               0000     MSGLU
  MESSAGE HEADER:  0000 0005 0105 3754 0000 0000 0000 02FF 0000 0004 0000 0000
  MESSAGE CONTENTS:
CP Q

    ORIGIN     TYPE     DESTINATION                        RECORD          5
               0000     MSGLU
  MESSAGE HEADER:  0000 0006 0105 3754 0000 0000 0000 02FF 0000 000F 0000 0000
  MESSAGE CONTENTS:
CP P HOST MSGLU

   E N D   O F   R E P O R T
```

Figure 25. Message Report

Appendix A. 3270 Polling and Selection Address Table

Use this table to determine the polling and selection addresses for a real or emulated 3270 terminal station, according to the terminal's control unit number and device number.

Each control unit has a number from 0 to 31. Each device (display station or printer) is attached to a control unit and has a number from 0 to 31 that represents its position on the control unit. For real 3270 terminals, these numbers are assigned when the terminals are installed. For emulated 3270 terminals, the numbers are defined by the host program with which the Series/1 communicates.

Each address is a 4-digit hexadecimal value. The first two digits are referred to as CU (control unit) and the last two as DA (device address).

You can use $.CONFIG to define your terminal stations. $.CONFIG generates the correct EMULATION CUDAs given the control unit device address. When you are prompted for the poll or selection CUDA, request guidance from $.CONFIG. You must enter the CUDAs directly if the terminal station controlling line has a type of 3271C.

The polling address for a control unit is the value from column 2 for the corresponding control unit number (column 1), followed by 7F.

The selection address for a control unit is 0000.

The polling address for a device is the value from column 2 for the corresponding control unit number (column 1), followed by the value from column 2 for the corresponding device number (column 1).

The selection address for a device is the value from column 3 for the corresponding control unit number, followed by the value from column 2 for the corresponding device number.

For example, the polling and selection addresses for control unit 5 and its devices 0, 1, and 2 are:

Station	Polling Address	Selection Address
Control Unit 5	C57F	0000
Device 0 on CU5	C540	E540
Device 1 on CU5	C5C1	E5C1
Device 2 on CU5	C5C2	E5C2

COLUMN 1 Control Unit or Device Number	COLUMN 2 Polling CU Polling DA Selection DA	COLUMN 3 Selection CU
0	40	60
1	C1	61(E1)[5]
2	C2	E2
3	C3	E3
4	C4	E4
5	C5	E5
6	C6	E6
7	C7	E7
8	C8	E8
9	C9	E9
10	4A	6A
11	4B	6B
12	4C	6C
13	4D	6D
14	4E	6E
15	4F	6F
16	50	F0(70)[6]
17	D1	F1
18	D2	F2
19	D3	F3
20	D4	F4
21	D5	F5
22	D6	F6
23	D7	F7
24	D8	F8
25	D9	F9
26	5A	7A
27	5B	7B
28	5C	7C
29	5D	7D
30	5E	7E
31	5F	7F

5 For 3270 emulation, the Series/1 communication hardware does not honor selection CU values 61 and F0. Instead, use the values E1 and 70 in your station definitions and in host system definitions.

6 The strapping for remote IPL on a BSC adapter card is also the high-order bit of the polling address. With IPL enabled, polling addresses must begin with 'C' or 'D', and selection addresses with 'E', or 'F'.

Glossary

This is a glossary of technical Communications Facility terms that appear in the book. Only terms unique to the Communications Facility are defined here. For definitions of Event Driven Executive terms, see the appropriate EDX book. For definitions of 3270 terms, see the *3270 Component Description* manual. For definitions of X.25/HDLC Communications Support terms, see the *X.25/HDLC Communications Support Programming and Operating Reference Manual.*For general data processing terms see *Vocabulary for Data Processing, Telecommunications, and Office Systems,* GC20-1699.

?. The $.CONFIG, $.HMU, $.PDBSTS, or $.UT2 command that displays a list of the utility's commands.

<IO>. The transaction identifier associated with the Communications Facility program $.PD<IO>.

> HMUABORT. The $.HMU command that ends any $.HMU functions in progress.

> HMUSTOP. The $.HMU command that stops the data set transmission requested by an RM or SM command.

$.CF. The name given to the control program after it is loaded under the name $.CFD or $.CFS.

$.CFD. The version of the control program for Communications Facility systems that use disk queuing of messages.

$.CFMENU. A sample application program, distributed as part of the Communications Facility, that demonstrates how to communicate with users at 3270-type terminals.

$.CFS. The version of the control program for Communications Facility systems using storage queuing of messages.

$.CONFIG. The utility program that allows the user to define and modify stations and maintain the system message data set.

$.DISP. The Communications Facility system station.

$.DSINIT. The utility program that initializes a data set for the disk queuing of messages.

$.HMU. The transaction-processing program that allows an operator at a Series/1 to maintain a remote Series/1.

$.IO0AB0. The input/output control program that manages communication between Series/1s attached to a Local Communications Controller.

$.IO0AB8. The input/output control program that manages X.25 packet level communication between a Series/1 and a DTE connected by an HDLC line, with or without an intervening X.25 packet-switching data network.

$.IO0AC0. The input/output control program that manages 3270 display stations and printers attached to the Series/1 by a multipoint BSC line.

$.IO0AD0. The input/output control program that provides 3270 emulation when a Series/1 is connected to a host processor through a channel attachment.

$.IO0AE0. The input/output control program that provides 3270 emulation when a Series/1 is connected to a host processor by a multipoint BSC line.

$.IO0A10. The input/output control program that manages communication between Series/1s connected by a point-to-point BSC line.

$.IO0A20. The input/output control program that manages communication between a Series/1 and various other systems connected by a point-to-point BSC line.

$.IO06F0. (1) The input/output control program that manages 3101 display stations attached to the Series/1 through a feature programmable communications adapter or a multipoint attachment as if they were 3277 display stations. (2) The renamed $.IO0680 I/O control program.

$.IO0630. The input/output control program that manages 3101 or 7485 display stations attached to the Series/1 through a teletypewriter adapter as if they were 3277 display stations.

$.IO0670. (1) The input/output control program that manages 4978 display stations as if they were 3277 display stations and Series/1 printers as if they were 3286 printers or 3287 printers in SCS mode. (2) The renamed $.IO0678 I/O control program.

