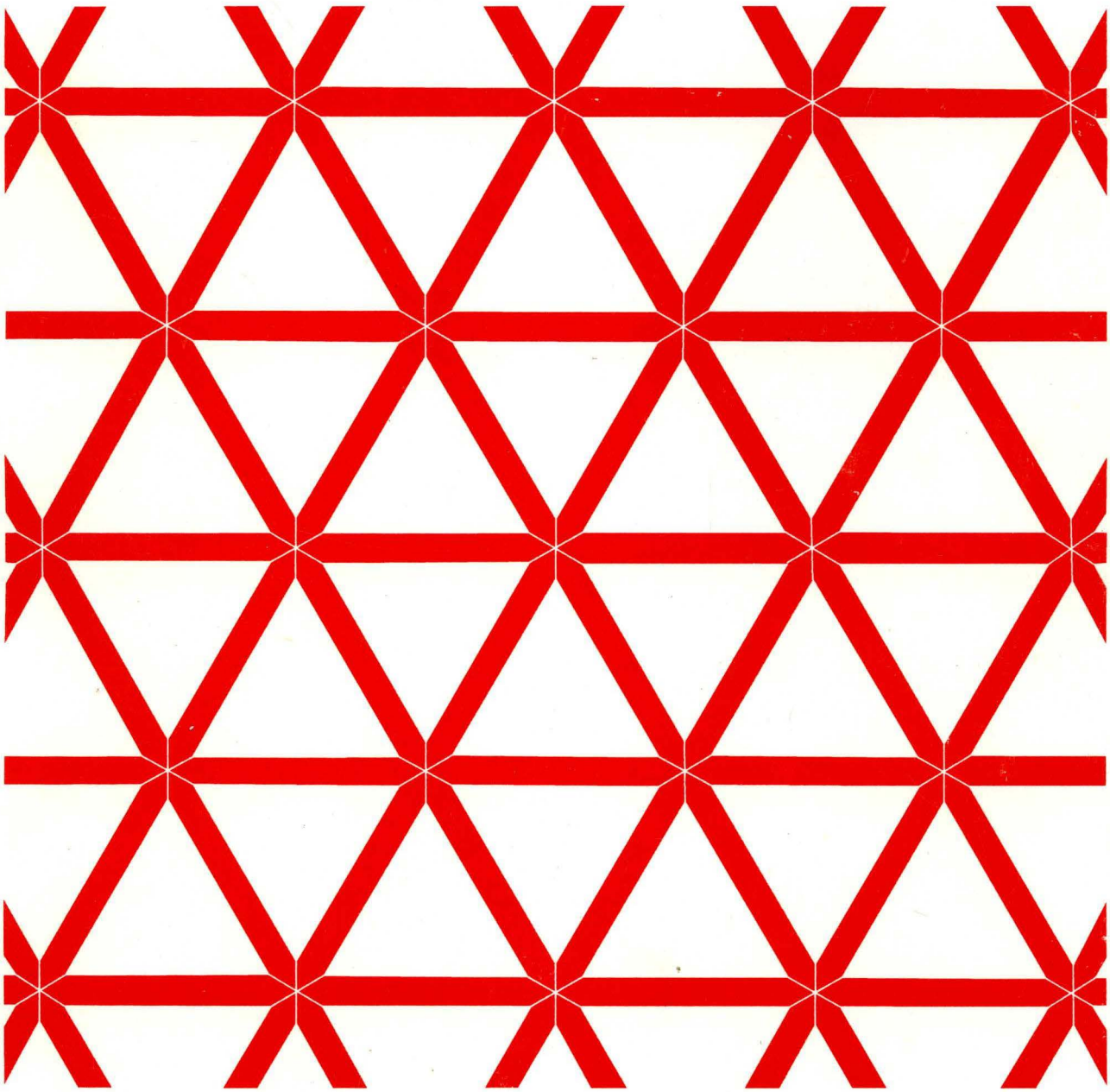
SC09-1255-00

Transmission Control Protocol/ Internet Protocol for MVS

User's Guide



SC09-1255-00

Transmission Control Protocol/ Internet Protocol for MVS

User's Guide

First Edition (June 1989)

This edition applies to the Transmission Control Protocol/Internet Protocol for MVS Licensed Program (Program Number 5685-061), and to all subsequent releases and modifications until otherwise indicated in new editions. Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Note to US Government Users: Documentation related to restricted rights. Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

IBM Canada Ltd.,
Information Development,
Department 849,
1150 Eglinton Avenue East,
North York, Ontario, Canada.
M3C 1H7

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

About This Book

This book describes how to use the end-user interfaces available with the IBM Transmission Control Protocol/Internet Protocol for MVS (TCP/IP for MVS) program. These interfaces allow MVS and non-MVS hosts to communicate across an internetwork (internet).

Throughout this book, the abbreviation MVS refers to the following IBM products:

- IBM Multiple Virtual Storage/System Product Version 1 Release 3.5 (MVS/370), or later
- IBM Multiple Virtual Storage/System Product Version 2 Release 1.3 (MVS/XA™), or later
- IBM Multiple Virtual Storage/System Product Version 3 Release 1.0 (MVS/ESA™), or later.

This book is written for a person who is familiar with the IBM Multiple Virtual Storage Operating System (OS/MVS), the IBM Time Sharing Option (TSO), and their commands.

This book does not contain any programming interfaces for customers. For information on the programming interfaces provided by the IBM TCP/IP for MVS program, refer to the *IBM TCP/IP for MVS: Programmer's Reference.*

What This Book Contains

This book contains the following information:

- Chapter 1, "Computer Networks and TCP/IP Protocols" gives an overview of TCP/IP networks and the TCP/IP set of protocols.
- Chapter 2, "Transferring Data Using the File Transfer Protocol" describes how to transfer data using the File Transfer Protocol (FTP).
- Chapter 3, "Sending Electronic Mail" describes how to send electronic mail using an interface to the Simple Mail Transfer Protocol (SMTP).
- Chapter 4, "Logging on to a Host" describes how to log on to foreign host systems using the Telnet Protocol.
- Chapter 5, "Obtaining Network Status Information" describes how to obtain network status information using the NETSTAT command.
- Chapter 6, "PING Command" describes how to perform network testing using the PING command.
- Chapter 7, "Using the Network File System" describes how to use the Network File System server feature.

MVS/XA and MVS/ESA are trademarks of the International Business Machines Corporation.

- Appendix A, "Miscellaneous Commands" describes a few miscellaneous commands.
- Appendix B, "Network File System Messages for the Client" lists the Network File System server feature error messages that are sent to the clients.
- Appendix C, "Other Related Publications" lists the publications that are used as the protocol specifications.

Terms Used in This Book

Within the TCP/IP for MVS environment, you should be familiar with usage of the following terms:

Term	Description
local host	In an internet, any computer to which an end-user or a functional unit is connected without the use of the internet.
foreign host	In an internet, any host on the network other than the local host.
data set	Refers to the basic unit of data storage for MVS. Unless otherwise specified, the use of this term indicates that the MVS host storing the data set is your local host system.
file	Refers to the basic unit of data storage for all foreign hosts. Unless otherwise specified, the use of this term indicates that the host storing the file is a non-MVS foreign host system.

The glossary at the end of this book defines additional terms associated with TCP/IP communication in an internet environment.

How to Read the Syntax Diagrams

This book presents the command descriptions using "railroad track" syntax diagrams. The structure of these diagrams is defined below.

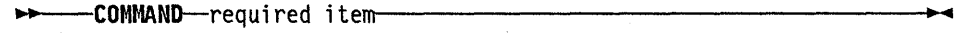
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

  The ►►— symbol indicates the beginning of a statement.

  The —► symbol indicates that the statement syntax is continued on the next line.

  The ►— symbol indicates that a statement is continued from the previous line.

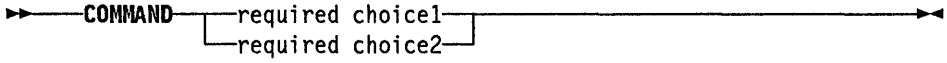
  The —►◄ symbol indicates the end of a statement.
- Required items appear on the horizontal line (the main path).



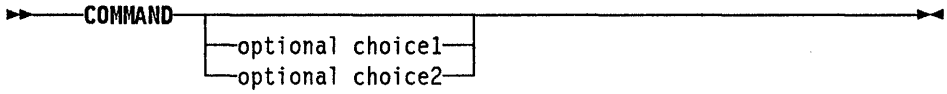
- Optional items appear below the main path.

- If you can choose from two or more items, they appear vertically, in a stack.

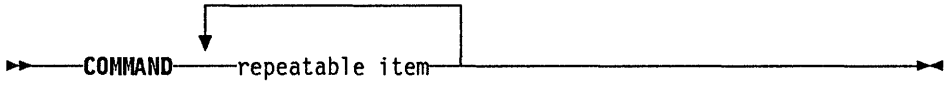
  If you *must* choose one of the items, one item of the stack appears on the main path.



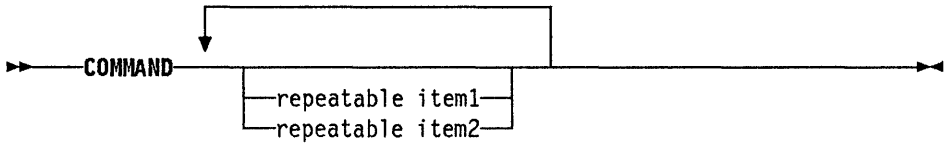
  If all items are optional, the entire stack appears below the main path.



- An arrow returning to the left above the main line indicates an item that can be repeated.



- A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.

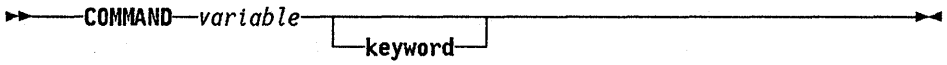


- Commands are shown in **BOLD UPPERCASE** letters in the syntax. They are not case sensitive.

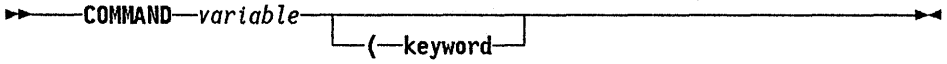
  Note: Network File System commands must be issued in lowercase at the client. Because of this restriction, they are shown in **bold lowercase**.

Keywords are shown in **bold lowercase** letters in the syntax. They should be entered exactly as shown, but are not case sensitive.

Variables appear in *italic lowercase* letters in the syntax. They represent user-supplied names or values.



- If punctuation marks, parentheses, arithmetic operators, or such symbols are shown, you must enter them as part of the syntax.



Specifying Data Sets and Files

Some of the commands in this book take a file name or a data set name as a parameter. The format used to name a file is host-dependent. Some systems limit the length of a file name, and some systems are case sensitive.

Specifying MVS Data Sets

If a host uses MVS, data is stored in data sets. Specify data sets in the following format:

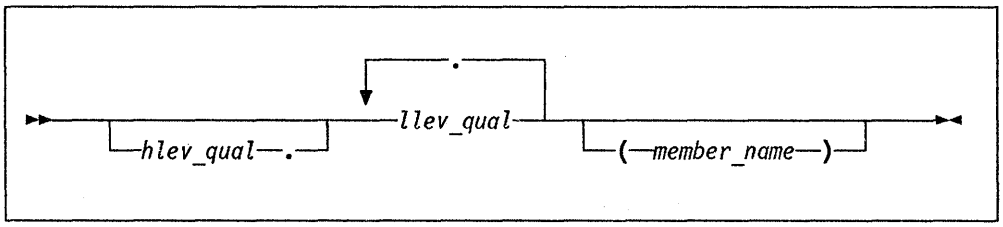


Figure 0-1. MVS Data Set Naming Format

Parameter	Description
hlev_qual	Is the high level qualifier of the data set. It is the user ID of the TSO user. If you specify this parameter, the complete data set name must be enclosed within single quotes (*'data_set_name'*).
llev_qual	Is a low level qualifier of the data set. You must specify this qualifier.
member_name	Can be specified with partitioned data sets (PDS). A PDS consists of sequential records in independent groups called members. Each member has a unique name.

When you specify a data set, the minimum qualifier that you can specify is the *llev_qual*. Values for the *hlev_qual* will be supplied by the system unless the data set name is contained within single quotes ('). The name contained within single quotes is considered to be a fully qualified data set.

Each qualifier and member name consists of one to eight alphanumeric characters and begins with an alphabetic (A-Z) or national ($,@, and #) character. All data set names are converted to uppercase. The total length of the data set name, including periods, must not exceed 44 characters. The member name can be an additional 10 characters, including the parentheses.

You can use the special character asterisk (*) for pattern matching when you specify a data set.

The following example shows you how to specify an MVS data set. For example, you want to specify the **accounts** member of the **user1.cprogs** PDS. The *hlev_qual* is **user1**, the *llev_qual* is **cprogs**, and the *member_name* is **accounts**. If the high level qualifier is supplied by the system, specify the PDS as follows:

cprogs(accounts)

The PDS can be fully qualified as follows:

'user1.cprogs(accounts)'

Specifying VM Files

If a host uses VM, data is stored in files. Specify VM files in the following format:

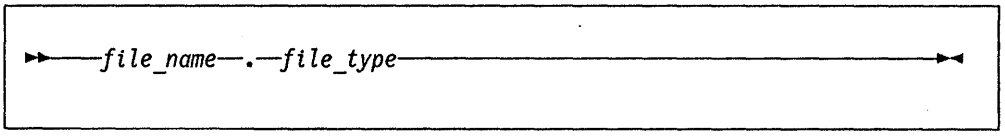


Figure 0-2. VM File Naming Format

Parameter	Description
file_name	Is the file name.
file_type	Is the file type.

Note: The file mode is not accepted by foreign VM hosts; it is taken to be the file mode associated with the current working directory. The file mode is never used in TCP/IP for MVS commands.

All VM file specifications are treated as if they are entered in uppercase. The file name and file type consist of one to eight alphanumeric characters. Other valid characters are dollar ($), number sign (#), at sign (@), plus (+), hyphen (-), colon (:), and underscore (_).

You can use the special character asterisk (*) for pattern matching.

The following example shows how to specify a VM file. For example, you want to specify the following file: **accounts cprog**. The *file_name* is **accounts**, and the *file_type* is **cprog**. Specify the file as follows:

accounts.cprog

Refer to the *IBM VM/SP: CMS User's Guide* (SC23-0356) for more information about the VM file system.

Specifying AIX Files

If a host uses the Advanced Interactive Executive (AIX) operating system, data is stored in files. Specify AIX files in the following format:

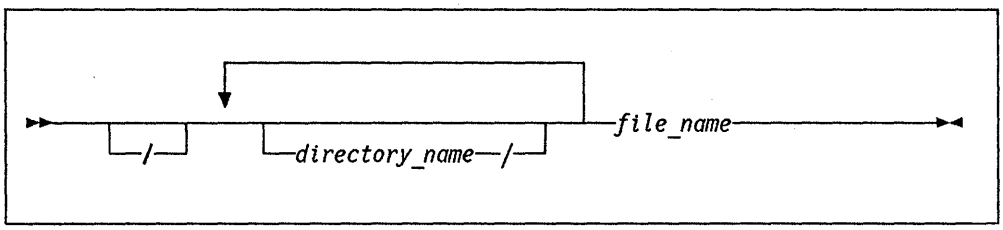


Figure 0-3. AIX File Naming Format

Parameter	Description
directory_name	Is a directory name. Directories contain the names of files, other directories, or both. Directory names can be up to 14 characters long.
file_name	Is the file name. It can be up to 14 characters long.

Specify the first slash (/) only when you begin at root directory. Otherwise, the file specification starts from the current directory.

The AIX Operating System distinguishes between uppercase and lowercase letters in file names.

Directory names and file names should not include characters that have a special meaning to the shell, including back slash (\), ampersand (&), and period (.).

The following example shows you how to specify an AIX file. For example, you want to specify a file called **accounts** that is contained in a directory called **cprograms**. This **cprograms** directory is, in turn, contained in the root directory. The *file_name* is **accounts**, and the *directory_name* is **cprograms**. Specify the file as follows:

/cprograms/accounts

where the first slash (/) represents the root directory. If the current directory is **cprograms**, specify the file as follows:

accounts

Refer to the *IBM RT Using the AIX Operating System* (SC23-0794) book for more information about the AIX file system.

Specifying DOS Files

If a host uses the Disk Operating System (DOS), data is stored in files. Specify DOS files in the following format:

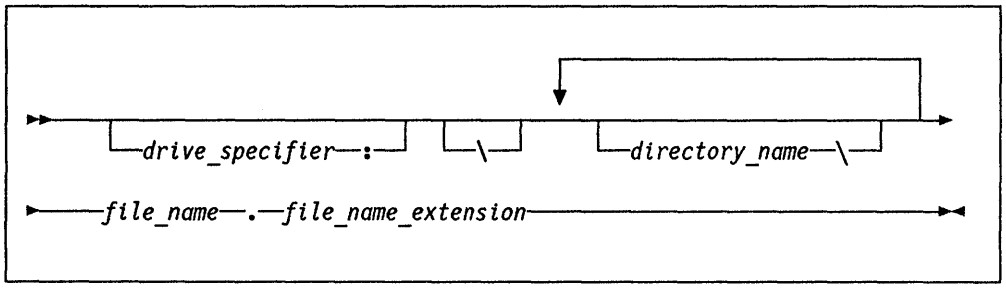


Figure 0-4. DOS File Naming Format

Parameter	Description
drive_specifier	Is the drive that contains the file you want to reference. If you omit the drive specifier, DOS assumes the file is located in the default drive.
directory_name	Is the directory name that contains the file you want to reference. It consists of one to eight characters.
file_name	Is the file name. It consists of one to eight characters.
file_name_extension	Is the file name extension. It consists of one to three characters.