$.IO0674. The input/output control program that manages 4973, 4974, and 4975 printers as if they were 3286 printers.

$.IO0678. The input/output control program that manages 4978 display stations with RPQ D02428 and 4980 display stations as if they were 3277 display stations.

$.IO0680. The input/output control program that manages 7485 display stations attached to the Series/1 through a feature programmable communications adapter or a multipoint attachment as if they were 3277 display stations.

$.IO14E8. The input/output control program that provides 3270 emulation when a Series/1 is connected to a host processor that uses SNA.

$.PANEL. An interactive program for creating panels to be displayed at a 3270-type terminal.

$.PD. The program dispatcher; the program that manages the processing of transactions.

$.PDBSTS. The utility program that allows a user to gain exclusive control of a disk volume directory in a remote Series/1 before loading a program that will update that directory.

$.PDIPL. The transaction-processing program that sends an IPL bootstrap to a remote Series/1, causing that Series/1 to IPL itself.

$.PDSCHD. The transaction-processing program that schedules another transaction for processing at a specific time.

$.PD<IO>. The transaction-processing program that performs disk I/O in response to remote disk requests.

$.PD>IO<. The station used for remote disk support.

$.PNLUT1. The utility program that prints the descriptions of panels created by the $.PANEL program.

$.RMU. The transaction-processing program that processes transactions received from the host management utility operating in another Series/1.

$.SETSTG. The utility program that changes the size of the message buffer pool and file control block pool for $.CFD.

$.SYSIPL. The data set containing CP commands to be executed when the Communications Facility is loaded.

$.SYSLCC. The data set containing IPL transactions, used to IPL remote nodes on a Local Communications Controller ring.

$.SYSMSG. The data set containing the text of error messages and informational messages issued by the Communications Facility and, optionally, by user programs.

$.SYSNET. The data set containing the definitions of all the stations in a node and of remote stations that will be communicated with from that node.

$.SYSPD (program dispatcher data set). The data set containing CP commands, path definitions, transaction definitions, and transactions that are to be processed when the program dispatcher is started.

$.SYSPNL. The data set containing panels displayed by the $.PANEL, $.CFMENU, and $.IO014E8 programs.

$.SYSPRT. The data set containing image and control store loads for 4978 terminals and members used to support printers as buffered devices.

$.SYSX25. The data set containing two-digit call IDs and their associated X.25 network addresses used during call establishment for switched virtual circuits. As shipped, it contains 10 records of /*.

$.UT1. The utility program that allows access to various Communications Facility functions for diagnostic purposes.

$.UT2. The utility program that allows the user to examine and purge messages on the $.WASTE queue or any other disk queue.

$.WASTE. The station to which undeliverable messages are sent.

$.WSC. The work session controller; the part of the Communications Facility that allows an application program to communicate with multiple EDX devices attached to any Series/1 in the network.

$.WSCIMG. The data set containing images that can be displayed through the work session controller and members used to save data for transaction-processing programs.

$INITIAL. The program that is loaded when EDX is IPLed.

AL. (1) The $.HMU command that allocates a data set. (2)The $.UT1 command that allocates a message station.

alias station. A station that defines an alternate name for another station in a Communications Facility configuration.

alphameric mode. An attribute of a 4978 device station that allows the user to enter any characters in a numeric field. Contrast with *numeric mode*.

alternate link vector. The network address of a station that is the alternate destination for messages sent to a particular station. Undeliverable transactions and X.25 control messages are sent to a station's alternate destination.

APL mode. An attribute of a 4978 device station that supports use of an APL keyboard.

ASSIST. The $.CONFIG command that displays or suppresses extra prompts.

basic mode. An attribute of a station that causes messages sent to or received from it to be transferred without addition or removal of control characters. Contrast with *record mode.*

BCUG (bilateral closed user group). The X.25 facility that allows two DTEs to establish communication with each other by simply specifying a BCUG ID. Additional benefits vary with the network provider.

between brackets (BETB) state. The state of a logical unit station that allows messages to be sent from host to terminal, or from terminal to host.

bilateral closed user group (BCUG). The X.25 facility that allows two DTEs to establish communication with each other by simply specifying a BCUG ID. Additional benefits vary with the network provider.

buffer, Communications Facility. A storage area, from 1 to 32K bytes long, preceded by a buffer header.

buffered device. A printer supported in a way that allows data to be retained and merged with variable data on write operations.

C. The PD command that changes a cell identifier.

capacity warning level. The percentage of a disk-queue data set's capacity which, if reached, causes a warning message to be issued.

call ID. A user-defined two-digit number that represents an X.25 network address. Call IDs are used when defining a switched virtual circuit station if calls are to be sent to or received from a specific X.25 network address. Call IDs are associated with X.25 addresses in the $.SYSX25 data set.

cause code. A 1-byte code in a restart, clear, or reset packet that indicates the reason for the restart, clear, or reset. The X.25 I/O control program includes this code in log and control messages that it sends as a result of receiving them from XHCS.

CCITT. International Telephone and Telegraph Consultative Committee. An organization of common carriers and PTTs whose main goal is to recommend standards that facilitate interconnection of communications equipment.

cell. A node in the Communications Facility configuration in which the program dispatcher runs or a non-Series/1 host system where transactions are processed.

CELL. The statement in $.SYSPD that identifies the local cell.

cell identifier. A 2-character name that uniquely identifies a cell.

CFBUF. The message buffer pool; a workspace pool in the Communications Facility control program that contains storage-queued messages.

CHANGE. The $.CONFIG command that changes an existing station definition.

circuit station. A station that represents an X.25 virtual circuit. See also *virtual circuit.*

closed user group (CUG). The X.25 facility that defines a group of DTEs that can communicate with each other. The number of DTEs that can be in a group and additional benefits vary with the network provider.

command message. A message, the content of which is a CP command.

command processor. A part of the Communications Facility that processes CP and, with the program dispatcher, PD commands.

command processor (CP) commands. A set of Communications Facility commands used to define and control the Communications Facility configuration and display information about it.

command-processing program. A program that processes a particular CP command. The command processor controls loading and execution of command-processing programs as it receives commands.

Communications Facility buffer. A storage area, from 1 to 32K bytes long preceded by a buffer header.

Communications Facility configuration. A complete set of nodes, cells, and stations that communicate with one another through the Communications Facility.