Specify the first back slash (\) only when you begin at the root directory.

The following characters are not allowed in file names and file name extensions:

period (.)	double quote (")	back left (\)
slash (/)	left bracket ([)	right bracket (])
colon (:)	split vertical bar (¦)	greater than (>)
less than (<)	plus (+)	equals (=)
semicolon (;)	comma (,)	

as well as ASCII characters less than X'20'. All other characters are allowed.

The following example shows you how to specify a DOS file. For example, you want to specify an executable DOS file called **accounts**. This file is contained on the host's **c** drive. The *drive_specifier* is **c**, the *directory_name* is **programs**, the *file_name* is **accounts**, and the *file_name_extension* is **exe** because it is an executable DOS file. Specify the file as follows:

c:\programs\accounts.exe

where the first back slash (\) represents the root directory.

Refer to the *IBM Disk Operating System: Version 4.0* (G360-2819) book for more information about the DOS file specification.

Specifying a Host Name

Many of the commands presented in this book require that you specify a host as a parameter. Specify the host by its internet host name in the form:

internet_host_name

where *internet_host_name* is a name associated to the internet address of the host.

You can also specify the host by its internet address in the dotted-decimal form as follows:

nnn.nnn.nnn.nnn

where *nnn* is a decimal number representing an 8-bit binary number. An internet address is made up of four 8-bit numbers. Specify these 8-bit numbers by their decimal equivalent, and delimit them with a period (.).

Specifying a host by its *internet_host_name* is useful because you do not have to you remember the more complex internal internet address of the host.

Recommended Reading

For an overview and introduction to internetworking with TCP/IP, read:

Introducing IBM's Transmission Control Protocol/Internet Protocol Products, GC09-1307.

IBM Publications Referenced in This Book

At times, this book refers you to other books for additional information. These are the books:

IBM Disk Operating System: Version 4.0, G360-2819

IBM MVS/370 JCL Reference, GC28-1350

IBM MVS Extended Architecture JCL Reference, GC28-1352

IBM RT Using the AIX Operating System, SC23-0794

IBM Transmission Control Protocol/Internet Protocol for MVS: Installation and Maintenance, SC09-1256

IBM Transmission Control Protocol/Internet Protocol for MVS: Programmer's Reference, SC09-1261

IBM TSO Extensions CLISTS, SC28-1304

IBM TSO Extensions Command Language Reference, SC28-1307

IBM TSO Extensions Interactive Data Transmission Facility: User's Guide, SC28-1104

IBM TSO Extensions User's Guide, SC28-1333

IBM VM/SP: CMS User's Guide, SC23-0356.

Other IBM Books that You Refer to

In addition to the books referred to, you may want to consult others in the TCP/IP library. Following is a list of the other books in the library:

IBM Transmission Control Protocol/Internet Protocol for MVS: Installation and Maintenance, SC09-1256

IBM Transmission Control Protocol/Internet Protocol for MVS: Programmer's Reference, SC09-1261

IBM Transmission Control Protocol/Internet Protocol for VM: Command Reference Manual, GC09-1204

IBM Transmission Control Protocol/Internet Protocol for VM: Installation and Maintenance Manual, GC09-1203

IBM Transmission Control Protocol/Internet Protocol for VM: Network File System and Remote Procedure Call Manual, SC09-1274

IBM Transmission Control Protocol/Internet Protocol for VM: Programmer's Manual, GC09-1206

IBM Transmission Control Protocol for the Personal System/2 Computer: Command Reference and Installation Manual, SC09-1270

IBM AIX PS/2 Transmission Control Protocol/Internet Protocol User's Guide, SC23-2047

IBM RT AIX Interface Program for use with TCP/IP: Version 2.2.1, SC23-2005.

For information about the IBM Time Sharing Option (TSO), you may want to refer to the:

IBM TSO Extensions Primer, GC28-1292.

Contents

Figures

Tables

Chapter 1. Computer Networks and TCP/IP Protocols

This chapter provides background information about computer networks, explains the concept of internetwork communication using Transmission Control Protocol/Internet Protocol (TCP/IP), and introduces the IBM TCP/IP for MVS product.

Network Basics

A computer network is a collection of computer nodes physically connected by a suitable communications medium. A computer node can be a microcomputer, computer workstation, minicomputer, or mainframe. The arrangement and connection of network nodes is known as the network topology.

Figure 1-1 shows several common network topologies.

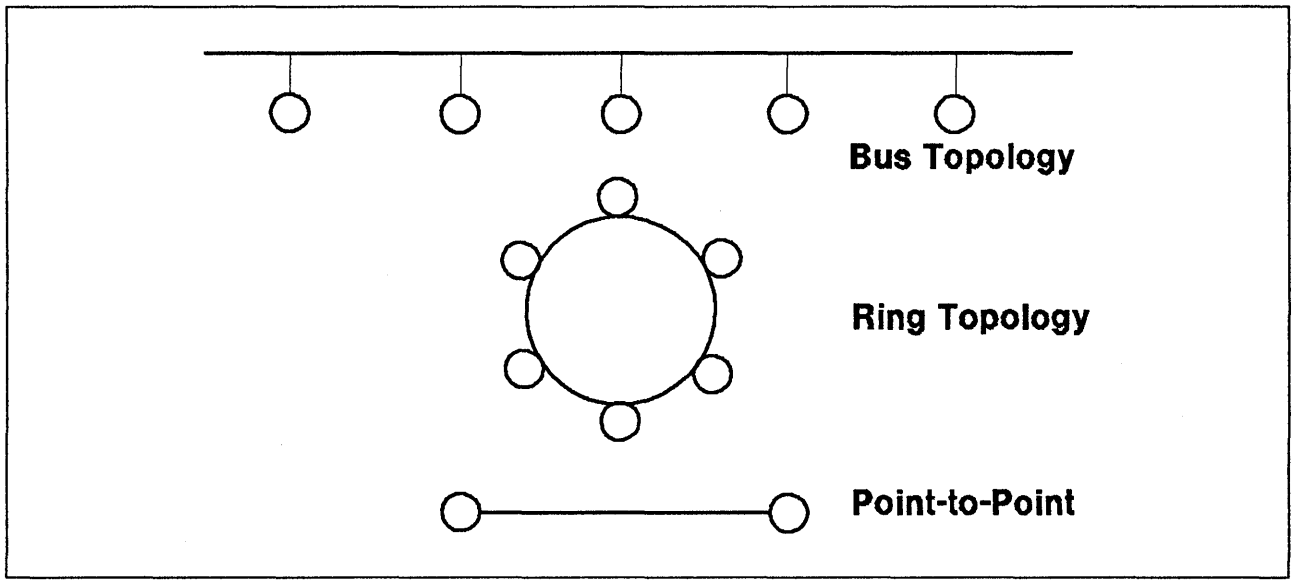


Figure 1-1. Examples of Bus, Ring, and Point-to-Point Network Topologies

The purpose of a computer network is to provide communication between nodes, resource sharing, and distributed computing. Communication applications include electronic mail, remote logon, and file transfer. Resource sharing refers to the access to limited resources, such as disk space and printers, by many computers on the network. Distributed computing refers to distribution of workload among hosts. A designated computer on the network (the server) makes a specialized service available to other computers on the network (the clients). Different computers provide different services for the benefit of the entire network.

A network where all nodes are treated the same, regardless of size, is called a peer-to-peer network.

Gateways

Networks are linked together through a common node called a gateway. The gateway performs all protocol conversion required for communication across networks. The network to which a node is physically connected is the local network; the network to which it is linked by a gateway is the foreign network. This local/foreign concept applies not only to networks, but also to hosts.

Internetwork Communication

An internetwork or *internet* is a collection of packet-switched networks interconnected by gateways to form a single, large virtual network.

Conceptually, an internet is equivalent to universal connectivity. It means all nodes on all interconnected networks can communicate as if they were all on the same physical network, regardless of their specific hardware or software architecture. This cooperation among otherwise incompatible networks and systems is known as *interoperability*.

Figure 1-2 shows the various ways networks can be interconnected in an internet.

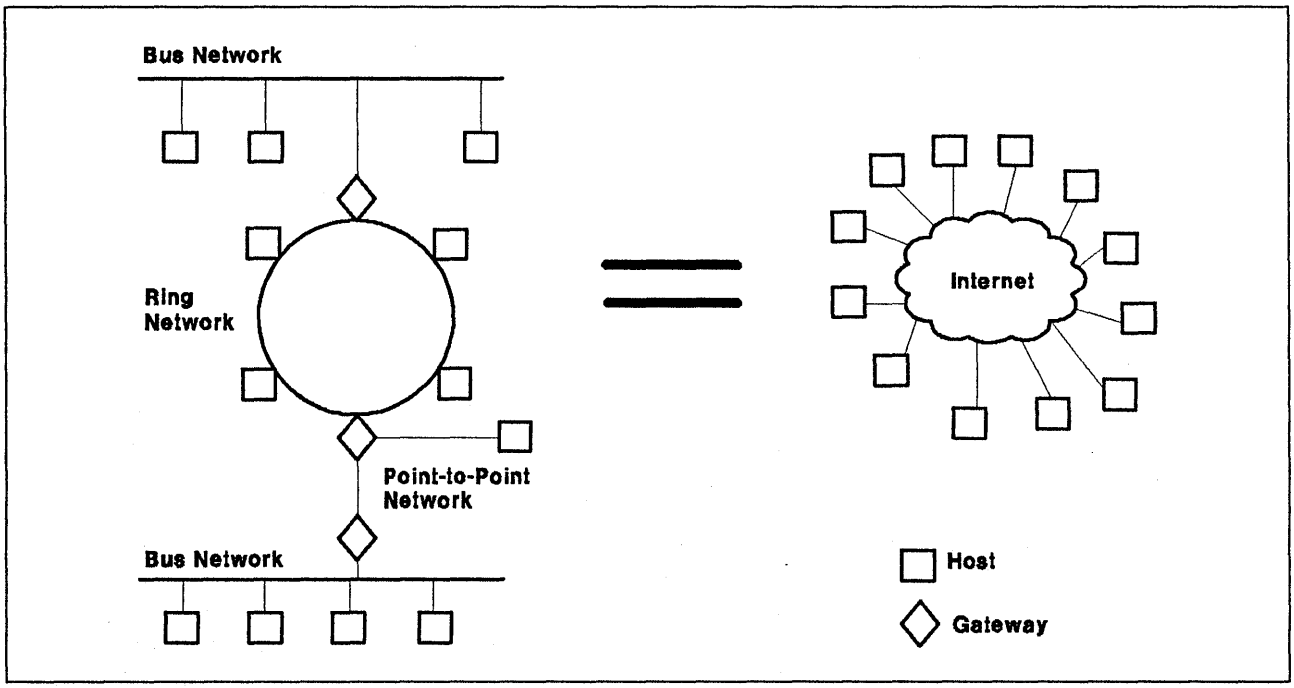


Figure 1-2. A Typical Internetwork

When capitalized, internet, that is, Internet, refers to a specific internetwork that includes ARPANET, MILNET, and NSFnet.

Each network connection of each node on an internetwork must be assigned a unique address according to conventions discussed later in this chapter. This is different from a hardware address, which is often preset by the manufacturer. Also, internet addresses follow a standard format, while different hardware types use different address lengths and formats.

TCP/IP Protocols and Network Software

Network protocols are formal descriptions of the sequence and content of data packets exchanged between network nodes. Because network protocols are implemented in the network software, these terms are often used interchangeably.

Internetwork communication is dependent on TCP/IP, a family of nonproprietary network level protocols, collectively referred to as a protocol suite. TCP/IP allows disparate packet-switched computer networks to function as a single coordinated entity. Originally developed to link military, government research and university networks, TCP/IP now has many commercial users and applications.

The TCP/IP protocol suite forms a layered structure of protocols (see Figure 1-3) ranging from low-level hardware-dependent software to high-level applications. Each TCP/IP layer provides services to the layer above it and uses the services provided by the layer below it. The lowest layer, which is next to the hardware, is not defined by TCP/IP. This layer consists of the hardware specific network protocols.

Figure 1-3 shows where TCP/IP protocols are positioned in relation to network software.

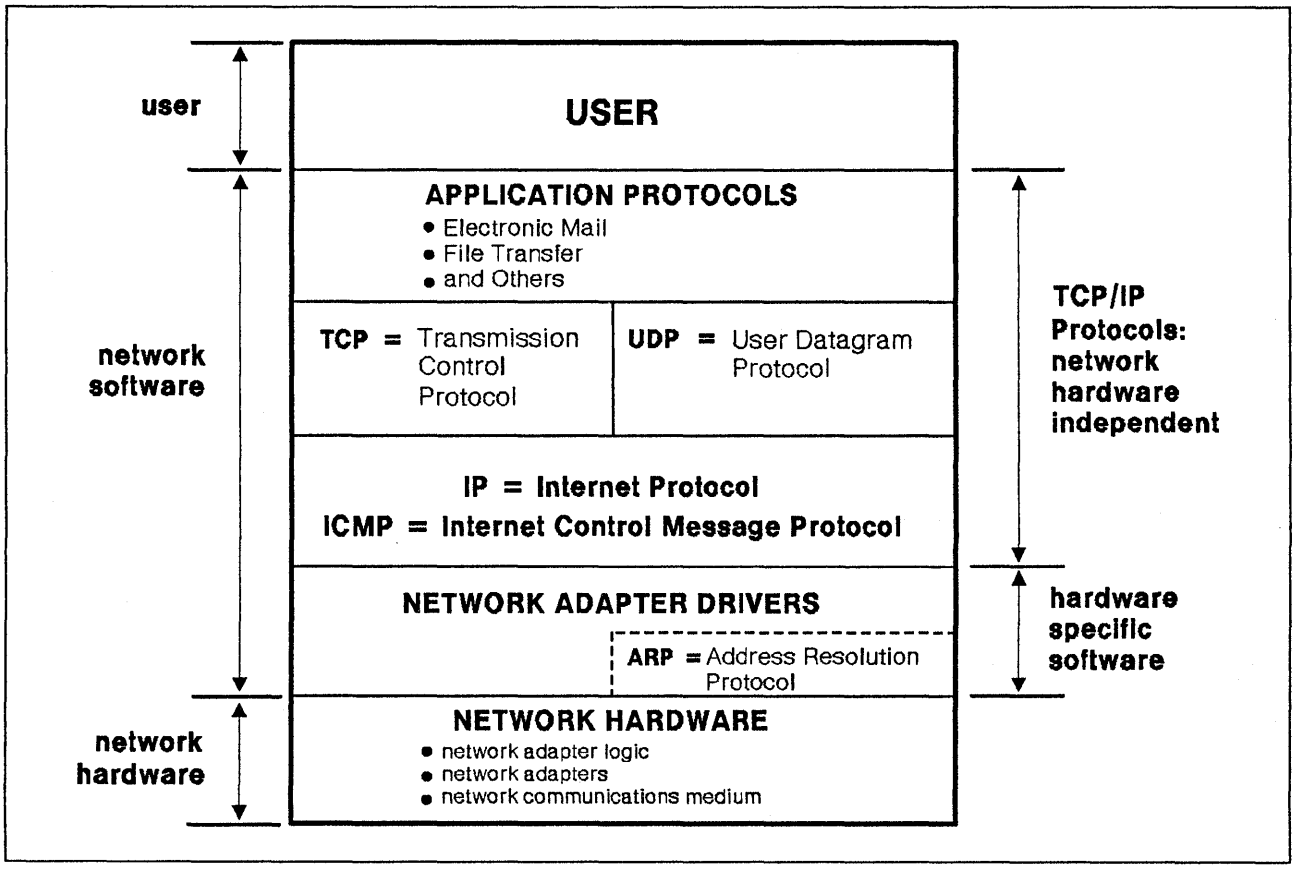


Figure 1-3. Relationship of TCP/IP to Network Software

Following are brief descriptions of the layers making up the TCP/IP protocol suite.

Internet Protocol

The Internet Protocol (IP) provides the basic transport mechanism for communication between hosts on the different networks that make up an internetwork. IP is responsible for making the underlying interconnected networks appear to the layers above as a single, large, virtual network. IP is thus responsible for implementing the internet concept, which it accomplishes by routing packets from a host on one network through a series of gateways to a host on another network. At the internet level, all communication is host-to-host, using fixed length addresses to identify source and destination hosts. The protocol layers above only need to know each host's internet address to make a connection.

In computer networks that use TCP/IP protocols, information is transmitted between nodes in the form of *packets* of data. Outgoing packets can be fragmented into multiple smaller packets. The packet of each fragment is automatically prefixed with an IP header. Incoming packets are reassembled, if necessary, and stripped of the header before being passed on to the next higher protocol layer, usually TCP or UDP. IP does not acknowledge receipt of a packet, nor is it responsible for retransmitting or providing flow and error control. Reliable delivery must be ensured by a higher-level protocol, such as TCP.

Integral to every IP implementation is the Internet Control Message Protocol (ICMP), used for reporting errors in datagram processing. Although ICMP is a basic part of IP, it treats IP as a higher-level protocol.

Address Resolution Protocol

The Address Resolution Protocol (ARP) dynamically maps internet addresses to hardware addresses on a local network.

Transmission Control Protocol

The Transmission Control Protocol (TCP) provides a reliable vehicle for delivering packets between hosts on an internet. TCP takes a stream of data, breaks it into datagrams, sends each one individually using IP, and reassembles the datagrams at the destination node. If any datagrams are lost or damaged during transmission, TCP detects this fact and resends the missing datagrams. The received data stream is a reliable copy of the transmitted data stream.

The interface to TCP is a set of library calls similar to the calls made by an application program to an operating system when manipulating files. TCP communicates asynchronously with applications in a general environment of interconnected networks, and assumes the presence of IP as the underlying protocol at the network level.

User Datagram Protocol

The User Datagram Protocol (UDP) allows application programs to send messages to other programs with a minimum of protocol conversion. Unlike TCP, UDP is a connectionless datagram protocol that requires minimal overhead, but does not guarantee delivery. UDP may be used instead of TCP when an application does not want to incur the overhead of TCP connection setup and breakdown, and is able to do its own acknowledgement and retransmission processing to ensure reliable data transfer. The interface to UDP is a set of library calls.

Other Network Protocols

The user interfaces directly with the Application Protocols layer. This layer consists of several independent protocols that implement the following applications:

File Transfer Protocol
The File Transfer Protocol (FTP) allows copies of files to be sent across an internet. Usually, a password is required to retrieve a file from a foreign host.

Simple Mail Transfer Protocol
The Simple Mail Transfer Protocol (SMTP) allows electronic mail to be exchanged among hosts across an internet.

Remote Terminal Protocol
The Telnet Protocol (Telnet) allows remote logon to hosts across an internet.

Internet Addressing

Each node on an internet must have a unique address called an *internet address.* This address is a 32-bit integer. Internet addresses are usually expressed in the form `www.xxx.yyy.zzz`, where each field is the decimal representation of one octet of the address. For example, the address whose hexadecimal representation is `X'82638001'` would be expressed as `130.99.128.1`. Addresses on the Internet[1] are administered by SRI International[2]. If you have your own internet, you are its administrator. It is strongly recommended that you have an Internet address.

A higher-level naming method called *domain naming* is used to eliminate the need for users to know numerical internet addresses. When a user specifies a domain name, it is resolved into an internet address by a domain name server.

Internet addresses have two parts: a network number and a host number on that network. The four-octet address is divided into network number and host number in any of three different ways, depending on the range of addresses into which the address falls. The first range of addresses is reserved for *Class A* networks, which are very large networks. The second range of addresses is reserved for *Class B* networks, which are medium-size networks. The third range of addresses is reserved for *Class*

---

[1] A specific internetwork that includes ARPANET, MILNET and NSFnet. These networks use the TCP/IP protocol suite.

[2] SRI International, 333 Ravenswood Avenue, Menlo Park, CA. 94025, 1-800-235-3155.

C networks, which are small networks. Some addresses are reserved for special purposes and for future classes of networks.

The division of addresses between network number and host number is as follows:

Table 1-1. Address Fields of Internet Addresses

Network Class	Network Number	Host Number
Class A	First octet	Last three octets
Class B	First two octets	Last two octets
Class C	First three octets	Last octet

A network can be divided into multiple smaller subnetworks called *subnets*. For example, a Class B network can be divided into multiple subnets, each having the same number of hosts as a Class C network. This extension to the IP addressing scheme allows a site to be seen from outside as having one network number. The Class B network could consist of multiple networks, each of smaller size.

To allow for subnet addressing, the host number portion of an internet address is divided into a subnet part and a host part. The network portion and subnet part are logically concatenated to form the subnetwork identifier. For example, if a Class B network had an address of `130.42`, it could be divided into 254 subnetworks with addresses ranging from `130.42.1` through `130.42.254` (0 and 255 are reserved).

On each network, a special host number is reserved for a broadcast address. This is the host number consisting of all 0 bits or all 1 bits. For example, the broadcast host number on a Class B network is 0 or 65 535. The broadcast host number on a Class C network, or on each subnet of a Class B network is 0 or 255.

Routing

Routing is the process of deciding where to send a packet based on its destination address. There are two kinds of routing involved in communications within an internet: direct and indirect.

Direct routing is used when the source and destination nodes are on the same network within the internet. The source node maps the destination internet address into a hardware address and sends packets to the destination node at this address. This mapping is normally performed through a translation table, but if no match can be found for a destination internet address, the Address Resolution Protocol (ARP) provides a default address.

Indirect routing is used when the source and destination nodes are on different networks within the internet. The source node sends packets to a gateway, bridge, or router on the same network using direct routing. From there, the packets are forwarded through intermediate gateways, bridges, or routers, as required, until they arrive at the destination network. Direct routing is then used to forward the packets to the destination host on that network. Each host, gateway, bridge, and router in the internet has a routing table that defines paths to other nodes in the internet.

Example of Using TCP/IP

To send a large file from a source node on one network to a destination node on another network, the source node waits for concurrence from the destination node. In this scenario, the source node is the client and the destination node is the server for the file transfer application (see Figure 1-4).

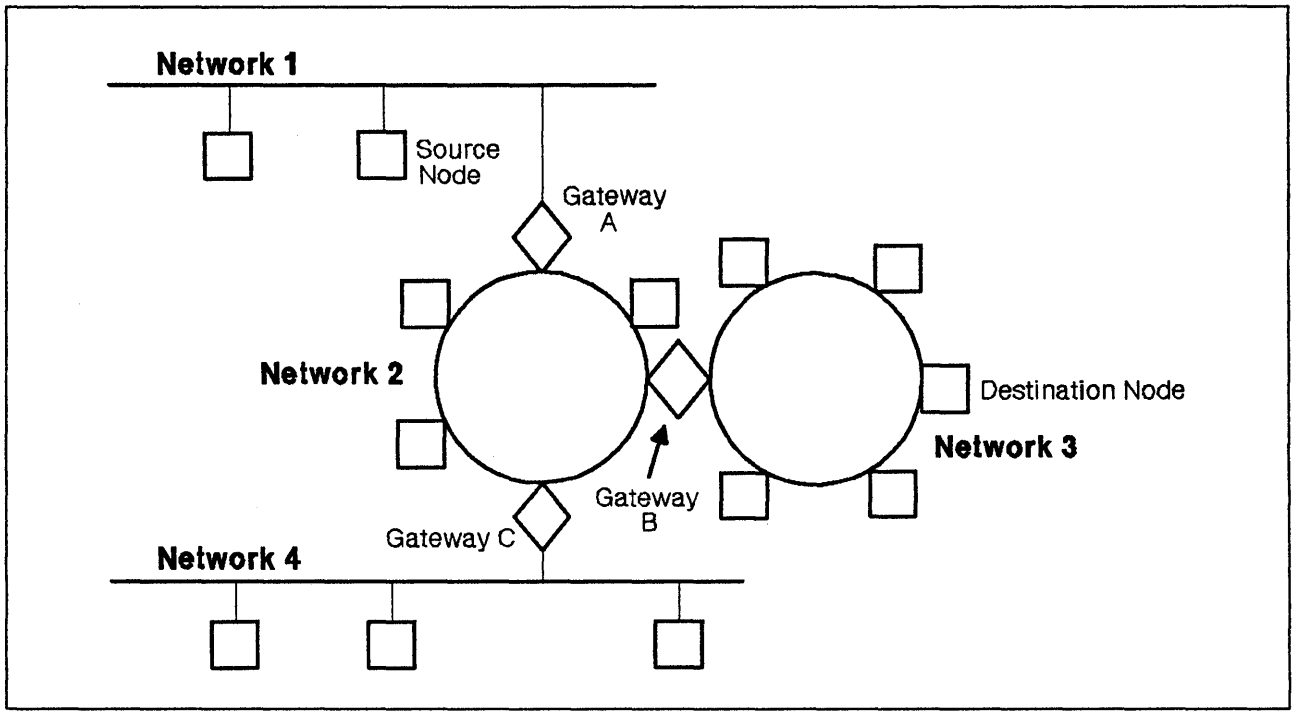


Figure 1-4. File Transfer Using TCP/IP

The file transfer application on the source node takes the file and uses the reliable data stream connection services of TCP to send the file to the file transfer application on the destination node. The source node's TCP layer breaks the file into packets and uses the packet routing services of IP to individually send each packet to TCP on the destination node. Packets do not necessarily arrive in sequence. It is up to the destination node's TCP layer to sort them out and restore the original file. If packets are lost in transmission, the source node resends the lost packets.

The source node's IP layer determines whether the destination node is on the local network or on a foreign network. If the destination node is on the local network, the source node's IP uses the services of the low-level network protocol to send the packet to the destination node's IP layer. If the destination node is on a foreign network, the source node's IP layer decides which gateway should route the datagram and uses the services of the low-level network protocol to send the packet to that gateway's IP layer. The gateway's IP layer in turn routes the packet through the foreign network until the destination node's IP layer is reached.

Chapter 2. Transferring Data Using the File Transfer Protocol

The File Transfer Protocol (FTP) is included in the Transmission Control Protocol/Internet Protocol (TCP/IP) suite to facilitate transfers of data sets or files between hosts that support TCP/IP. Using the FTP command and its subcommands, you can access multiple machines in a single session.

Specifically, FTP allows you to:

- Establish a connection to a foreign host's FTP server
- Identify yourself to the foreign host's FTP server
- Obtain status information about FTP on the foreign host
- List directories belonging to the foreign host
- Transfer data sets or files to and from the foreign host
- Delete or rename data sets or files on the foreign host
- Obtain assistance for the FTP subcommands.

FTP Subcommands

The following table gives a short description of the various FTP subcommands. The table refers you to a page containing additional information on the use of the subcommand. The table also lists the minimum abbreviation for each subcommand. You cannot enter the FTP subcommands unless you are in the FTP environment (see "Entering the FTP Environment" on page 2-3).

Table 2-1 (Page 1 of 2). FTP Subcommands Quick Reference

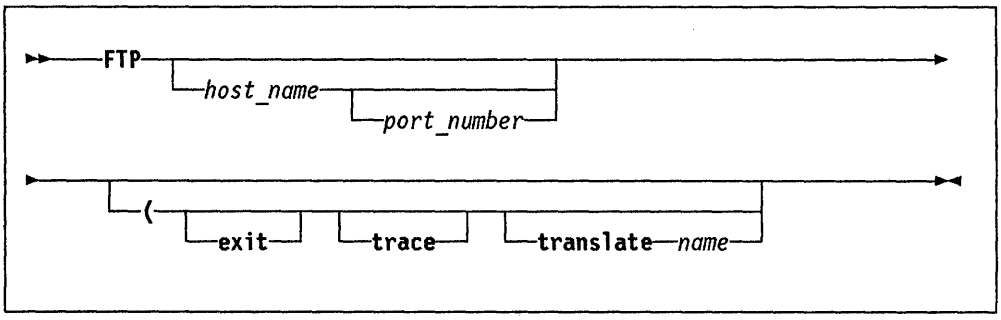
Subcommand	Minimum Abbreviation	Description	Page
ACCT	AC	Sends host dependent account information.	2-7
APPEND	AP	Appends a data set on your local host to a file on the foreign host.	2-18
ASCII	AS	Sets the transfer type to ASCII.	2-17
BINARY	B	Sets the transfer type to IMAGE.	2-18
CD	CD	Changes the working directory.	2-11
CLOSE	CL	Disconnects from the foreign host.	2-27
DELETE	DELE	Deletes a single file on the foreign host.	2-21
DIR	DI	Lists the directory entries for files on the foreign host.	2-13
EBCDIC	EB	Sets the transfer type to EBCDIC.	2-17
GET	G	Copies a file from the foreign host to your local host.	2-19
HELP or ?	H	Obtains assistance for FTP subcommands.	2-5

Table 2-1 (Page 2 of 2). FTP Subcommands Quick Reference

Subcommand	Minimum Abbreviation	Description	Page
LCD	LC	Changes the working directory on the local host.	2-11
LMKDIR	LM	Allocates a Partitioned Data Set (PDS) on the local host.	2-14
LOCSITE	LOCSI	Sets information in the local host.	2-23
LOCSTAT	LOCST	Displays FTP status information for your local host.	2-9
LPWD	LP	Displays the name of the active working directory on the local host.	2-15
LS	LS	Lists the names of files on the foreign host.	2-13
MDELETE	MD	Deletes multiple files on the foreign host.	2-21
MGET	MG	Copies multiple files from the foreign host to your local host.	2-19
MKDIR	MK	Creates a directory in the current foreign directory.	2-14
MODE	MO	Specifies the mode or data format of the transfer.	2-16
MPUT	MP	Copies multiple data sets on your local host to the foreign host.	2-20
NOOP	NO	Checks whether the foreign host is still responding.	2-10
OPEN	O	Opens a connection to a foreign host.	2-6
PASS	PA	Supplies a password to the foreign host.	2-7
PUT	PU	Copies a data set on your local host to the foreign host.	2-20
PWD	PW	Displays the name of the active working directory on the foreign host.	2-15
QUIT	QUI	Leaves the FTP command environment.	2-28
QUOTE	QUO	Sends an uninterpreted string of data.	2-22
RENAME	REN	Renames a file on the foreign host.	2-22
SENDPORT	SENDP	Enables or disables automatic transmission of the FTP server PORT subcommand.	2-26
SENDSITE	SENDS	Enables or disables automatic transmission of the SITE subcommand.	2-26
SITE	SI	Sends information to the foreign host using site-specific commands.	2-23
STATUS	STA	Displays status information of the foreign host.	2-9
SUNIQUE	SU	Toggles storage methods.	2-27
SYSTEM	SY	Displays the name of the foreign host's operating system.	2-9
TSO	TS	Passes a command to your local host's TSO environment.	2-22
TYPE	TY	Specifies the transfer type.	2-16
USER	U	Identifies you to a foreign host.	2-6

Entering the FTP Environment

Prior to transferring files between your local host and the foreign host, you must enter the FTP environment. To do so, issue the FTP command as follows:



Parameter	Description
host_name	Is the name of the foreign host you are connecting to. Specify the foreign host by its host name or its internet address.
port_number	Is a decimal integer identifying a port on the foreign host. By specifying a port, you override the default *port_number* on the foreign host. The default port is the host's FTP server. This parameter is used for system testing only.
exit	Forces FTP to terminate when an error occurs.
trace	Writes the tracing information to the data set pointed to by DD SYSPRINT. Consult your IBM TCP/IP Support Group when using **trace**. This parameter is used for system testing only. Abbreviations for this parameter are accepted (**trac** is the shortest abbreviation accepted).
translate *name*	Specify if you want to use a non-standard translation table. If you specify this parameter, FTP uses the translation table in the *user_id.name*.TCPXLBIN data set instead of the standard translation table provided with the TCP/IP for MVS program (*user_id*.STANDARD.TCPXLBIN or TCPIP.STANDARD.TCPXLBIN). If *user_id.name*.TCPXLBIN does not exist, FTP uses TCPIP.*name*.TCPXLBIN. If *user_id.name*.TCPXLBIN and TCPIP.*name*.TCPXLBIN do not exist, or if they were incorrectly created, FTP terminates with an error message. Abbreviations for this parameter are accepted (**tran** is the shortest abbreviation).

Once you have entered the FTP environment, you see the following prompt on your screen:

```
READY
ftp forest
MVS TCP/IP FTP R1.0
Connecting to FOREST 129.43.11.11, port 21
220 FOREST FTP server (IBM RT) ready.
USER (identify yourself to the host):
sleepy
>>>USER sleepy
331 Password required for sleepy.
Password:
password
>>>PASS ********
230 User sleepy logged in.
Command:
```

Figure 2-1. Example of the Displayed Prompt in the FTP Environment

If you do not specify a *host_name* with the FTP command, you are prompted for one. If you are unable to open a connection to a foreign host by specifying a *host_name* with the FTP command, or if you enter the *host_name* incorrectly, use the OPEN subcommand (see "Opening a Connection to the FTP Server" on page 2-6) to re-try opening the connection.

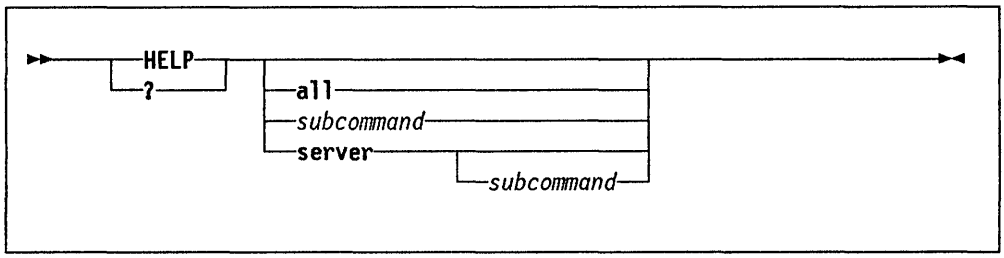
If you specify a *host_name* with the FTP command, you are prompted to identify yourself. After successfully identifying yourself, you are prompted for a password if the foreign host requires it. If you enter the password correctly, you are connected to the foreign host.