Communications Facility terminal. A device defined to the Communications Facility, controlled by an I/O control program, and accessed from a program through SEND and RECEIVE instructions.

configuration processor ($.CONFIG). A utility program used to define and modify stations and maintain the system message data set.

control message. A message defined by the Communications Facility that contains information related to controlling the X.25 network. Circuit stations with a usage type of STD+ and the X.25 I/O control program can send and receive these messages using the SEND S and SEND SM instructions. The length of the control message varies depending on which type it is. See also the individual control message.

control program ($.CF). The part of the Communications Facility that includes the message dispatcher, the command processor, and the log processor.

COPY. The $.CONFIG command that creates a station definition from an existing station.

CP. (1) The PD command that sends a CP command to a cell. (2) The $.UT1 command that sends a CP command.

CP commands. A set of Communications Facility commands used to define and control the Communications Facility configuration and display information about it.

CPRSTART. The command that restarts the Communications Facility.

CR. The $.HMU command that changes a remote cell number.

CUG (closed user group). The X.25 facility that defines a group of DTEs that can communicate with each other. The number of DTEs that can be in a group and additional benefits vary with the network provider.

data circuit-terminating equipment (DCE). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection, including the signal conversion and coding between the data terminal equipment (DTE) and the line. In the &cfl, software functions provide a connection point for devices capable of interfacing to a packet-switching data network as DTEs. The Communications Facility does not provide all the DCE support defined by Recommendation X.25; it cannot be an X.25 network.

data message. A message, the content of which is user data to be sent from one station to another.

data stream RPQ. An RPQ (D02428) that improves performance when a 4978 display station is managed as if it were a 3277 display station.

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both, and provides for the data communication control function according to protocols. In the Communications Facility, DTE is hardware or software that is capable of attaching to an X.25 packet-switching network.

DB. The $.UT1 command that displays the message buffer pool.

DCE (data circuit-terminating equipment). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection, including the signal conversion and coding between the data terminal equipment (DTE) and the line. In the Communications Facility, software functions provide a connection point for devices capable of interfacing to a packet-switching data network as DTEs. The Communications Facility does not provide all the DCE support defined by Recommendation X.25; it cannot be an X.25 network.

DDM (device descriptor module). An XHCS module that defines an HDLC line. The name of the Communications Facility line station representing the line must have the same name as its DDM.

DDSK-30. The 30 megabyte disk within the 4952, 4954, or 4956 processor or within the 4965 storage and I/O expansion unit. Because DDSK-30 is the keyword on the EDX DISK statement, EDX and Communications Facility books use it to identify the disk unit.

DDSK-60. The 60 megabyte disk within the 4952, 4954, or 4956 processor or within the 4965 storage and I/O expansion unit. Because DDSK-60 is the keyword on the EDX DISK statement, EDX and Communications Facility books use it to identify the disk unit.

DCE (data circuit=terminating equipment). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection, including the signal conversion and coding between the data terminal equipment (DTE) and the line. In the Communications Facility, software functions provide a connection point for devices capable of interfacing to a packet-switching network as DTEs. The Communications Facility does not provide all the DCE support defined by Recommendation X.25; it cannot be an X.25 network.

DE. (1) The $.HMU command that deletes a data set. (2) The $.UT1 command that deletes a station definition.

deadband. A percentage of the capacity of a disk-queue data set. Once the usage of the data set has exceeded its capacity warning level, no further warning messages are issued until usage of the data set falls below the capacity warning level by the amount of the deadband and then reaches the capacity warning level again.

DEF. The CP command that defines a new station.

DEFINE. The $.CONFIG command that defines a station.

device descriptor module (DDM). An XHCS module that defines an HDLC line. The name of the Communications Facility line station representing the line must have the same name as its DDM.

device station. A Series/1 terminal or printer to be managed as if it were a 3270 device in the Communications Facility configuration.

device type. The combination of station type and station subtype for a Series/1 terminal or printer being managed as if it were a 3270 device. Device type indicates which input/output control program is to control a particular device.

diagnostic aid utility ($.UT1). The utility program that allows access to various Communications Facility functions for diagnostic purposes.

direct link vector. The network address of a station that is the default destination for messages sent by a particular station.

disk queue. A message queue on disk, used to hold low-priority messages destined for a particular station until the station is ready to receive them.

disk-queue file control block. A control block that contains information about a station that has a disk queue.

disk-queue data set. A data set used to hold low-priority messages destined for a particular station until the station is ready to receive them.

disk-queue data set initialization utility ($.DSINIT). The utility program that initializes a data set for the disk queuing of messages.

dispatcher, message. The part of the Communications Facility that determines the final destination of a message and routes it through the system to that destination.

dispatcher, program ($.PD). The part of the Communications Facility that manages the processing of transactions.

DISPLAY. The $.CONFIG command that displays a station definition.

DM. The $.UT1 command that displays a member of the $.SYSMSG data set.

DQ. (1) The $.PDBSTS command that releases exclusive control of a real disk volume that corresponds to a pseudo volume in another cell. (2) The $.UT1 command that displays the system storage pool. (3) The $.UT2 command that specifies which message queue is to be processed.

DTE (data terminal equipment). That part of a data station that serves as a data source, data sink, or both, and provides for the data communication control function according to protocols. In the Communications Facility, DTE is hardware or software that is capable of attaching to an X.25 packet-switching network.

DU. The $.HMU command that dumps storage to a disk

dynamic program dispatcher. See *program dispatcher.*

EDIT. The $.CONFIG command that stores or alters message text in $.SYSMSG.

EDX terminal. A terminal defined to the EDX operating system and used to perform EDX system functions.

emulation, 3270. The facility that allows a host processor to communicate with a Series/1 as if it were communicating with a 3270 system; also, the I/O control program that provides 3270 emulation over a BSC line ($.IOOAEO).

EN. (1) The $.HMU command that ends execution of $.HMU and $.RMU. (2) The $.PDBSTS command that terminates $.PDBSTS. (3) The $.UT2 command that terminates $.UT2.

END. The $.CONFIG command that terminates $.CONFIG.

error message. A message sent to the Communications Facility system log to indicate that an error has occurred.

EQ. The $.PDBSTS command that gains exclusive control of a real disk volume in a remote cell that corresponds to a pseudo volume in the local cell.