Allocating FTP Input and Output Data Sets

When the FTP command is invoked, a check is made to see if a data set is allocated to DD INPUT. If so, subcommands are read from that data set rather than from your terminal. Similarly, a check is also made to see if a data set is allocated to DD OUTPUT. If so, all output from the FTP subcommands is written to that data set rather than to your terminal.

Obtaining Assistance

While in the FTP environment, you may need assistance with the FTP subcommands. Issue the HELP (or ?) subcommand to obtain a description of the FTP subcommands. The format of the HELP (or ?) subcommand is as follows:



Parameter	Description
all	Gives a detailed description of all the FTP subcommands on your local host.
subcommand	Is the name of the FTP subcommand you need help with.
server	Gives the help offered by the foreign host's FTP server. If you specify *subcommand* with the **server** parameter, you receive the help information about the subcommand offered by the foreign host's FTP server. You can use this parameter only if you have an established connection to the foreign host.

If you issue the HELP (or ?) subcommand without a parameter, a list of subcommands and a general description of the help information available appears.

Establishing a Connection to an FTP Server

After you have entered the FTP environment, you may want to establish a connection to a different foreign host than the one you specified with the FTP command. Follow the steps below to establish this connection.

Note: Normally, you do not need to perform the steps below and, consequently, you do not need the information in this section. In most cases, you specify the foreign host to which you want to establish a connection with the FTP command. At that time, you are prompted by the foreign host to supply the required information (user ID, password).

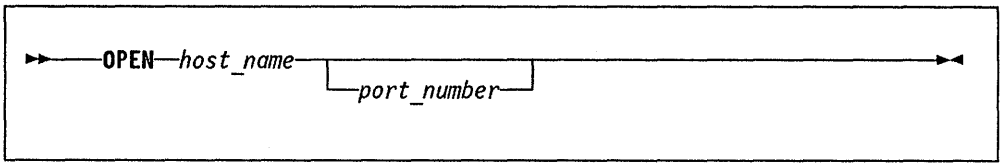
1. Close the existing connection
2. Open the connection
3. Identify yourself to the foreign host
4. Supply a password (if required) to the foreign host
5. Supply account information (if required) to the foreign host.

Opening a Connection to the FTP Server

Use the OPEN subcommand to open a connection to the foreign host's FTP server. Use the OPEN subcommand in the following situations:

- If, after closing a connection (see "Disconnecting From a Foreign Host" on page 2-27), you want to open another connection without leaving the FTP environment.
- If you are unable to open a connection when you specified a *host_name* with the FTP command. This situation occurs if you did not enter the *host_name* correctly with the FTP command, if network problems were being experienced, or if the foreign host was down at the time you issued the FTP command.

The format for the OPEN subcommand is as follows:



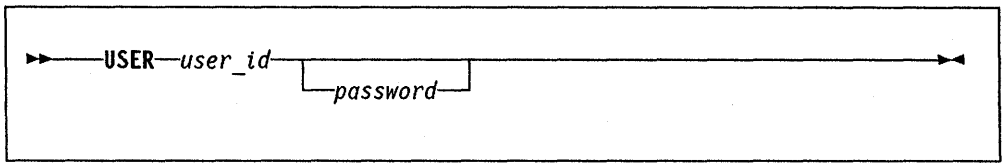
Parameter	Description
host_name	Is the name of the foreign host you are opening a connection to. Specify the foreign host by its host name or its internet address.
port_number	Is a decimal integer identifying a port on the foreign host. By specifying a port, you override the default *port_number* on the foreign host. The default port is the host's FTP server. This parameter is used for system testing only.

If you specify no parameters with the OPEN subcommand, you are prompted for the host name.

If you issue the OPEN subcommand when a connection already exists, the OPEN subcommand fails. Issue the CLOSE subcommand (see "Disconnecting From a Foreign Host" on page 2-27) to close the connection before attempting to open a new connection.

Identifying Yourself to the FTP Server

You must identify yourself to the foreign host after opening a connection to it. Some hosts prompt you for identification. To identify yourself to the foreign host, issue the USER subcommand as follows:



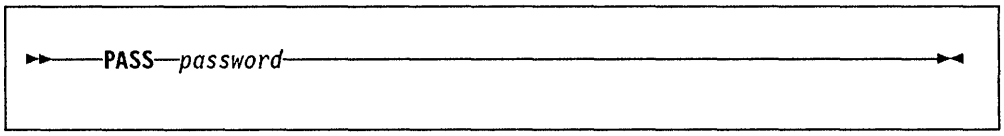
Parameter	Description
user_id	Is the name that identifies you to the foreign host you are establishing a connection to. This name is also the name that you would use to log on to the foreign host using the TELNET command (see Chapter 4, "Logging on to a Host" on page 4-1). If the foreign host does not require a user ID, specify **anonymous** for *user_id*.
password	Is the password associated with the *user_id* on the foreign host. This parameter may not be required on some hosts.

If you do not supply your *password* when invoking the USER subcommand, you are prompted to do so if it is required by the foreign host. If you enter your password incorrectly, you are not prompted to enter it again. Re-issue the USER subcommand to enter the password correctly.

Supplying a Password to the FTP Server

Some hosts may require a password after you identify yourself to validate that you are who you claim to be. Supply this password with the USER subcommand (see "Identifying Yourself to the FTP Server" on page 2-6), or with the PASS subcommand. Some implementations prompt you for a password after you enter the USER subcommand without specifying the *password* parameter. Where you are not prompted for the password, use the PASS subcommand to supply it.

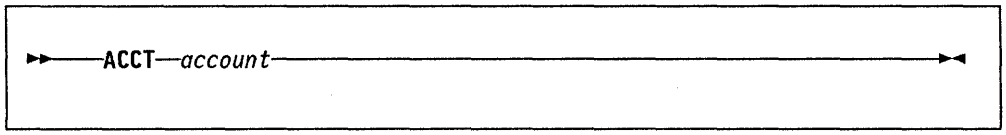
If you enter your password incorrectly, re-issue the USER subcommand (see "Identifying Yourself to the FTP Server" on page 2-6). The format of the PASS subcommand is as follows:



Parameter	Description
password	Is the password associated with the *user_id* on the foreign host you are establishing a connection to. This parameter may not be required on some hosts.

Supplying Account Information to the FTP Server

Some hosts require account information before they allow you to access files. If the foreign host requires account information, and you are not specifically prompted for it, use the ACCT subcommand to send this information to the foreign host as follows:



Parameter	Description
account	Is a string identifying the account information that is required by the foreign host.

For example, you may have to use the ACCT subcommand when the foreign host is running the TCP/IP for VM program. The foreign VM host may require passwords for read and write access to its minidisks. If you are not prompted by the foreign VM host for the passwords, use the ACCT subcommand to send these passwords to the foreign VM host.

The TCP/IP for MVS FTP server does not require account information.

Example of a Typical FTP Session

The following figure is an example of a typical FTP session. Some of the subcommands presented in this example are discussed later in this chapter.

```
READY
ftp forest
MVS TCP/IP FTP R1.0
Connecting to FOREST 129.43.11.11, port 21
220 FOREST FTP server (IBM RT) ready.
USER (identify yourself to the host):
sneezy
>>>USER sneezy
331 Password required for sneezy.
Password:
password
>>>PASS ********
230 User sneezy logged in.
Command:
ls
>>>PORT 129,43,11,2,3,249
200 PORT command successful.
>>>NLST
150 Opening data connection for /bin/ls (129.43.11.2,1017) (0 bytes).
witch.letter
fairy.letter
warlock.letter
dwarf.letter
226 Transfer complete.
```

```
Command:
get witch.letter witch.script
>>>PORT 129,43,11,2,3,251
200 PORT command successful.
>>>RETR witch.letter
150 Opening data connection for witch.letter (129.43.11.2,1019)
226 Transfer complete.
3089 bytes transferred. Transfer rate 5.82 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
READY
```

Figure 2-2. Example of a Typical FTP Session

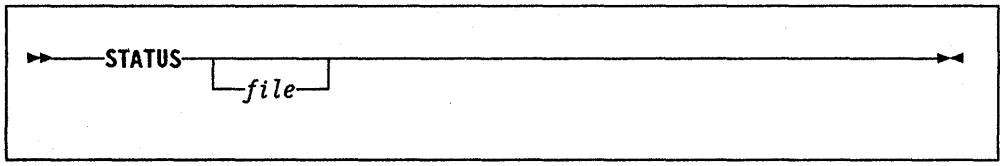
Obtaining Status and System Information

You can use FTP subcommands to obtain status and system information:

- Use STATUS to retrieve status information from the foreign host
- Use SYSTEM to display the name of the foreign host's operating system
- Use LOCSTAT to display FTP status information about your local host
- Use NOOP to inquire if the foreign host is still responding.

Retrieving Status Information From the Foreign Host

Use the STATUS subcommand to retrieve status information from the foreign host. You can retrieve status information on a file or general status information, such as a summary of activity. The format of the STATUS subcommand is as follows:

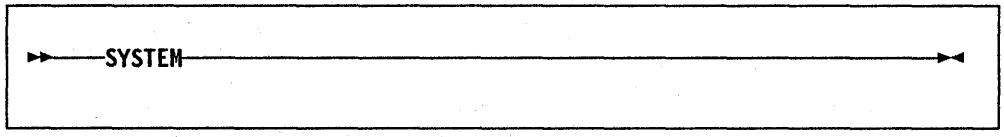


Parameter	Description
file	Is a directory or file on the foreign host whose status information you want. If you omit *file*, you receive general status information. The TCP/IP for MVS FTP server does not support *file*.

The status information that you receive is dependent on the foreign host.

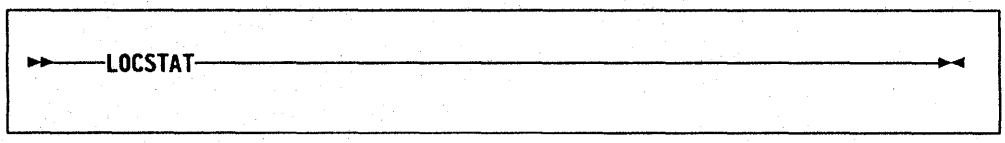
Displaying the Operating System Name

Use the SYSTEM subcommand to display the name of the of the foreign host's operating system. The format of the SYSTEM subcommand is as follows:



Displaying FTP Status Information

Use the LOCSTAT subcommand to display status information about FTP on your local host. The format of the LOCSTAT subcommand is as follows:

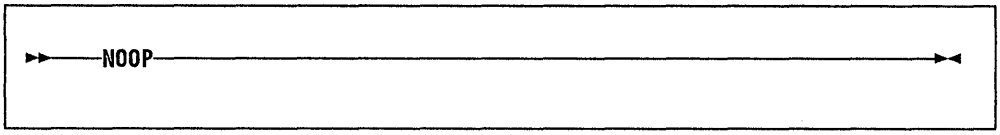


The following information is displayed:

- The value of the TRACE flag (set using the FTP command)
- The SENDPORT setting (true or false)
- The SENDSITE setting (true or false)
- The name, port number, and logon status of the foreign host
- The port number of your local host
- The FTP data type (ASCII, EBCDIC, Image) and transfer mode (Block, Stream)
- The SUNIQUE setting (on or off).

Determining if the Foreign Host is Responding

Use the NOOP subcommand to determine if the foreign host is still responding. The format of the NOOP subcommand is as follows:



If the foreign host is responding, you receive a response of 'OK'. If it is not, you receive a response of 'Host is not responding'. A host does not respond when a communication failure has occurred. It may take a few minutes for 'Host is not responding' to be displayed. Following this response, you are prompted for the next FTP subcommand. If you specified the **exit** parameter with the FTP command (see "Entering the FTP Environment" on page 2-3) and an error occurs, FTP terminates.

Working With Directories

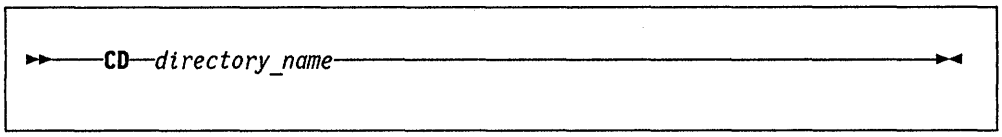
Some FTP subcommands allow you to perform the following tasks when working with directories:

- CD changes the working directory or file group on the foreign host.
- LCD changes the working directory or file group on the local host.
- DIR displays a list of directory entries or a list of files in a file group on the foreign host.
- MKDIR creates a directory in the current working directory on the foreign host.
- LMKDIR allocates a PDS on the local host.
- LS displays a list of the file names in the working directory or file group on the foreign host.
- PWD displays the name of the working directory on the foreign host.
- LPWD displays the name of the working directory on the local host.

Changing the Current Working Directory

On the Foreign Host

Use the CD subcommand to change the current working directory or file group on the foreign host. The format of the CD subcommand is as follows:



Parameter	Description
directory_name	Is the name of a file directory, a fully qualified data set, or a prefix on the foreign host.

If the foreign host is using the TCP/IP for MVS program, the *directory_name* identifies either a common prefix for a group of data sets or the qualifier(s) of a PDS. By default, the CD subcommand considers the *directory_name* to be a PDS qualifier.

If the *directory_name* is a common prefix qualifier for a group of data sets, end the data set specification with a period (.) as follows:

qualifier1.qualifier2....qualifiern.

For example,

jones.source.

appends 'jones.source' to the existing working directory.

The high-level qualifiers of the *directory_name* are supplied by the system unless the *directory_name* is contained within single quotes ('). The *directory_name* contained within single quotes is considered to be fully qualified.

Remove the lowest-level of the current working directory as follows:

CD ..

Note: If you have a partitioned data set with a name that is also the beginning qualifiers of a list of data sets, CD defaults to the PDS. To specify a list of data sets, add a period (.) after the last qualifier.

The MVS FTP server establishes the user ID supplied by the USER subcommand (see "Identifying Yourself to the FTP Server" on page 2-6) as the default working directory. If the directory is the name of a PDS, all file names refer to members of that PDS.

If the foreign host is using the TCP/IP for VM program, specify the *directory_name* as follows:

userid [*minidisk_address*]

OR

userid [*.minidisk_address*]

where the [and the] delimit optional information. For example:

jones 191

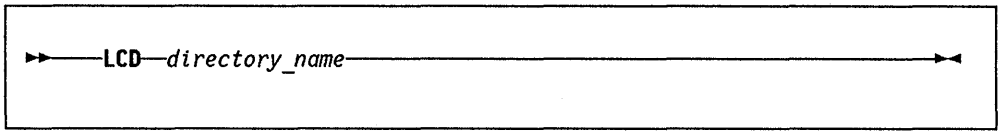
OR

jones.191

where 191 is the default minidisk address.

On the Local Host

Use the LCD subcommand to change the current working directory on the local host. The format of the LCD subcommand is as follows:



Parameter	Description
directory_name	Is the name of a fully qualified data set or a prefix on the local host.

The *directory_name* identifies either a common prefix for a group of data sets or the qualifier(s) of a PDS. By default, the LCD subcommand considers the *directory_name* to be a PDS qualifier.

If the *directory_name* is a common prefix qualifier for a group of data sets, end the data set specification with a period (.) as follows:

qualifier1.qualifier2....qualifiern.

For example,

jones.source.

appends 'jones.source' to the existing local directory name.

The high-level qualifiers of the *directory_name* are supplied by the system unless the *directory_name* is contained within single quotes ('). The *directory_name* contained within single quotes is considered to be fully qualified.

Remove the lowest-level of the current working directory as follows:

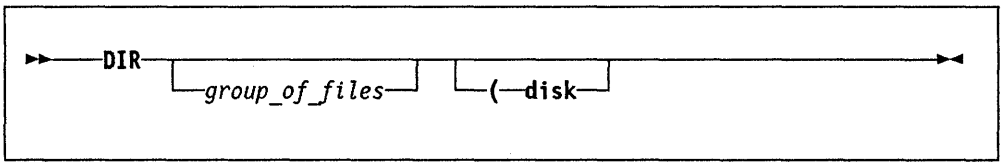
```
LCD ..
```

Note: If you have a partitioned data set with a name that is also the beginning qualifiers of a list of data sets, LCD will default to the PDS. To specify a list of data sets, add a period (.) after the last qualifier.

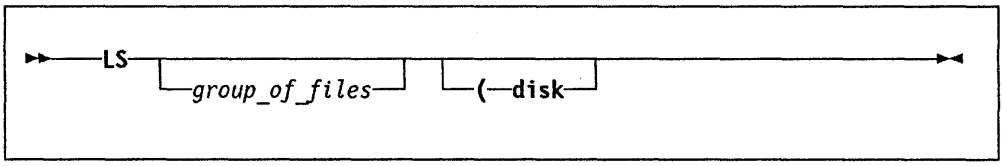
Listing the Directory Entries

Use the DIR and LS subcommands to display the contents of the foreign host's current directory or file group. Use the DIR subcommand to get a complete list of the directory or file group entries, with auxiliary information about the entries. Use the LS subcommand to get a list containing only the file names in the current directory or file group.

The format of the DIR subcommand is as follows:



The format of the LS subcommand is as follows:



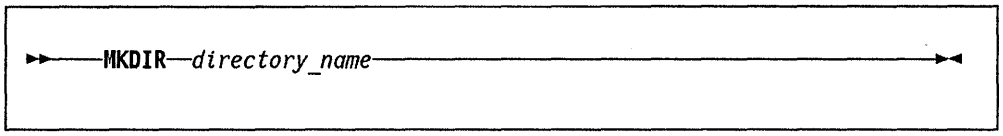
Parameter	Description
group_of_files	Is the entry name in the foreign host's current directory or file group to be listed. You can use special characters for pattern matching when specifying the *group_of_files*. These characters are dependent on the foreign host's FTP server. Refer to "Specifying Data Sets and Files" on page vi for more information on pattern matching and on specifying data sets and files.
disk	For the DIR subcommand, **disk** specifies that the output of the subcommand is stored in the data set called *user_id*.DIROUTP, instead of being displayed on the screen. For the LS subcommand, **disk** specifies that the output of the subcommand is stored in the data set called *user_id*.LSOUTPUT, instead of being displayed on the screen.

If a *group_of_files* is not specified, all the entries in the current directory or file group are displayed. Make a directory or file group current with the CD subcommand (see "Changing the Current Working Directory" on page 2-11).

If the foreign host is using the TCP/IP for MVS program and the current working directory on that host is a PDS, the DIR and LS subcommands give you a list of PDS members.

Creating a Directory on the Foreign Host

Use the MKDIR subcommand to create a directory on the foreign host. The format of the MKDIR subcommand is as follows:

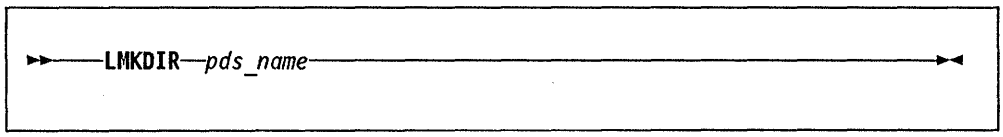


Parameter	Description
directory_name	Is the name of the directory to be created.

The MKDIR subcommand sends a request to the foreign host's FTP server to create a directory with name *directory_name* in the current foreign directory. When the request is sent to a foreign MVS server, a PDS is created with either a fully qualified name (if *directory_name* is specified in single quotes), or appended to the existing foreign directory name.

Creating a PDS on the Local Host

Use the LMKDIR subcommand to create a PDS on the local host. The format of the LMKDIR subcommand is as follows:



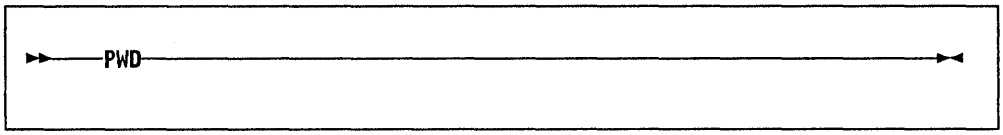
Parameter	Description
pds_name	Is the name of the PDS to be created.

A PDS is created with either the fully qualified name (if *pds_name* is specified in single quotes), or appended to the existing local directory name. The FTP client will use the allocation parameters specified with the LOCSITE subcommand (see "Sending Site-Dependent Information to a Host" on page 2-23).

Displaying the Name of the Current Directory

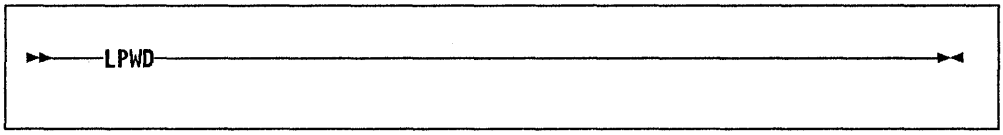
On the Foreign Host

Use the PWD subcommand to display the name of the current directory on the foreign host. The format of the PWD subcommand is as follows:



On the Local Host

Use the LPWD subcommand to display the name of the current directory on the local host. The format of the LPWD subcommand is as follows:



Transferring Data

Some FTP subcommands allow you to perform the following tasks with data sets or files:

- MODE sets the format in which the data is transferred.
- TYPE, ASCII, EBCDIC, and BINARY specify the transfer type to be used during the data transfer.
- APPEND appends a data set on your local host to a file on the foreign host.
- GET and MGET retrieve files from the foreign host.
- PUT and MPUT send data sets to the foreign host.
- DELETE and MDELETE delete files on the foreign host.
- RENAME renames files on the foreign host.

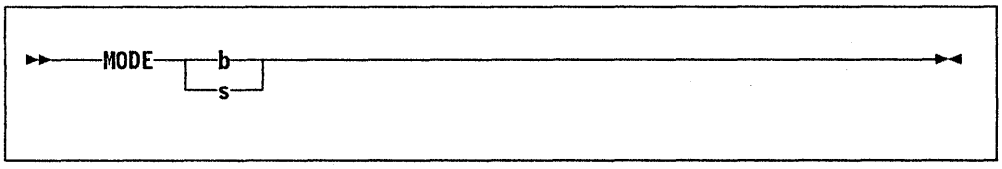
Data Transfer Methods

The appropriate transmission attributes must be used to transfer files between two hosts to preserve the content and structure of the files. Table 2-2 on page 2-16 suggests how these attributes should be set for different host systems. A text file contains standard, displayable characters only, specifically not including the carriage return (ASCII X'0D' and EBCDIC X'0D') or line feed (ASCII X'0A' and EBCDIC X'25') characters. A binary file may contain any character.

Table 2-2. Recommended Methods for Data Transfer		
Transfer Between Host Types	**Transfer Type**	**Mode**
MVS or VM to MVS or VM – text data	EBCDIC	Stream
MVS or VM to MVS or VM – binary data	EBCDIC	Block
MVS or VM to ASCII – text data	ASCII	Stream
ASCII to MVS or VM – text data	ASCII	Stream
ASCII to MVS or VM to ASCII – all data **Note:** The MVS (or VM) host is used for storage only. Data is not used on the MVS (or VM) host.	Binary (Image)	Stream

Specifying the Mode of Data Transfer

Use the MODE subcommand to specify the mode, or data format, in which the data transfer is to take place. The MODE subcommand defines how the bits of data are to be transmitted. It is used as follows:



Parameter	Description
b	Specifies block mode. The data is transmitted as a series of data blocks, preceded by one or more header bytes. Block mode allows the transfer of binary data and preserves the logical record boundaries of the file.
s	Specifies stream mode. Any representation type can be used with stream mode. Stream mode is the default transfer mode. This transfer mode is more efficient because no data block information is transferred.

Setting the Data Transfer Type

Depending on the data representation supported by the foreign host to which you are sending data, you may have to change the data transfer type. FTP supports three data transfer types:

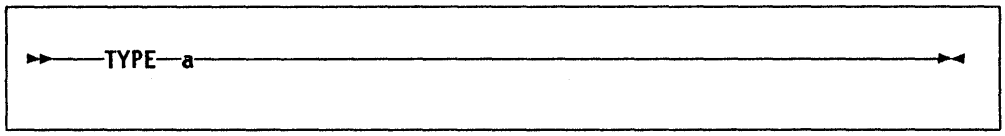
- ASCII
- EBCDIC
- Binary (Image).

Use the TYPE, ASCII, EBCDIC, and BINARY subcommands to set this transfer type. ASCII is the default transfer type.

Setting the Transfer Type to ASCII

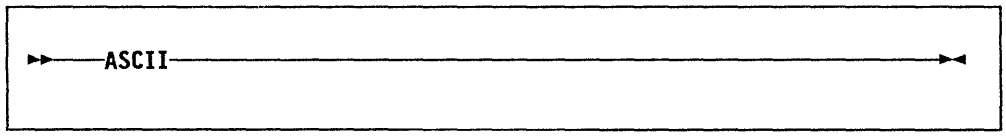
Use the TYPE subcommand with the **a** parameter or the ASCII subcommand to set the data transfer type to ASCII. ASCII is the default transfer type.

The format of the TYPE subcommand is as follows:



Parameter	Description
a	Specifies the transfer type as ASCII. A conversion from EBCDIC to ASCII is done in the MVS host before the data is transmitted or as soon as the data is received. A transfer type of ASCII is intended primarily for the transfer of text files to or from an ASCII host whose FTP components do not support the EBCDIC data representation.

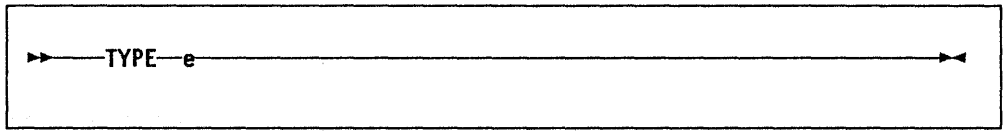
The format of the ASCII subcommand is as follows:



Setting the Transfer Type to EBCDIC

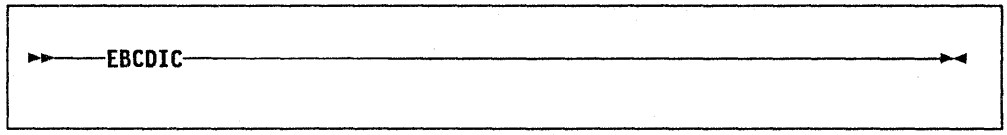
Use the TYPE subcommand with the **e** parameter or the EBCDIC subcommand to set the data transfer type to EBCDIC.

The format of the TYPE subcommand is as follows:



Parameter	Description
e	Specifies the transfer type as EBCDIC. The EBCDIC transfer type is intended for efficient transfer between hosts that use EBCDIC for their internal character representation.

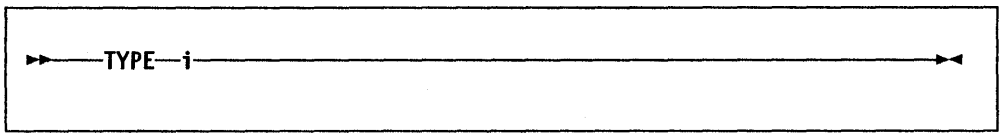
The format of the EBCDIC subcommand is as follows:



Setting the Transfer Type to Binary (Image)

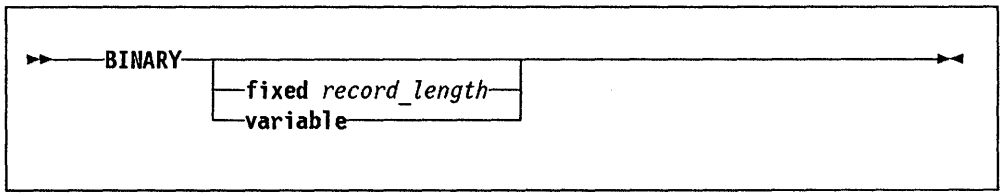
Use the TYPE subcommand with the **i** parameter or the BINARY subcommand to set the data transfer type to binary (image).

The format of the TYPE subcommand is as follows:



Parameter	Description
i	Specifies the transfer type as binary (image). With the binary transfer type, data is sent as contiguous bits, packed into 8-bit bytes. The binary transfer type is intended for the efficient storage and retrieval of files and for the transfer of binary data.

The format of the BINARY subcommand is as follows:

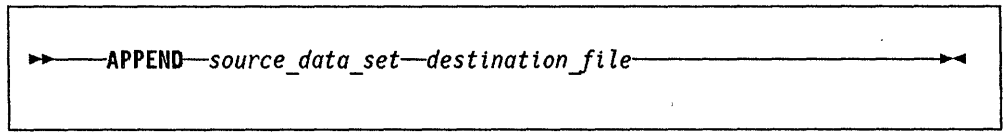


Parameter	Description
fixed *record_length*	Specifies that files are stored in fixed record format with records of length *record_length*.
variable	Specifies that data is stored in variable record format. This is the default.

The BINARY subcommand with these parameters can be used before the GET and the MGET subcommands (see "Copying Files to Your Local Host" on page 2-19) to specify how binary data is stored. Refer to "Sending Site-Dependent Information to a Host" on page 2-23 for specifications on storing binary files on a host.

Appending a Data Set to a File on the Foreign Host

Use the APPEND subcommand to append a data set on your local host to a file on the foreign host. The format of the APPEND subcommand is as follows:



Parameter	Description
source_data_set	Is the name of the data set on your local host to be appended.
destination_file	Is the name of the file on the foreign host to which your data set will be appended.

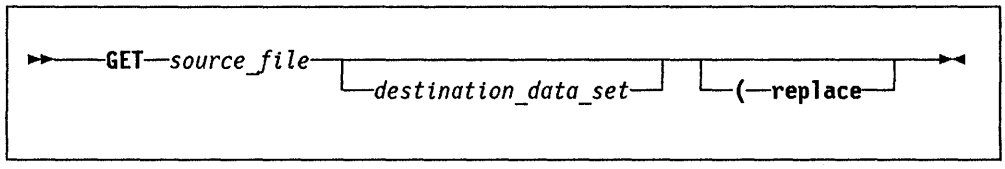
If the foreign host is an MVS or a VM host and if the file has a fixed-record format, the file format and record length of the file are always preserved. Records from the data set on your local host can be truncated or padded with blanks when necessary.

To append to a file on a foreign host, you must have a defined working directory on that host (see "Changing the Current Working Directory" on page 2-11), and you must have write privileges to the files in this working directory (see "Supplying Account Information to the FTP Server" on page 2-7).

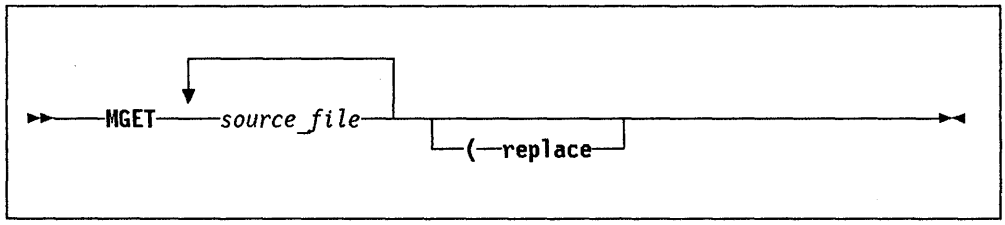
Refer to "Specifying Data Sets and Files" on page vi for more information on specifying data sets and files.

Copying Files to Your Local Host

Use the GET subcommand to copy a file from the foreign host to your local host. The format of the GET subcommand is as follows:



Use the MGET subcommand to copy multiple files from foreign host to your local host. The format of the MGET subcommand is as follows:



Parameter	Description
source_file	Is the name of the file to be retrieved from the foreign host.
destination_data_set	Is the name of the data set on your local host that will be created as a result of the GET subcommand. The last qualifier of the data set will be the same as the file name on the foreign host if you use the MGET subcommand, or if you use the GET subcommand without specifying a *destination_data_set*.

replace Causes a data set on your local host to be overwritten if it already exists. If the data set already exists, and you do not use the **replace** parameter, the existing data set is not overwritten. A message informing you of this is displayed.

If the name of a file is not acceptable to your local host, the file is not transferred. To get a file from the foreign host, you must have a defined working directory on that host (see "Changing the Current Working Directory" on page 2-11) and you must have read privileges to the files in this working directory (see "Supplying Account Information to the FTP Server" on page 2-7).

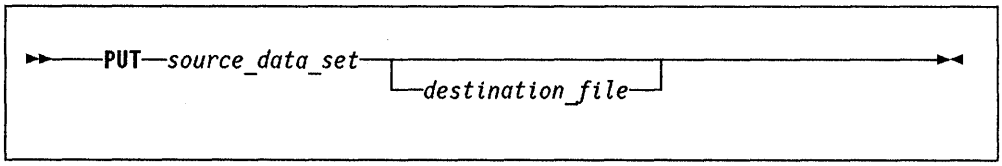
Because more than one file can be copied with the MGET subcommand, the *source_file* parameter of the MGET subcommand can be repeated many times, with each *source_file* delimited by a blank.

You can use special characters for pattern matching when specifying the *source_file* with the MGET subcommand. These characters are dependent on the foreign host's FTP server.

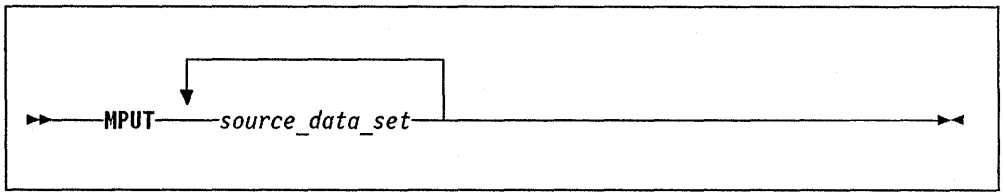
Refer to "Specifying Data Sets and Files" on page vi for more information on pattern matching and on specifying data sets and files.

Copying Data Sets to the Foreign Host

Use the PUT subcommand to copy a file from your local host to the foreign host. The format of the PUT subcommand is as follows:



Use the MPUT subcommand to copy multiple files from your local host to the foreign host. The format of the MPUT subcommand is as follows:



Parameter	Description
source_data_set	Is the name of the data set on your local host being sent to the foreign host.
destination_file	Is the name of the file that the delivered data set will be given on the foreign host. The default *destination_file* name is the same as the *source_data_set* name.

By default, if you use the PUT subcommand and the foreign host already has a file with the name *destination_file*, the foreign host overwrites the existing file. If the foreign host does not have a file with the name *destination_file*, the foreign host creates a new file.