EQU. A statement in $.SYSPD that defines a transaction that has attributes similar to another transaction.

EX. The $.HMU command that executes a program at a remote site.

F. (1) The CP command that modifies the attributes of an existing station. (2) The PD command that modifies an entry in the path table or the transaction identifier table.

facilities. A set of optional characteristics and capabilities available from the network provider to switched virtual circuits during call establishment. In the Communications Facility, these facilities may be included in the SVC circuit station definition and/or in a call request control message from an application program. See also the individual facility.

fast select (FS). The X.25 facility that allows data to be appended to a call request, call clear, or call accept packet. Contrast with *fast select restricted.*

fast select restricted (FSR). The X.25 facility that allows data to be appended to a call request or call clear packet. Call accept packets are not allowed. Contrast with *fast select.*

FCB (file control block). A control block that contains information about a station that has a disk queue.

FILE. The CP command that displays disk-queuing parameters and changes or assigns a station's data set name for disk-queuing.

file control block (FCB). A control block that describes a queue of messages on disk.

FS (fast select). The X.25 facility that allows data to be appended to a call request, call clear, or call accept packet. Contrast with *fast select restricted.*

FSR (fast select restricted). The X.25 facility that allows data to be appended to a call request or call clear packet. Call accept packets are not allowed. Contrast with *fast select.*

GA. The $.UT1 command that gets the addresses of system control blocks and tables.

GOTEST. The PD command that allows execution of a program whose transaction identifier is in test mode.

H. (1) The CP command that removes a station from the active network. (2) The PD command that lists all the PD commands and their functions.

halted station. A station whose station block has been deleted from S$POOL.

HDLC (high-level data link control). The group of standards defining the link level for communications with a public data network. As defined by Recommendation X.25, LAPB conforms to one subset of standards and SDLC normal response mode conforms to a different subset. The Communications Facility adheres only to the LAPB subset.

header, message. 24 bytes at the beginning of a message that contain such information as its origin, destination, and priority.

HELP. (1) The CP command that lists all the CP commands and their functions. (2) The $.CONFIG command that displays the $.CONFIG commands.

high-speed loader ($.HSL). A small, efficient loader used by the program dispatcher to load transaction-processing programs.

HMU. The transaction identifier associated with the Communications Facility program $.HMU.

host management utility ($.HMU). The transaction-processing program that allows an operator at a Series/1 to maintain a remote Series/1.

host processor. A computer in a Communications Facility configuration where control functions are performed; it may be a Series/1 or another type of computer.

I. The PD command that inserts an entry into the path table or the transaction table.

ID. (1) The PD command that checks whether a cell is active. (2) The $.HMU command that checks a remote cell ID.

image library management utility ($.WSCUT1). The utility program that converts a screen image that was created by the EDX $IMAGE program and stores it in $.WSCIMG for use by the work session controller. It is also used to display the image.

incoming call control message. The Communications Facility message sent be the X.25 I/O control program to a STD+ circuit station to indicate that a remote DTE wishes to begin communications.

initialization data set ($.SYSIPL). A data set containing CP commands that is read when the Communications Facility is loaded.

input character count. The number of characters sent by a station.

input hold. A condition of a station in which all messages it sends are discarded.

input message sequence number. The number of messages sent by a station.

input/output control program (IOCP). A program that handles transmission of messages to and from a particular type of device or line in a Communications Facility configuration.

IOCP (input/output control program). A program that handles transmission of messages to and from a particular type of device or line in a Communications Facility configuration.

IPL. The transaction identifier associated with the Communications Facility program $.PDIPL.

IPL transaction data set ($.SYSLCC). The data set that contains IPL transactions, used to IPL remote nodes on a Local Communications Controller ring.

LA. The $.UT1 command that lists all stations.

LD. The $.PDBSTS command that loads a program.

line station. A station that represents a telecommunication line in a Communications Facility configuration.

line type. The combination of station type and station subtype for a communication line. Line type indicates which input/output control program (such as point-to-point or 3270 control) is to control a particular line.

LINK. (1) The CP or $.CONFIG command that defines a connection between two stations—either a direct link vector or an alternate link vector. (2) The work session controller high-level language subroutine that enables an application program to complete its own execution by loading and executing some other application program.

linked station. A station that has a single specified station as the default destination of messages it sends.

LIST. The $.CONFIG command that displays the contents of $.SYSMSG and $.SYSNET.

local node. The node from which the Communications Facility configuration is being viewed.

local station. A station that exists at the local node.

log message. A message that is sent to the Communications Facility system log.

log processor. The part of the Communications Facility that formats error and informational messages and sends them to the system log.

logical channel. A logical conduit for packets on a DTE-to-DCE link. Logical channels are assigned to virtual circuits dynamically (SVCs) or statically (PVCs). All packets on the circuit flow through the logical channel. See also *logical channel identifier.*

logical channel identifier (LCI). A number assigned to a logical channel to uniquely identify the channel and all packets flowing through it. In the Communications Facility, permanent virtual circuits have LCIs in their station definitions to assign them to that particular logical channel.

logical unit (LU) station. A station that represents an SNA logical unit (a terminal or a printer) in a Communications Facility configuration.

LU. The $.UT1 command that displays the storage address of a station's control block.

LU (logical unit) station. A station that represents an SNA logical unit (a terminal or a printer) in a Communications Facility configuration.

M. The PD command that sends a message to an EDX terminal.

mapped partition. A partition that contains the common area (system tables and station blocks).

MENU. (1) The transaction identifier associated with the Communications Facility program $.WSMENU. (2) The work session controller high-level language subroutine that enables the application program to end its own operation and return control to the primary application load program.

message. A unit of data to be transmitted from one station to another.

message buffer pool (CFBUF). A workspace pool in the Communications Facility control program that contains storage-queued messages.

message data set ($.SYSMSG). The data set containing the text of error messages and informational messages issued by the Communications Facility and, optionally, by user programs.

message dispatcher. The part of the Communications Facility that determines the final destination of a message and routes it through the system to the destination.

message header. 24 bytes at the beginning of a message that contain such information as the origin, destination, and priority of the message.

message priority. An attribute of a message that determines where it is placed in the destination station's message queue.

message queue. A queue of messages destined for a single station, either in processor storage or on disk.

message sequence number. A number associated with a message representing its sequence with respect to its origin.