By default, if you use the MPUT subcommand, the foreign host creates files with the same names as the *source_data_set* names and overwrites any files that already exist with those names.

The storage method determines if an existing file on the foreign host is overwritten. By default, existing files are overwritten. Use the SUNIQUE subcommand to change the storage method (refer to "Changing the Storage Method" on page 2-27 for information on changing the storage method on the foreign host).

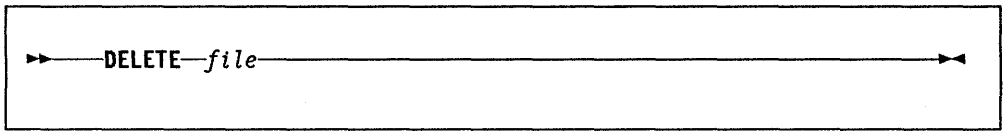
Because more than one data set can be copied with the MPUT subcommand, the *source_data_set* parameter of the MPUT subcommand can be repeated many times, with each *source_data_set* delimited by a blank.

You can use the asterisk (*) character for pattern matching when specifying the *source_data_set* with the MPUT subcommand.

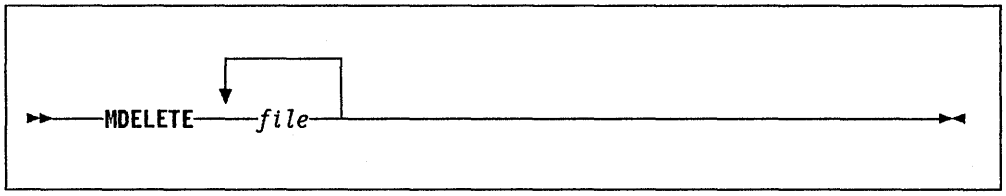
Refer to "Specifying Data Sets and Files" on page vi for more information on pattern matching and on specifying data sets and files.

Deleting Files on the Foreign Host

Use the DELETE subcommand to delete a file on the foreign host. The format of the DELETE subcommand is as follows:



Use the MDELETE subcommand to delete multiple files on the foreign host. The format of the MDELETE subcommand is as follows:



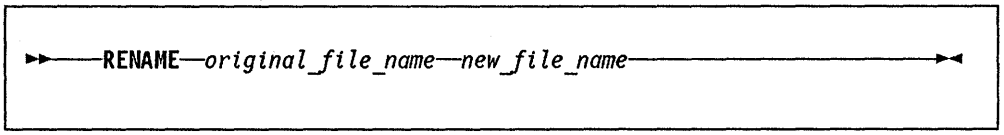
Parameter	Description
file	Is the name of the file to be deleted on the foreign host.

Because more than one file can be deleted with the MDELETE subcommand, the *file* parameter of the MDELETE subcommand can be repeated many times, with each *file* delimited by a blank.

Special characters for pattern matching can be used when specifying the *file* with the MDELETE subcommand. These characters are dependent on the foreign host's FTP server.

Renaming Files on the Foreign Host

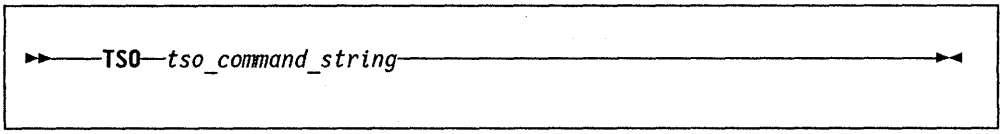
Use the RENAME subcommand to rename a file on the foreign host. The format of the RENAME subcommand is as follows:



Parameter	Description
original_file_name	Is the present name of the file on the foreign host.
new_file_name	Is the new name for the file. If the file specified by *new_file_name* already exists, a message indicating this is displayed, and the file is not renamed.

Passing TSO Commands to Your Local Host

Use the TSO subcommand to pass a command to your local host's Time Sharing Option (TSO) environment. The format of the TSO subcommand is as follows:

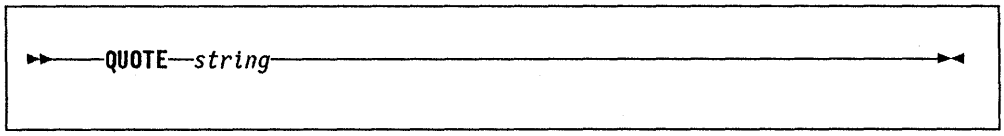


Parameter	Description
tso_command_string	Is a TSO command and its parameters.

Note: The use of other TCP/IP commands from within FTP are not allowed (unpredictable results).

Sending an Uninterpreted String of Data

Use the QUOTE subcommand to send an uninterpreted string of data to the foreign host's FTP server. The format of the QUOTE subcommand is as follows:



Parameter	Description
string	Is the data string to be sent to the foreign host's FTP server.

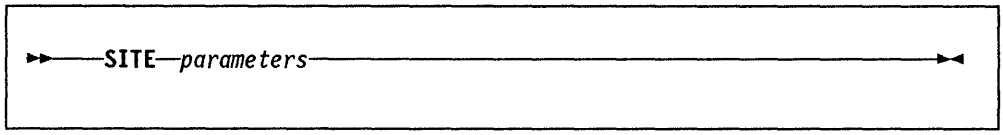
The QUOTE subcommand bypasses the FTP interface of your local host. Use the QUOTE subcommand to send commands that can be understood by the foreign host's FTP server, but not by the the FTP interface on your local host.

Note: The use of the QUOTE subcommand can cause synchronization errors between the two ends of an FTP connection. Be cautious when using the QUOTE subcommand so that you do not send commands to the other foreign host's FTP server that could affect the synchronization of the FTP connection.

Sending Site-Dependent Information to a Host

To the Foreign Host

Use the SITE subcommand to send information that the foreign host will use to provide services specific to that host system. Issue the HELP **server site** subcommand (see "Obtaining Assistance" on page 2-5) to display a list of the types of services (and their correct syntax) available on the foreign host. The format of the SITE subcommand is as follows:



Parameter	Description
parameters	Are parameters dependent on the foreign host system. The SITE subcommand sends this information immediately.

If the foreign host is using the TCP/IP for MVS program, use the SITE subcommand to send data set allocation attributes to the foreign host. The site-dependent information sent with the SITE subcommand remains active until you issue a new SITE subcommand. The latter SITE subcommand adds to, or changes, the attributes established by previous SITE subcommands. If you send one or more incorrect parameters with the SITE subcommand, an error message, specifying the incorrect parameter(s), is displayed. All correct parameters are set regardless of any incorrect parameter(s) and do not need to be re-issued.

If the foreign host is using the TCP/IP for MVS program, you can send the *parameters* in the following format:

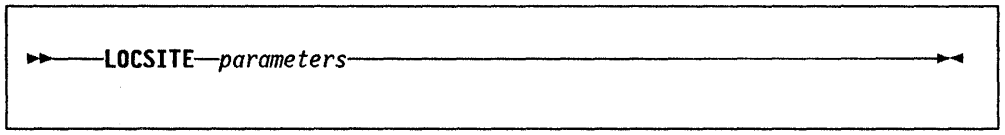
ATTRIBUTE = *value*

Table 2-3 on page 2-24 lists the possible attributes that you can send with the SITE subcommand if the foreign host is using the TCP/IP for MVS program. The table contains a description of each attribute. The possible corresponding values for these attributes are also listed (in the table, each possible *value* is separated by a comma).

The attributes that do not have a corresponding *value* field should not be followed by an equal (=) sign.

To the Local Host

Use the LOCSITE subcommand to specify information that will be used by the local host to provide services specific to that host system. Issue the HELP **locsite** subcommand (see "Obtaining Assistance" on page 2-5) to display a list of the types of services (and their correct syntax) available on the local host. The format of the LOCSITE subcommand is as follows:



Parameter	Description
parameters	Are parameters dependent on the local host system. Separate each of the parameters by blanks.

The site-dependent information set with the LOCSITE subcommand remains active until you issue a new LOCSITE subcommand. The latter LOCSITE subcommand adds to, or changes, the attributes established by previous LOCSITE subcommands. If you specify one or more incorrect parameters with the LOCSITE subcommand, an error message, specifying the incorrect parameter(s), is displayed. All correct parameters are set regardless of any incorrect parameter(s) and do not need to be re-issued.

Specify the *parameters* in the following format:

ATTRIBUTE = *value*

Table 2-3 lists the possible attributes that you can specify with the LOCSITE subcommand. The table contains a description of each attribute. The possible corresponding values for these attributes are also listed (in the table, each possible *value* is separated by a comma). The attributes that do not have a corresponding *value* field should not be followed by an equal (=) sign.

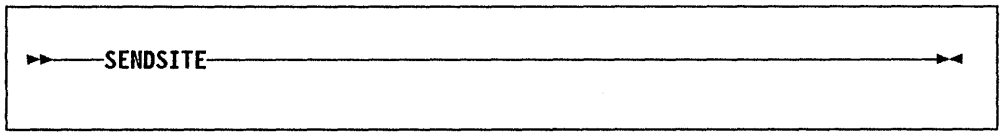
Table 2-3 (Page 1 of 2). SITE and LOCSITE Parameters for MVS Hosts

Attribute	Possible Values	Description
AUTORECALL		Permits migrated Hierarchical Storage Manager (HSM) data sets to be recalled automatically.
NOAUTORECALL		Prevents migrated HSM data sets from being recalled automatically.
AUTOMOUNT		Permits data sets on volumes that are not mounted to be mounted automatically.

Table 2-3 (Page 2 of 2). SITE and LOCSITE Parameters for MVS Hosts		
Attribute	**Possible Values**	**Description**
NOTAUTOMOUNT		Prevents data sets on volumes that are not mounted from being mounted automatically.
DIRECTORYMODE		Specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. In directory mode, this qualifier is the only one used by the MPUT, MGET, LS, and DIR subcommands.
DATASETMODE		Specifies that all the data set qualifiers below the current directory are treated as entries in the directory (disables DIRECTORYMODE).
PRIMARY	*integer*	Specifies the amount of primary allocation.
SECONDARY	*integer*	Specifies the amount of secondary allocation.
DIRECTORY	*integer*	Specifies the number of directory blocks to be allocated for the directory of a PDS.
RECFM	F, FB, FBA, V, VB, VBA, VBS	Specifies the record format of a data set.
BLOCKSIZE	*integer*	Specifies the block size of a data set.
BLOCKS		Specifies that primary and secondary allocations are in blocks.
LRECL	*integer*	Specifies the record size in a data set.
DCBDSN	*data_set_name*	Specifies a data set to be used as a model for a new allocation.
VOLUME	*volume_serial*	Specifies the volume serial number for allocation.
QDISK	*volume_serial*	Displays statistics about available space on a specific volume or on a list of volumes.
TRACKS		Specifies that primary and secondary allocations are in tracks.
CYLINDERS		Specifies that primary and secondary allocations are in cylinders.

Enabling and Disabling the Sending of Site Information

Use the SENDSITE subcommand to enable or disable the automatic sending of file or data set characteristics preceding a data transfer to the foreign host's FTP server. This site information is sent using the SITE subcommand (see "Sending Site-Dependent Information to a Host" on page 2-23). The format of the SENDSITE subcommand is as follows:



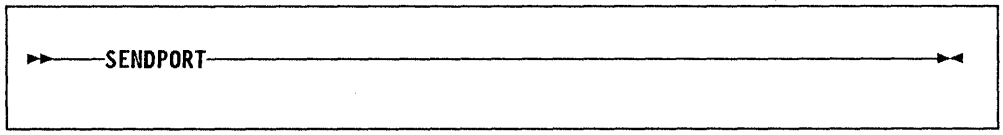
The default is to send the site information automatically.

SENDSITE is a "toggle" subcommand. This means that the SENDSITE subcommand changes the existing setting for the sending of site information. For example, if the sending of site information is enabled, issuing a SENDSITE subcommand disables the sending of site information.

To determine if the sending of site information is enabled or disabled on your local host, use the LOCSTAT subcommand (see "Displaying FTP Status Information" on page 2-9).

Enabling and Disabling the Sending of Port Information

Use the SENDPORT subcommand to enable or disable the sending of port information to the foreign host's FTP server. This information indicates to the foreign host the port number on your local host that is used for data transfer. The format of the SENDPORT subcommand is as follows:



FTP normally sends port information when establishing each data transfer.

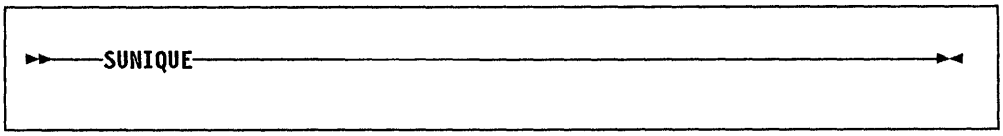
SENDPORT is a "toggle" subcommand. This means that the SENDPORT subcommand changes the existing setting for the sending of port information. For example, if the sending of port information is enabled, issuing a the SENDPORT subcommand disables the sending of port information.

The SENDPORT subcommand is useful for disabling the sending of port information in communications with FTP implementations that ignore port information, while indicating incorrectly that the information has been accepted.

To determine if the sending of port information is enabled or disabled on your local host, use the LOCSTAT subcommand (see "Displaying FTP Status Information" on page 2-9).

Changing the Storage Method

Use the SUNIQUE subcommand to change the storage method used when copying data to the foreign host. The format of the SUNIQUE subcommand is as follows:



By default, SUNIQUE is set off. This means that FTP overwrites an existing file on the foreign host when a PUT or MPUT subcommand (see "Copying Data Sets to the Foreign Host" on page 2-20) is issued.

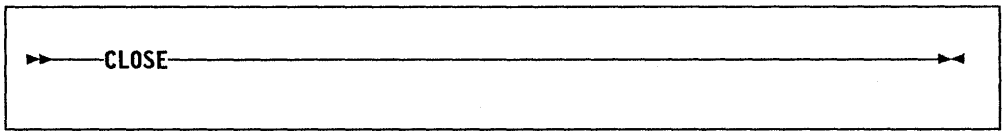
If SUNIQUE is set on, FTP does not overwrite an existing file on the foreign host when a PUT or MPUT subcommand is issued. The file is saved with a different and unique name. The FTP server on the foreign host returns the name of the newly created file to your local host, where it is displayed on your terminal.

SUNIQUE is a "toggle" subcommand. This means that the SUNIQUE subcommand changes the existing setting of the storage method. For example, if the storage method is set so that an existing file is not overwritten (set to on), issuing a SUNIQUE subcommand changes the storage method so that an existing file is overwritten (set to off).

To determine the SUNIQUE setting, use the LOCSTAT subcommand (see "Displaying FTP Status Information" on page 2-9).

Disconnecting From a Foreign Host

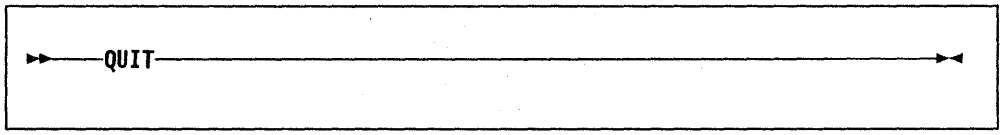
After successfully opening a connection to a foreign host, you cannot connect to a new foreign host without closing the current session. Use the CLOSE subcommand to disconnect from the current foreign host. The format of the CLOSE subcommand is as follows:



After closing a connection to a foreign host with the CLOSE subcommand, you still remain in the FTP environment. Refer to "Leaving the FTP Environment" on page 2-28 for information on how to leave the FTP environment. Issue the OPEN subcommand to open a connection to a new foreign host (see "Opening a Connection to the FTP Server" on page 2-6).

Leaving the FTP Environment

Use the QUIT subcommand to leave the FTP environment. The format of the QUIT subcommand is as follows:



If you have an established connection to a foreign host, issuing the QUIT subcommand closes this connection. You then leave the FTP environment. If you want to disconnect from a foreign host but still want to remain in the FTP environment, use the CLOSE subcommand (see "Disconnecting From a Foreign Host" on page 2-27).