message station. A queue of messages, not associated with a Communications Facility program or device.

message type. An attribute of a message that indicates whether it is a data, command, log, transaction or status message.

multinode mode. A mode of operation of the message dispatcher that allows routing of messages to stations in remote nodes.

name, station. A 1-to 8-character alphameric value that uniquely identifies each station in a node.

NAU. See *network address.*

network address. A 4-character hexadecimal value that uniquely identifies a station in the network. The first two characters are the node assignment, and the last two are the station address.

network configuration data set ($.SYSNET). The data set containing the definitions of all the stations in a node and of remote stations that will be communicated with from that node.

node. A Series/1 in the Communications Facility configuration.

node assignment. The first 2 characters of a station's network address; they uniquely identify a node in the network.

node station. A station that represents a remote node in a Communications Facility configuration.

non-display mode. An attribute of a terminal or device station that causes lowercase data to be converted to uppercase. For 4978 device stations, it also causes non-display output fields to be converted to lowercase, which displays as blanks. Contrast with *text mode.*

non-remove mode. An attribute of an I/O control program that allows the sending of duplicate messages caused by a terminal user pressing RESET and ENTER. Contrast with *remove mode.*

nontransparent mode. A mode of BSC transmission that prohibits bit patterns with a value less than X'40' from being transmitted as data.

numeric mode. An attribute of a 4978 device station that allows the user to enter only digits, decimal points, and minus signs in a numeric field. Contrast with *alphameric mode.*

output character count. The number of characters received by a station.

output hold. A condition of a station in which messages can be sent to it, but it can't receive messages from its queue.

output message sequence number. The number of messages received by a station.

P. (1) The CP command that stops a station. (2) The PD command that deactivates a path or transaction.

packet. The basic transmission unit on a data link accessing an X.25 network. See also *packet size.*

packet size. The size of the largest data packet sent to an X.25 network. The packet size is defined in the circuit or DxE line station definitions.

packet switching. The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet.

packet-switching data network (PSDN). A communications network that uses the mechanism of packet switching to transmit data. See also *packet switching.*

panel. A screen image for a 3270 display station or a Series/1 device being managed as a 3270 display station.

panel data set ($.SYSPNL). The data set containing panels displayed by the $.PANEL and $.CFMENU programs.

panel design aid ($.PANEL). An interactive program for creating panels to be displayed at a 3270-type terminal.

panel print utility ($.PNLUT1). The utility program that prints the descriptions of panels created by the $.PANEL program.

path. The route or linkage used to get a transaction from the program dispatcher in one cell to the program dispatcher in another cell.

PATH. A statement in $.SYSPD that defines a path.

path definition table. A table used by the program dispatcher that defines the paths to remote cells.

path table. Synonymous with *path definition table.*

PD commands. A subset of Communications Facility CP commands used to control the operation of the program dispatcher.

permanent virtual circuit (PVC). A permanent virtual connection that provides services similar to a leased line. Data sent to the network through a logical channel being used as a PVC is always delivered to a specific logical channel at a specific DTE destination in the network. In the Communications Facility, PVC circuit stations represent permanent virtual circuits.

physical unit (PU) station. A station that represents an SNA physical unit in a Communications Facility configuration.

preferred path. The path used to route transactions for unknown cells. See *PATH*.

primary cell identifier. The field of a transaction that contains the identifier of the cell in which the transaction is to be processed.

primary mode. An attribute of a point-to-point line station that defines the local node as the primary Series/1. Contrast with *secondary mode*.

primary transaction identifier. The field of a transaction that contains the transaction identifier.

printer busy (PRTBY) state. The state of a printer LU station which rejects all message traffic from the host until a device end status is received from its printer station.

printer data set ($.SYSPRT). The data set containing image and control store loads for 4978 terminals and members used to support printers as buffered devices.

priority, message. An attribute of a message that determines where it is placed in the destination station's message queue.

program dispatcher ($.PD). The part of the Communications Facility that manages the processing of transactions.

program dispatcher (PD) commands. A subset of Communications Facility CP commands used to control the operation of the program dispatcher.

program dispatcher data set ($.SYSPD). The data set containing CP commands, path definitions, transaction definitions, and transactions, that are to be processed when the program dispatcher is started.

program station. See *user station*.

prompt screen, SNA. An optional message put out by SNA logical units to prompt the terminal operator to log on to a host SNA application.

protected field. A field on a display screen in which the user is not allowed to enter data; also, the definition of such a field in a panel or an EDX screen image.

protocol identifier (ID). The optional 4-byte field included in the user data portion of a cell request or incoming call packet that provides an additional means of screening incoming calls from an X.25 network. In the Communications Facility, switched virtual circuit station definitions may include a protocol ID.

PSDN (packet-switching data network). A communications network that uses the mechanism of packet switching to transmit data. See also *packet switching*.

pseudo disk. A definition in one Series/1 of a disk that is attached to another Series/1.

PT. The $.HMU command that performs pass-through processing.

PU. The $.UT2 command that purges messages from a message queue.

PU (physical unit) station. A station that represents an SNA physical unit in a Communications Facility configuration.

PVC (permanent virtual circuit). A permanent virtual connection that provides services similar to a leased line. Data sent to the network through a logical channel being used as a PVC is always delivered to a specific logical channel at a specific DTE destination in the network. In the Communications Facility, PVC circuit stations represent permanent virtual circuits.

Q. (1) The CP command that displays information about Communications Facility stations, BSC lines, EDX terminals, EDX-SNA control blocks, and EDX-XHCS control blocks. (2) The PD command that displays the transaction identifier table, the path definition table, remote disk definitions, and scheduled transactions.

queued message utility ($.UT2). The utility program that allows the user to examine and purge messages on the $.WASTE queue or any other message queue.

R. The PD command that removes an entry from the path table or the transaction table.

RE. (1) The $.UT1 command that receives the next message on a station's queue. (2) The $.UT2 command that reports on messages in a particular queue.

READ. The CP command that causes execution of CP commands contained in a data set.

reason code. A code indicating the reason each undeliverable message is in the $.WASTE queue.

reason message. A message preceding each undeliverable message in the $.WASTE queue. The reason message contains the date, time, and reason code.

receive (RECV) state. The state of an LU station that allows messages to flow from host to terminal, but not from terminal to host.

recognized private operating agency (RPOA). The X.25 facility that defines a particular transit network through which a virtual call is to be routed internationally, when more that one RPOA transit network exists at an international gateway.

Recommendation X.25. The CCITT recommendation that defines standards for the connection of processing equipment to a packet-switching data packet level. It addresses the physical level, the link level, and the packet level. The Communications Facility adheres to the recommendation as amended in 1981.

record mode. An attribute of a station that causes control characters to be removed from messages received from it and control characters added, if necessary, to messages sent to it. Contrast with *basic mode*.

remote disk support. A feature of the Communications Facility that allows a program to access a disk volume that is attached to another Series/1.

remote IPL utility ($.PDIPL). The transaction-processing program that sends an IPL bootstrap to a remote Series/1, causing that Series/1 to IPL itself.

remote management utility ($.RMU). The transaction-processing program that processes transactions received from the host management utility, operating in another Series/1.

remote cell. Any cell in the Communications Facility configuration other than the local cell.

remote node. Any node in the Communications Facility configuration other than the local node.

remote station. A station in a remote node.

remove mode. An attribute of an I/O control program that prevents the sending of duplicate messages caused by a terminal user pressing RESET and ENTER. Contrast with *non-remove mode*.

RESERVE. A statement in $.SYSPD that reserves space in the path table or the transaction table.

RETRY mode. An attribute of a circuit station that causes the I/O control program to reestablish the virtual call when a call is cleared for a switched virtual circuit whose contact type is INIT. Contrast with *STOP mode*.

reverse charging (REV). The X.25 facility used to request that the cost of a communications session be charged to the called data terminal equipment.

RM. The $.HMU command that allows the host cell to receive a data set from the remote cell.

RMU. The transaction identifier associated with the Communications Facility program $.RMU.

RPOA (recognized private operating agency). The X.25 facility that defines a particular transit network through which a virtual call is to be routed internationally, when more than one RPOA transit network exists at an international gateway.

RTE. A statement in $.SYSPD that overrides a transaction's primary cell identifier.

S. (1) The CP command that starts a station. (2) The PD command that activates a path or a transaction.

S$POOL. The system storage pool; a workspace pool in the common area, used by the Communications Facility for station blocks and work areas.

SCHD. The transaction identifier associated with the Communications Facility program $.PDSCHD.

scheduler ($.PDSCHD). The transaction-processing program that schedules another transaction for processing at a specific time.

SCS (SNA character string). A character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

SCS mode. An attribute of a printer device station that causes data sent to it to be interpreted as an SNA character string. Contrast with *3270 mode*.

SD. The $.HMU command that shuts down $.RMU and, optionally, starts a program in the remote cell.

SE. The $.UT1 command that sends a text message to a station.

secondary cell identifier. A field of a transaction, whose meaning is defined by the program that processes the transaction. It may, for example, be the identifier of the cell to which an acknowledgment is to be sent.

secondary mode. An attribute of a point-to-point line station that defines the local node as the secondary Series/1. Contrast with *primary mode*.

secondary transaction identifier. A field of a transaction, whose meaning if defined by the program that processes the transaction. It may, for example, be the identifier of a transaction to be sent as an acknowledgment.

SEND CP. The instruction that sends a CP command.

sequence number, message. A number associated with a message representing its sequence with respect to its origin.

set $.CFD storage utility ($.SETSTG). The utility program that changes the size of the message buffer pool and the file control block pool in $.CFD.

SET EXIT. The $.CONFIG command that sets the exit code.

SET LOG. The CP command that assigns a log device or station for system log messages.

SET NODE. The CP command that sets the node address of a Series/1 and puts the message dispatcher into multinode mode.

single-node mode. A mode of operation in which the message dispatcher disregards the node assignment portion of network addresses; the message dispatcher is in single node mode when the $.DISP station's node assignment is 00.

SM. The $.HMU command that sends a data set from the host to the remote cell.

SNA prompt screen (panel). An optional menu put out by SNA logical units that prompts the terminal operator to log on to a host SNA application.

ST. (1) The CP command that displays message statistics and Local Communication Controller hardware statistics. (2) The work session controller command that sets the transaction identifiers of the transactions that are to be sent when a PF key on a static screen terminal is pressed after a WK command. (3) The $.UT2 command that identifies the station whose message queue is to be processed.

standard field. A protected field of a panel (created through $.PANEL) that has default attributes.

started station. A station that is represented by a station block in S$POOL.

station. A named unit of hardware or software managed by the Communications Facility. Stations were called *logical units* in previous Communications Facility field developed programs.

station address. The last two characters of a station's network address. They uniquely identify the station within the node.

station block. A control block in S$POOL that contains information about a started station.

station name. A 1-to 8-character alphameric value that uniquely identifies each station in a node.

station subtype. An attribute of a station that further defines its type; for example, a device-type station may have a subtype such as 3101, 4978, or printer.

station type. An attribute of a station that specifies its type; for example, a device-type station may have a subtype such as 3101, 4978, or printer.

status message. A message sent with a SEND S or SEND SM command that results in a unique return code (+6) when it is received. A status message is used to (1) tell a station to stop or halt (2) send X.25 control messages between the X.25 I/O control program and applications linked (alternate or direct) to a STD+ circuit station.

STOP mode. The attribute of a circuit station that causes the I/O control program to stop the station when a call is cleared for a switched virtual circuit whose contact type is INIT. Contrast with *RETRY mode.*

stopped station. A station, represented by a station block in S$POOL, for which the flow of messages has been temporarily stopped. Messages sent to a stopped station are undeliverable.

subtype, station. An attribute of a station that further defines its type; for example, a device-type station may have a subtype such as 3101, 4978, or printer.

SVC (switched virtual circuit). A dynamically-established connection between two pieces of data terminal equipment (DTE). The switched virtual circuit is the packet network equivalent of a switched or dial-up line. In the Communications Facility, SVC circuit stations represent switched virtual circuits.

switched virtual circuit (SVC). A dynamically-established connection between two pieces of data terminal equipment (DTE). The switched virtual circuit is the packet network equivalent of a switched or dial-up line. In the Communications Facility, SVC circuit stations represent switched virtual circuits.

system initialization data set ($.SYSIPL). The data set containing CP commands to be executed when the Communications Facility is loaded.