MVS FTP Server Response Messages

In response to the FTP subcommands that you enter, the TCP/IP for MVS program displays the sequence of subcommands, if any, that are sent to the foreign host's FTP server. In addition, the subcommands' coded response is also displayed. These replies ensure the synchronization of requests and actions in the process of data transfer, and guarantee that your process always knows the state of the foreign host's FTP server. The following table describes the possible response messages:

Table 2-4 (Page 1 of 2). FTP Response Messages

Message Number	Description
110	Restart marker reply
120	Service ready in *nnn* minutes
125	Data connection already open; transfer starting
150	File status okay; about to open data connection
200	Command okay
202	Command not implemented; not used on this host
211	System status, or system help reply
212	Directory status
213	File status
214	Help message
220	Service ready for new user
226	Closing data connection; requested file action successful
230	User logged on; proceed
250	Requested file action okay, completed

Table 2-4 (Page 2 of 2). FTP Response Messages	
Message Number	**Description**
257	*Path_name* created
421	Service not available; closing Telnet connection
425	Cannot open data connection
426	Connection closed; transfer ended abnormally
450	Requested file action not taken; file busy
451	Requested action aborted; local error in processing
452	Requested action not taken; insufficient storage space in system
500	Syntax error; command unrecognized
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command not implemented for that parameter
530	Not logged on
532	Need account for storing files
550	Requested action not taken; file not found or no access
551	Requested action aborted; page type unknown
552	Requested file action ended abnormally; exceeded storage allocation
553	Requested action not taken; file name not allowed

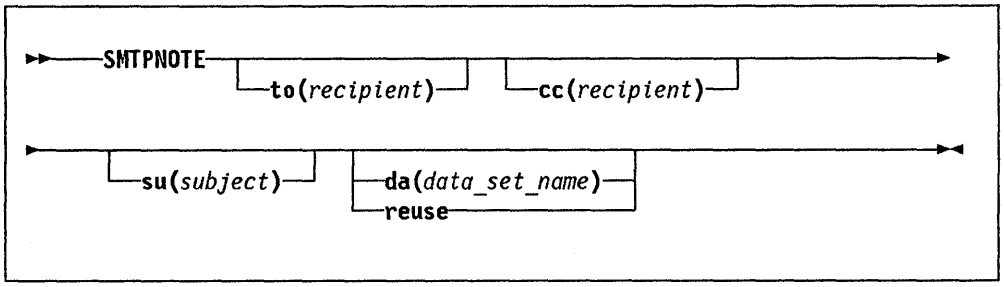
Chapter 3. Sending Electronic Mail

Use the SMTPNOTE command, provided with the TCP/IP for MVS program, to prepare and send electronic mail. Recipients of the mail can be users on your local host, users on Network Job Entry (NJE) or Remote Spooling Communication Subsystem (RSCS) hosts, or users on TCP hosts. The SMTPNOTE command uses the Simple Mail Transfer Protocol (SMTP) to send the mail.

Using the SMTPNOTE Command

Use the SMTPNOTE command to send electronic mail to one or more recipients on NJE, RSCS, or TCP networks. The SMTPNOTE command allows you to prepare the mail using the facilities of the Time Sharing Option (TSO) EDIT command, or to send mail prepared with an editor of your choice.

The format of the SMTPNOTE command is as follows:



Parameter	Description
to	Is used to specify a single recipient for the mail. If you do not specify the **to** parameter, you will be prompted to enter a list of recipients. Enter the name of each recipient on a separate line and end the list with a blank line. You must specify at least one mail recipient.
cc	Is used to specify a single carbon copy recipient for the mail. If you do not specify the **cc** parameter, you will be prompted to enter a list of carbon copy recipients. If there are no carbon copy recipients, press the ENTER key at that time, otherwise, enter the name of each recipient on a separate line and end the list with a blank line.
recipient	Is the path address of the mail recipient. The format of *recipient* is equivalent to path syntax, as described in RFC 821, without the < and > delimiters. This *recipient* parameter will likely have one of the following formats: *user_id@host_name* User on a host in your domain, possibly on your local node.

user_id@host_name.domain
User on a host in a specified domain.

user_id%nje_host_name@gateway_name.domain
User on an NJE or RSCS node connected to a TCP network at *gateway_name*. See "Electronic Mail Gateway" on page 3-5 for more information.

@host1,@host2,...,@hostn:user_id@host_name
User on a host that is not known by the local domain name server, but that can be reached by following the path from *host1 to hostn*.

su Is used to specify the subject of the note. If you do not specify the **su** parameter, you will be prompted for the subject. If there is no subject for the note, press the ENTER key at that time.

subject Is the subject of the note. It can be any arbitrary string of characters, but is limited to a total length of 233 characters.

da Is used to indicate that the text of the mail is contained in a data set. The data set can have any record format, can be blocked or unblocked, and can have records of up to 243 characters in length (records longer than 243 characters will be truncated).

data_set_name Is the name of the data set containing the text of the note. It is a valid data set name, and is fully qualified if it is contained within quotes (').

reuse Is specified to cause SMTP to re-use the contents of a note that was previously cancelled. If no note was cancelled, the **reuse** parameter is ignored.

All lower case characters in the above parameters are changed to upper case by TSO before being passed to SMTPNOTE. If you wish to specify parameter values that contain lower case characters, wait to be prompted before you enter these values.

Note: SMTPNOTE is implemented as a CLIST. If your parameter values include blanks, single quotes, semicolons, or commas, the values must be enclosed within single quotes, and each occurrence of a single quote must be replaced with two single quotes. For example, to specify the **da** parameter with a fully qualified data set name of 'USER.DATASET', the parameter is specified as **da**(''''USER.DATASET''''). For more information see *IBM TSO Extensions CLISTS* (SC28-1304).

Preparing and Sending Mail

After you enter the SMTPNOTE command, you are prompted for the mail recipients (TO:), the carbon copy recipients (CC:), and the subject of the note (SUBJECT:), if they were not specified with the SMTPNOTE command. If you enter a list of recipients, enter the name of each recipient on a separate line, and indicate the end of the list by entering a blank line.

After you answer the prompts, SMTPNOTE invokes the TSO EDIT command to allow you to prepare your note. Typically, the editor starts in INPUT mode. Enter the text of your note line by line. When your note is complete, enter a null line (that is,

do not type anything when prompted) and press the ENTER key. The editor switches to EDIT mode.

If you invoke the SMTPNOTE command with the **reuse** or **da** parameter, you are immediately placed in EDIT mode. The contents of the previously cancelled note, or the data set that you specify, will already be part of the note. You can add to, or change the data that is already present.

In EDIT mode you can use all the functions of the editor. You can also return to INPUT mode, send the note, or cancel the note. For a complete description of the EDIT command, see *IBM TSO Extensions Command Language Reference* (SC28-1307).

To send the note enter SAVE or END in EDIT mode. To cancel the note enter CANCEL in EDIT mode. If you cancel a note, you can recover what you typed by invoking the SMTPNOTE command with the **reuse** parameter. The recipients and subject of the note are not saved, and must be re-entered.

For an example of preparing and sending mail, see Figure 3-1.

```
READY
smtpnote
TO:        irvine@mvs2.accounting
           bekker@mvs2.accounting
           mcgregr@mvs1.accounting

CC:
SUBJECT:   Travel Expenses

INPUT
0010 Could we please postpone the expense review since I will be out
0020 of town the week of the 19th.
0030
0040 How about the 23rd?  Thank you, John.
0050

ENTER 'SAVE' OR 'END' TO SEND THE NOTE.
USE 'CANCEL' TO TERMINATE WITHOUT SENDING THE NOTE.
FOR A COMPLETE LIST OF EDIT SUBCOMMANDS ENTER 'HELP'.

EDIT
save
READY
```

Figure 3-1. Example of Preparing and Sending a Mail

Receiving Mail

Use the TSO RECEIVE command to receive SMTP mail, as you would other mail and messages. For more information, see *IBM TSO Extensions Interactive Data Transmission Facility: User's Guide* (SC28-1104).

Nondelivery Notes

When SMTP cannot deliver a piece of mail, a nondelivery note explaining the reason for nondelivery is sent to the sender. Nondelivery can occur for several reasons, such as if a destination host is unreachable, or if the recipient does not have a user ID on the destination host. If this happens, the body of the original piece of mail is returned as part of the nondelivery notification. For an example of a nondelivery note, see Figure 3-2.

```
Date: Mon, 11 Sep 89 23:58:05 CST
From: SMTP@castle.org
To:   CHARMING@COTTAGE
Subject: Undeliverable Mail

castle.org unable to deliver following mail to recipient(s):
      <sneezy@forest.com>
castle.org unable to connect for 3 days to host:
           forest.com

       ** Text of Mail follows **
Received: from COTTAGE by castle.org Thu, 14 Sep 89 23:51:52 CST
Date:     14 Sep 89 23:51:51 CST
From:     Prince Charming <CHARMING@COTTAGE>
To:       sneezy@forest.com

Sneezy, I won't be able to meet you for lunch tomorrow.  Prince Charming.
```

Figure 3-2. Example of a Nondelivery Note

If a recipient is unknown at the destination host, the destination host does not accept the mail for delivery, and a nondelivery note is forwarded to the sender. For an example of an unknown recipient note, see Figure 3-3.

```
Date: Mon, 11 Sep 89 00:09:39 CST
From: SMTP@castle.org
To:   CHARMING@COTTAGE
Subject: Undeliverable Mail

castle.org unable to deliver following mail to recipient(s):
    <SnowWhite@forest.com>.
castle.org received negative reply from host:
    forest.com
550 <SnowWhite@forest.com>... User unknown

       ** Text of Mail follows **
Received: from COTTAGE by castle.org; Mon, 11 Sep 89 00:09:30 CST
Date:     11 Sep 89 00:09:28 CST
From:     CHARMING@COTTAGE
To:       SnowWhite@forest.com

Snow White - are you still living here?
```

Figure 3-3. Example of an Unknown Recipient Note

Electronic Mail Gateway

SMTP can be configured to run as a mail gateway between TCP network users and users located on an NJE or RSCS network attached to the local host (see Figure 3-4).

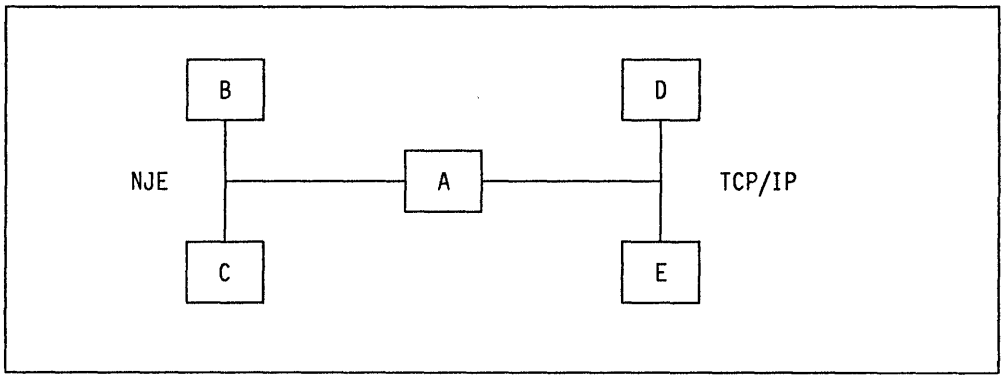


Figure 3-4. SMTP as a Mail Gateway

For example, when A is the local MVS host, running both the TCP/IP for MVS program and NJE:

B and C are hosts attached to host A by means of an NJE network.
D and E are hosts attached to host A by means of a TCP network.

Users on hosts A, B, and C can send mail to users on TCP hosts D and E using the SMTPNOTE command. This method is described in "Using the SMTPNOTE Command" on page 3-1.

Users on TCP hosts D and E can send mail to users on host A using addresses in the following format:

user_id@A.domain

where *user_id* is the user ID of the TSO user on host A, and *A.domain* is the TCP host name of host A.

Users on TCP hosts D and E can send mail to users on NJE hosts B and C using addresses in the following format:

user_id%nje_host_name@A.domain

where *user_id* is the user ID of the user on the NJE host, *nje_host_name* is the name of the NJE host (B or C), and *A.domain* is the TCP host name of host A.

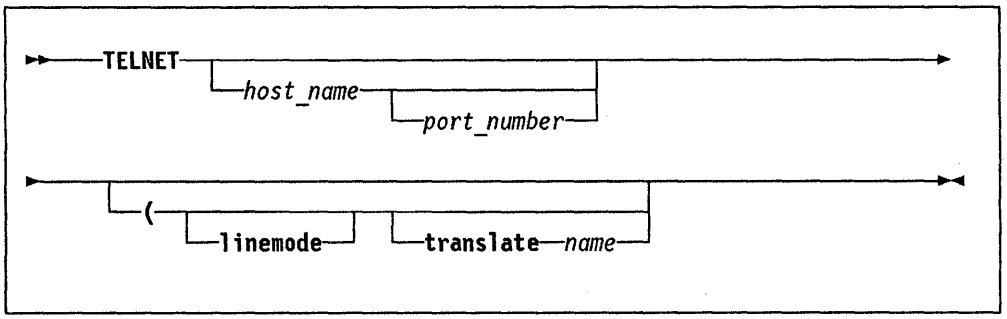
Chapter 4. Logging on to a Host

The Telnet protocol provides a standardized interface, allowing terminal devices and terminal-oriented processes on hosts that support TCP/IP to communicate with each other.

Use the TELNET command to log on to a foreign host that supports TCP/IP.

TELNET Command Format

The format of the TELNET command is as follows:



Parameter	Description
host_name	Is the name or internet address of the foreign host you want to log on to.
port_number	Is a decimal integer identifying the port you want to connect to on the foreign host. The port must conform to the Telnet protocol. The default *port_number* is 23. This parameter is used for system test purposes only.
linemode	Specify if you want to operate in line mode. It prevents operation in transparent mode. Abbreviations for this parameter are accepted (**l** is the shortest abbreviation).
translate *name*	Specify if you want to use a non-standard translation table. If you specify this parameter, TELNET uses the translation table in the *user_id.name*.TCPXLBIN data set instead of the standard translation table provided with the TCP/IP for MVS program (*user_id*.TELNET.TCPXLBIN or TCPIP.TELNET.TCPXLBIN). If *user_id.name*.TCPXLBIN does not exist, TELNET uses TCPIP.*name*.TCPXLBIN. If *user_id.name*.TCPXLBIN and TCPIP.*name*.TCPXLBIN do not exist, or if they were incorrectly created, TELNET terminates with an error message. Abbreviations for this parameter are accepted (**t** is the shortest abbreviation).

In 3270 transparent mode, all full-screen capabilities of the foreign host are functional at your local display station. In line mode, the foreign host's output displays on your screen one line at a time, without full-screen capabilities.

When you use the TELNET command to connect to a foreign host running the TCP/IP for MVS or TCP/IP for VM program, your terminal session on the foreign host emulates a terminal session on that host.

If you invoke the services of the MVS Telnet server from a non-MVS or a non-VM host, a transparent mode of operation may not be possible. You will be automatically connected to the MVS server as a line-mode, start-stop TTY terminal.

If your Telnet session ends for any reason, the following message is displayed:

```
Session ended.  < ENTER >  to return to TSO.
```

When you return to TSO, a message, giving the reason why the Telnet session ended, is displayed. The following is an example of what can be displayed when you return to TSO.

```
Telnet terminated -- Foreign host is no longer responding
```

The TELNET command supports IBM 3270-type display stations. Examples of supported display stations are:

IBM 3178 Display Station
IBM 3179 Display Station
IBM 3180 Display Station
IBM 3191 Display Station
IBM 3192 Display Station
IBM 3193 Display Station
IBM 3194 Display Station
IBM 3275 Display Station Model 2
IBM 3276 Control Unit Display Station Models 2, 3, and 4
IBM 3277 Display Station Model 2
IBM 3278 Display Station Models 2, 3, 4, and 5
IBM 3279 Color Display Station Models 2 and 3.

TELNET Subcommands

Use the TELNET subcommands to control the Telnet session. Table 4-1 gives a short description of the various TELNET subcommands and refers you to a page where you can obtain additional information on the use of the specific subcommand. The table also lists the minimum abbreviation for each subcommand.

To invoke a TELNET subcommand, press the function key designated to get the program's attention (refer to "TELNET Function Keys" for information about the function keys that are available). After pressing the attention key, you are prompted to enter a TELNET subcommand. You can enter the TELNET subcommands in upper or lower case.

Table 4-1. TELNET Subcommands Quick Reference

Subcommand	Minimum Abbreviation	Description	Page
AO	AO	End the display of output.	4-5
AYT	AY	Query the existence of the connection.	4-4
HELP	H or ?	Receive assistance with the TELNET command.	4-4
IP	I	Interrupt the current process.	4-5
PA1	P	Send the PA1 keystroke to the session to which you connected (transparent mode only).	4-6
QUIT	Q	Quit the TELNET session.	4-6
SYNCH	S	Clear the data path.	4-5

When connected to the MVS Telnet server as a line-mode, start-stop TTY terminal, two TELNET subcommands have a special meaning: the AO and IP subcommands cause a single attention to be presented.

Line-mode terminals have a key labeled Break or Attn. Issuing the AO or the IP subcommand is equivalent to pressing either of these keys once. Programs treat the Break key or the Attn key as a request to interrupt the program.

TELNET Function Keys

In Transparent Mode

In transparent mode, the only function key available is the PA1 attention key. It indicates that you want to invoke a TELNET subcommand.

Refer to "Sending the PA1 Keystroke to a Host" on page 4-6 for information on how to send the PA1 keystroke to the foreign host session.

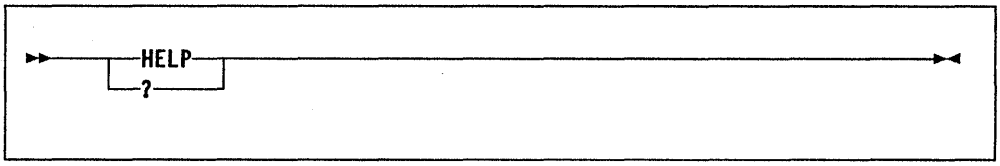
In Line Mode

The following table describes the function keys that are available in line mode.

Table 4-2. TELNET Function Keys in Line Mode	
Function Key	**Description**
PF4 to PF12, PF16 to PF24	Allows you to invoke a TELNET subcommand.
PF1, PF13	Retrieves the previous input line.
PF2, PF14	Scrolls half way up the screen.
PF3, PF15	Turns off input line display. Use either of these keys before entering your password.