system station. The station block, named $.DISP, that represents the Communications Facility control program.

system storage pool (S$POOL). A workspace pool in the EDX supervisor, used by the Communications Facility for station blocks and work areas.

SYST CP. The transaction that sends a CP command to a cell.

SYST HL. The transaction that causes the high-speed loader to load a program.

SYST ID. The transaction that checks whether a cell is active.

SYST LU. See *SYST ST.*

SYST MS. The transaction that sends a message to an EDX terminal.

SYST RC. The transaction that sets the number of times the program dispatcher will attempt to load a program when storage is not available.

SYST RH. The transaction that releases previously held transactions.

SYST SP. The transaction that starts remote disk support in a cell.

SYST ST. The transaction that checks whether a station is started.

SYST TI. The transaction that sends the system time and date to a cell.

SYST TR. The transaction that starts or stops a trace of transactions.

SYST UP. The transaction that specifies whether or not the program dispatcher is to load program $.UP*xxxx* when it receives the undefined transaction *xxxx*.

SYST WH. The transaction that checks whether a program is loaded.

T. The PD command that sends the system time and date to a cell.

terminal, Communications Facility. A device defined to the Communications Facility, controlled by an I/O control program, and accessed from a program through SEND and RECEIVE instructions.

terminal, EDX. A terminal defined to the EDX operating system and used to perform EDX system functions.

terminal, work session controller. A terminal managed by the work session controller and accessed from an application program by means of work session controller transactions.

terminal station. A station that represents a 3270 control unit, display, or printer attached to a Series/1, or a station block used to emulate a 3270 control unit, display, or printer in a Communications Facility configuration.

text mode. An attribute of a terminal or device station that causes lowercase data to be transferred without modification. Contrast with *non-display mode.*

TID (transaction identifier). The 4-character name of a transaction.

TID statement. A statement in $.SYSPD that identifies a transaction to be processed in the local cell.

TRAC. The PD command that starts or stops a trace of transactions.

TRAN. The PD command that sends transactions.

transaction. A special-format, user-defined message, routed through the Communications Facility network by the program dispatcher and processed at its destination by a specific transaction-processing program.

transaction identifier (TID). The 4-character name of a transaction.

transaction identifier table. The table that defines the transactions to be processed in the local cell. It contains, for each transaction, its identifier, its attributes, and the name and attributes of its associated programs.

transaction message. A message, the content of which is a transaction.

transaction-processing program. A program designed to process transactions. The program dispatcher controls loading and execution of transaction-processing programs as it receives transactions.

transaction table. Synonymous with *transaction identifier (TID) table.*

transaction type. A 2-character indicator of the actions that occur when a transaction is entered: loading one of four types of program, creating a station, and/or sending the transaction message to the station.

transparent mode. A mode of BSC transmission that allows any bit pattern to be transmitted as data.

type, line. The combination of station type and station subtype for a communication line. Line type indicates which input/output control program (such as point-to-point or 3270 control) is to control a particular line.

type, message. An attribute of a message that indicates whether it is a data, command, log, transaction, or status message.

type, station. An attribute of a station that specifies its type (for example, line, device, terminal, user, or message station).

type, transaction. A 2-character indicator of the actions that occur when a transaction is entered: loading one of four types of program, creating a station, and/or sending the transaction message to the station.

UBRETRY mode. The attribute of an SNA LU station that causes the I/O control program to attempt to rebind a session that was terminated by the host. Contrast with *UBSTOP mode.*

UBSTOP mode. The attribute of an SNA LU station that causes the I/O control program to stop the station when the host terminates a session. Contrast with *UBRETRY mode.*

undeliverable message. A message that cannot be delivered because its destination station is stopped or is unknown to the message dispatcher.

unprotected field. A field on a display screen in which the user is allowed to enter data; also, the definition of such a field in a panel or an EDX screen image.

UP. The PD command that specifies whether or not the program dispatcher is to load program $.UP*xxxx* when it receives the undefined transaction *xxxx.*

user station. A station that represents a user or system program in a Communications Facility configuration.

V. The CP command that removes the local node from the Local Communications Controller ring.

vector station. A station block that represents a remote station in a multinode Communications Facility configuration.

virtual call. A temporary logical connection between two pieces of data terminal equipment. Virtual calls are placed through switched virtual circuits.

virtual circuit. A logical connection established between two pieces of data terminal equipment. It can be permanent—defined when you subscribe to your network port—or it can be switched—dynamically established when a call is placed. The Communications Facility manages stations that represent these circuits; XHCS manages the circuits. See also *switched virtual circuit* and *permanent virtual circuit.*

volume protection utility ($.PDBSTS). The utility program that allows a user to gain exclusive control of a disk volume directory in a remote Series/1 before loading a program that will update that directory.

window. (1) The number of data packets a DTE or DCE can send across a logical channel before waiting for authorization to send another data packet. It is the main mechanism for pacing the flow of X.25 packets across an X.25 network. In the Communications Facility, window is defined in the line or circuit station definition. (2) In $.PANEL, the area on the screen that can be seen one time when defining a 3270 panel.

work session controller ($.WSC). The part of the Communications Facility that allows an application program to communicate with multiple EDX devices attached to any Series/1 in the network.

work session controller data set ($.WSCIMG). The data set containing images that can be displayed through the work session controller and members used to save data for transaction-processing programs.

workspace pool. An area of processor storage from which the Communications Facility programs allocate buffers and work areas. The pool includes information used to control the allocation of elements in the pool.

WR. The $.HMU command that sends the remote cell a message which $.RMU returns to the host cell.

WSC. (1) The transaction identifier associated with the Communications Facility program $.WSC. (2) The command that starts a work session controller terminal.

X.25 control message. See *control message.*

X.25 data set. See *$.SYSX25*

X.25 network. A packet-switching data network that adheres to the standards defined by the CCITT Recommendation X.25.

X.25 network address. A field of up to 15 binary-coded decimal (BCD) digits that identifies the DTE to which a call is directed or from which a call originated. The Communications Facility provides a data set, $.SYSX25, in which the user may relate this address to a two-digit call ID.

XHCS. The IBM Series/1 Event Driven Executive X.25/HDLC Communications Support licensed program (Program Number 5719-HD2). XHCS allows an application program, such as the Communications Facility, to communicate with remote applications through an HDLC communications link using X.25 packet level procedures.

XOFF mode. An attribute of a point-to-point line station that, in conjunction with record mode, causes data for that line it to be written in nontransparent mode. Contrast with *XON mode.*

XON mode. An attribute of a point-to-point line station that, in conjunction with record mode, causes data for that line it to be written in transparent mode. Contrast with *XOFF mode.*

3270 control. The input/output control program that controls 3270 displays and printers attached to the Series/1.

3270 emulation. The facility that allows a host processor to communicate with a Series/1 as if it were communicating with a 3270 system; also, the input/output control program that provides 3270 emulation over a BSC line.

3270 mode. An attribute of a printer device station that causes data sent to it to be interpreted as a 3270 data stream. Contrast with *SCS mode.*

Index

Special Characters

C

D

F

G

H

L

M

N

O

P

S

T

3

4

7

READER'S COMMENT FORM

SL23-0105-0

IBM Series/1 EDX
Communications Facility
Operator's Guide

This form may be used to comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers).

IBM may use and 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.

If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
  FOLD ON TWO LINES, SEAL AND MAIL

Fold and tape

Please do not staple

Fold and tape

No postage
necessary
if mailed
in the U.S.A.

Business Reply Mail

First Class Permit 40 Armonk New York

Postage will be paid by:

International Business Machines Corporation
System Products Division
Dept 4J1/037 - PAS5 MS33
1501 California Avenue
P. O. Box 10500
Palo Alto, CA 94304

Fold and tape

Please do not staple

Fold and tape

READER'S COMMENT FORM

SL23-0105-0

IBM Series/1 EDX
Communications Facility
Operator's Guide

This form may be used to comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers).

IBM may use and 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.

If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
  FOLD ON TWO LINES, SEAL AND MAIL

Fold and tape Please do not staple Fold and tape

No postage necessary if mailed in the U.S.A.

Business Reply Mail

First Class Permit 40 Armonk New York

Postage will be paid by:

International Business Machines Corporation
System Products Division
Dept 4J1/037 - PAS5 MS33
1501 California Avenue
P. O. Box 10500
Palo Alto, CA 94304

Fold and tape Please do not staple Fold and tape

READER'S COMMENT FORM

SL23-0105-0

IBM Series/1 EDX
Communications Facility
Operator's Guide

This form may be used to comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers).

IBM may use and 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.

If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, SEAL AND MAIL

Fold and tape

Please do not staple

Fold and tape

No postage necessary if mailed in the U.S.A.

Business Reply Mail

First Class Permit 40 Armonk New York

Postage will be paid by:

International Business Machines Corporation
System Products Division
Dept 4J1/037 - PAS5 MS33
1501 California Avenue
P. O. Box 10500
Palo Alto, CA 94304

Fold and tape

Please do not staple

Fold and tape

READER'S COMMENT FORM

SL23-0105-0

IBM Series/1 EDX
Communications Facility
Operator's Guide

This form may be used to comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers).

IBM may use and 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.

If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
  FOLD ON TWO LINES, SEAL AND MAIL

Fold and tape

Please do not staple

Fold and tape

No postage necessary if mailed in the U.S.A.

Business Reply Mail

First Class Permit 40 Armonk New York

Postage will be paid by:

International Business Machines Corporation
System Products Division
Dept 4J1/037 - PAS5 MS33
1501 California Avenue
P. O. Box 10500
Palo Alto, CA 94304

Fold and tape

Please do not staple

Fold and tape

READER'S COMMENT FORM

SL23-0105-0

IBM Series/1 EDX
Communications Facility
Operator's Guide

This form may be used to comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers).

IBM may use and 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.

If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
  FOLD ON TWO LINES, SEAL AND MAIL

Fold and tape | Please do not staple | Fold and tape

No postage necessary if mailed in the U.S.A.

Business Reply Mail

First Class Permit 40 Armonk New York

Postage will be paid by:

International Business Machines Corporation
System Products Division
Dept 4J1/037 - PAS5 MS33
1501 California Avenue
P. O. Box 10500
Palo Alto, CA 94304

Fold and tape | Please do not staple | Fold and tape

Technical Newsletter

This Newsletter No.	SN23-8601
Date	March 27, 1987
Base Publication No.	SL23-0105-1
	S1-40
Previous Newsletters	SN23-0161

IBM Series/1
EDX Communications Facility
Operators Guide

Program Number 5719-CF2

This technical newsletter provides replacement pages to add new information for Version 2.1 PTF 2 of the Communications Facility. The replacement pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be replaced are:

v to iii	139, 140
47, 48	175, 176
118.3, 118.4 (added)	

Technical changes are indicated by a vertical line to the left of each change.

Summary of Changes

The retry option and the cold start option were added.
Technical and typographical corrections have also been made.

Note: *Please file this cover letter at the back of the manual to provide a record of changes.*

IBM Corporation, Department 4J1/036-MS33, 1520 California Avenue, P.O. Box 10500, Palo Alto, California 94304

Printed in USA

IBM Technical Newsletter

This Newsletter No.	SN23-0161
Date	September 9, 1986
Base Publication No.	SL23-0105-1 S1-40
Previous Newsletters	None

IBM Series/1
EDX Communications Facility
Operator's Guide

This technical newsletter provides replacement pages to add new information for Version 2.1 PTF 1 of the Communications Facility. The replacement pages remain in effect for subsequent versions and modifications unless specifically altered. Pages to be replaced are:

47, 48	143, 144
67, 68	153-156
99, 100	159-164
100.1, 100.2 (added)	215, 216
111, 112	253-256
117, 118	303, 304
118.1, 118.2 (added)	

Technical changes are indicated by a vertical line to the left of each change.

Summary of Changes

EDX Secondary SNA2 support was added.

Technical and typographical corrections have also been made.

Note: *Please file this cover letter at the back of the manual to provide a record of changes.*

IBM Corporation, Department 4J1/037-PAS5, 1501 California Avenue, P.O. Box 10500, Palo Alto, California 94304

Printed in USA

SL23-0105-0

SL23-0105-0
Program Number:
5719-CF2
File Number: S1-40

Printed in U.S.A.