Obtaining Assistance

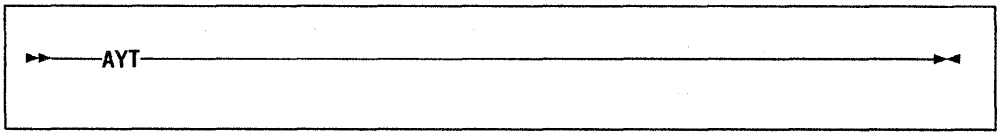
Use the HELP or the ? subcommand to receive assistance with the TELNET command. The format of the HELP or the ? subcommand is as follows:



When you issue the HELP or the ? subcommand, a list of valid TELNET subcommands is displayed.

Querying the Connection

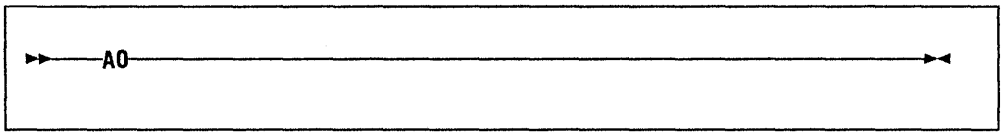
If you open a session to a foreign host, you can use the AYT (Are You There) subcommand to ensure that the connection exists. This is useful when the response from the foreign host is slow, and you want to ensure that the connection still exists. The format of the AYT subcommand is as follows:



If the connection exists, the terminal makes a sound if you are operating in transparent mode. You receive a message from the Telnet server if you are operating in line mode.

Terminating Output Display

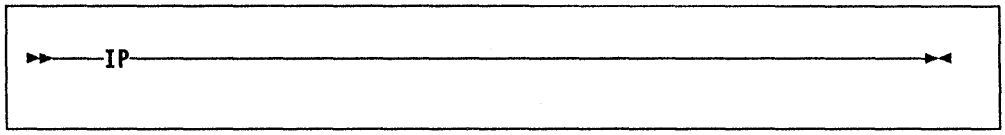
Use the AO (Abort Output) subcommand to terminate the output display of a command. This subcommand is useful once you have issued a command and then decide that you want to terminate the displaying of its output. The format of the AO subcommand is as follows:



Use the AO subcommand to clear any output that has already been produced but that has not been displayed on your terminal.

Interrupting The Process

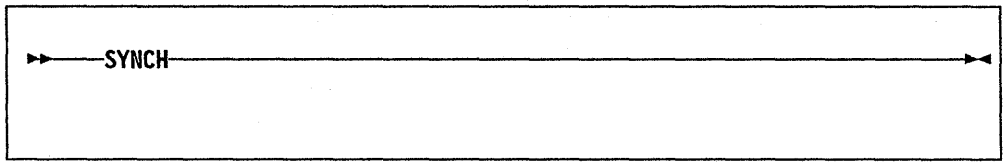
Use the IP (Interrupt Process) subcommand to interrupt the current process running on the foreign host. The format of the IP subcommand is as follows:



Many hosts allow you to suspend, interrupt, or terminate the operation of a process. This capability is useful when you want to stop a process that is in an unending loop, or when you want to stop a process that you inadvertently started. The IP subcommand invokes this function on the foreign host.

Clearing the Data Path

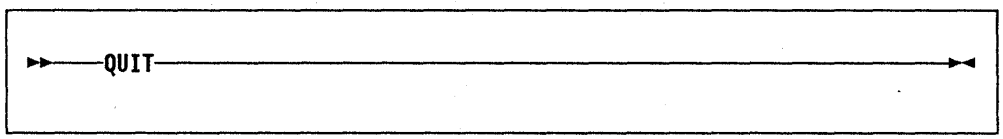
Use the SYNCH subcommand to clear the data path to the foreign host. The format of the SYNCH subcommand is as follows:



The SYNCH subcommand does not clear TELNET subcommands that are in the data path.

Ending the Telnet Session

Use the QUIT subcommand to end the Telnet session. The format of the QUIT subcommand is as follows:

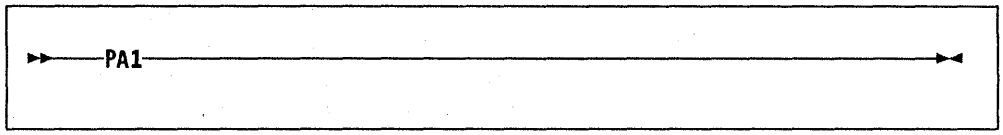


If you are connected to a foreign host running the TCP/IP for MVS program, issuing the QUIT subcommand may only disconnect you, not log you off, from the foreign host. The Virtual Telecommunications Access Method (VTAM) application you are accessing determines whether you will also be logged off from the foreign host. For example, the TSO application disconnects you but does not log you off.

To log off from the foreign host, use the logoff procedure of that host before issuing the QUIT subcommand.

Sending the PA1 Keystroke to a Host

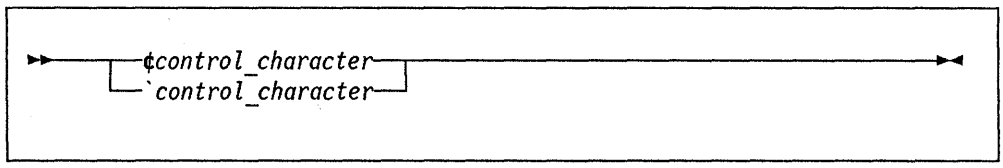
Use the PA1 subcommand to send the PA1 keystroke to the foreign host session. The format of the PA1 subcommand is as follows:



The PA1 subcommand replaces the PA1 attention key on the foreign host session. This is required because the PA1 key is the Telnet attention key. The PA1 subcommand operates only in transparent mode.

Sending ASCII Control Characters to a Host in Line Mode

Use the cent sign (¢) or the grave accent (`) if you are logged on to a foreign host in line mode and want to send an ASCII control character to that host. The format to send these control characters is as follows:



Parameter **Description**

control_character
Is the ASCII control character that you want to send to the foreign host. Control characters are specific to the foreign host.

For example, send **Ctl-p** in one of the following formats:

¢p

`p

Other control characters are shown in Table 4-3.

Table 4-3. Control Characters	
ASCII Output	**Character Input**
01 - 1A (Ctl-a - Ctl-z)	`A - `Z
1B - 1F	`2 - `6
5B (left square bracket - [)	`{
5D (right square bracket -])	`}
FF (DEL)	`#

Suppressing Carriage Return and Line Feed

Use the cent sign (¢) or the grave accent (`) if you are logged on to a foreign host in line mode and want to suppress the sending of a carriage return and line feed. Type the cent sign (¢) or the grave accent (`) before pressing the ENTER key.

Figure 4-1 on page 4-8 shows the execution of a BSD UNIX[1] program called MORE. This program displays one line or one page at a time. A carriage return character (CR) causes it to display one line, while a blank character causes it to display one page. If you are executing this program from an MVS host, use the grave accent (`) character to suppress the CR that is normally sent when you press the ENTER key.

[1] Trademark of American Telephone and Telegraph Company.

```
% more hosts.local
NET : 4.0.0.0 : SATNET :
NET : 6.0.0.0 : YPG-NET :
NET : 7.0.0.0 : EDN-TEMP :
NET : 8.0.0.0 : BBNCCNET :
NET : 9.0.0.0 : IBM :
NET : 10.0.0.0 : ARPANET :
NET : 12.0.0.0 : ATT :
NET : 13.0.0.0 : XEROX-NET :
NET : 14.0.0.0 : PDN :
NET : 15.0.0.0 : HP-INTERNET :
NET : 18.0.0.0 : MIT-TEMP :
NET : 21.0.0.0 : DDN-RVN :
NET : 23.0.0.0 : DDN-TC-NET :
NET : 24.0.0.0 : MINET :
NET : 25.0.0.0 : RSRE-EXP :
NET : 26.0.0.0 : MILNET :
NET : 27.0.0.0 : NOSC-LCCN-TEMP :
NET : 28.0.0.0 : WIDEBAND :
NET : 29.0.0.0 : MILX25-TEMP :
NET : 30.0.0.0 : ARPAX25-TEMP :
NET : 31.0.0.0 : UCDLA-NET :
NET : 35.0.0.0 : MERIT :
--More--(0%)  [HIT <ENTER> HERE.  ONE LINE IS DISPLAYED]
NET : 36.0.0.0 : SU-NET-TEMP :
--More--(0%)
```

```
NET : 39.0.0.0 : SRINET-TEMP :
--More--(0%)  [HIT <BLANK>, <ACCENT GRAVE>, <ENTER>]
NET : 39.0.0.0 : SRINET-TEMP :
NET : 41.0.0.0 : BBN-TEST-A :
NET : 42.0.0.0 : CAN-INET :
NET : 44.0.0.0 : AMPRNET :
NET : 46.0.0.0 : BBNET :
NET : 128.1.0.0 : BBN-TEST-B :
NET : 128.2.0.0 : CMU-NET :
NET : 128.3.0.0 : LBL-IP-NET1 :
NET : 128.4.0.0 : DCNET :
NET : 128.5.0.0 : FORDNET :
NET : 128.6.0.0 : RUTGERS :
NET : 128.7.0.0 : KRAUTNET :
NET : 128.8.0.0 : UMDNET :
NET : 128.9.0.0 : ISI-NET :
NET : 128.10.0.0 : PURDUE-CS-EN :
NET : 128.11.0.0 : BBN-CRONUS :
NET : 128.12.0.0 : SU-NET :
NET : 128.13.0.0 : MATNET :
NET : 128.14.0.0 : BBN-SAT-TEST :
NET : 128.15.0.0 : S1NET :
NET : 128.16.0.0 : UCLNET :
NET : 128.17.0.0 : MATNET-ALT :
--More--(1%)
```

Figure 4-1. Sample Output of the MORE Program

It is useful if the command environment of the foreign host responds to the typing of a single character, without the need for a carriage return and line feed after it. This function is also useful when your cursor is at the end of the input field, but you need to continue the line without introducing a carriage return.

Chapter 5. Obtaining Network Status Information

Use the NETSTAT command to obtain information about the network connections of your host. This information is retrieved from the TCPIP address space.

NETSTAT Command Format

The format of the NETSTAT command is one of the following:

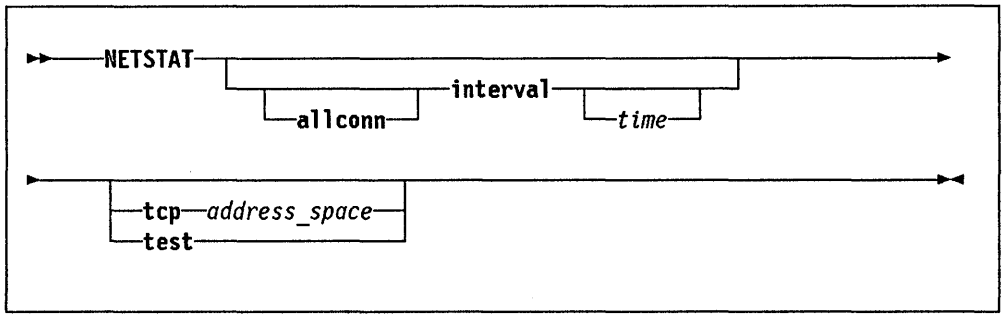


Figure 5-1. NETSTAT Command in Full Screen Mode

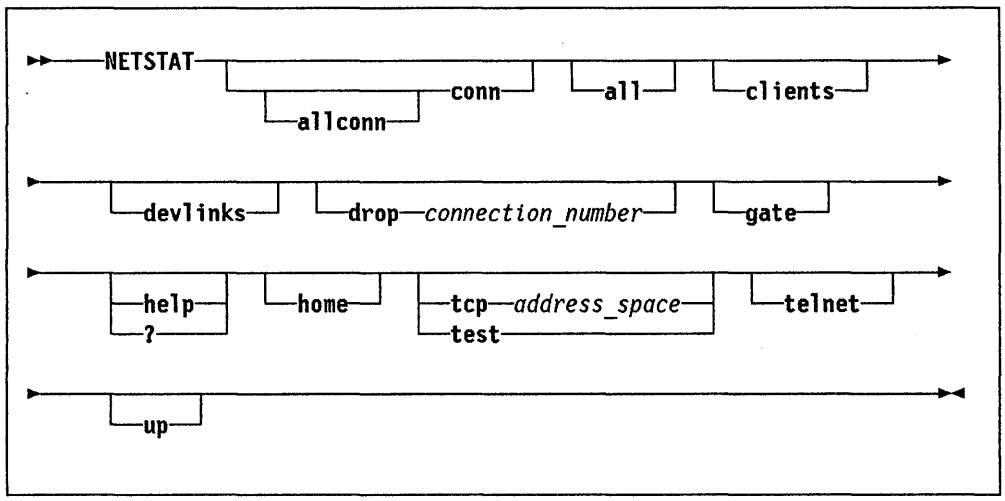


Figure 5-2. NETSTAT Command in Non Full Screen Mode

The order in which you enter the NETSTAT command parameters is not significant. Specifying the NETSTAT command with no parameters is equivalent to specifying the command with the **conn** parameter (see "Displaying Information About TCP Connections and UDP Ports" on page 5-3).

You can place more than one parameter on the command line. The only parameters you can specify with the **interval** parameter are **allconn**, **tcp** or **test**.

Internet addresses are interpreted and printed symbolically if possible. If an address cannot be interpreted, it is displayed in the dotted decimal notation. Unspecified

addresses or ports are displayed as an asterisk (*). Known port numbers are displayed symbolically. Other port numbers are displayed numerically. An internet address - port number pair is displayed as the internet address and port number separated by two periods (..).

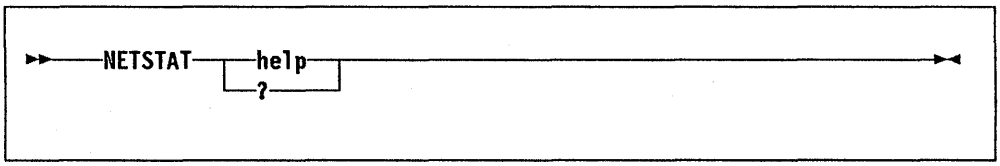
The following table gives a short description of how the various NETSTAT parameters affect the output of the command. The table refers you to a page where additional information on the use of the specific parameter can be obtained. The table also lists the minimum abbreviation for each parameter.

Table 5-1. NETSTAT Parameters

Parameter	Minimum Abbreviation	Description	Page
all	**all**	Displays detailed information about all TCP connections.	5-12
allconn	**allc**	Displays all the connections.	5-3
clients	**cl**	Displays information about TCP clients.	5-5
conn	**co**	Displays information about active TCP connections and UDP ports.	5-3
devlinks	**de**	Displays information about all devices and TCPIP links.	5-6
drop	**dr**	Drops the specified TCP connection.	5-13
gate	**g**	Displays information about IP routing.	5-9
help or ?	**he**	Displays information about the NETSTAT parameters.	5-2
home	**ho**	Displays the home list.	5-9
interval	**i**	Displays refreshed full screen information about active TCP connections and UDP ports.	5-3
tcp	**tc**	Displays detailed information about the specified address space.	5-12
telnet	**tel**	Displays the status of the internal Telnet server.	5-10
test	**tes**	Displays detailed information about the TCPIPTES address space.	5-12
up	**u**	Displays information about when TCPIP was started.	5-11

Obtaining Assistance

To receive assistance with the NETSTAT command, specify the **help** or **?** parameter with the NETSTAT command as follows:



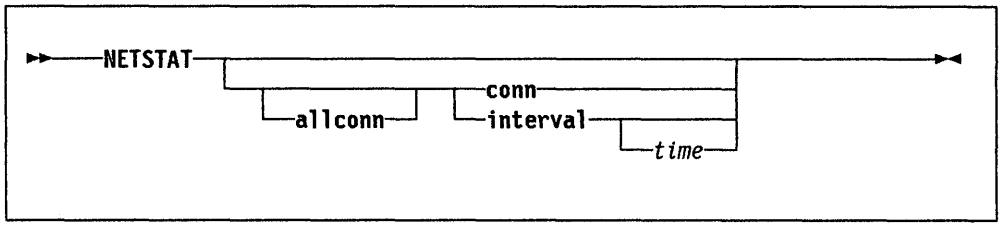
If you specify the **help** or **?** parameter, information on the various NETSTAT parameters is displayed.

Displaying Information About TCP Connections and UDP Ports

To display information about active TCP connections and UDP ports, specify the **conn** or the **interval** parameter with the NETSTAT command. An active connection is one that is not in the "closed" or in the "time-wait" state.

The **allconn** parameter is specified along with the **conn** or the **interval** parameter to modify the output to include information about connections that are not active.

The format of the command is as follows:



If you specify the **conn** parameter, the following information is displayed for each active connection:

- User ID
- Connection number
- Local internet address - port number pair
- Foreign internet address - port number pair
- Connection state.

The **conn** parameter is the default for the NETSTAT command.

A sample output of the NETSTAT **conn** command is shown below:

```
READY
netstat conn
MVS TCP/IP Netstat R1.0

Active Transmission Blocks
User Id  Conn  Local Socket          Foreign Socket        State
-------  ----  ------------          --------------        -----
INTCLIEN 1011  *..TELNET             *..*                  Listen
INTCLIEN 1004  HOSTA..TELNET         HOSTB..1127           Established
SNEEZY   UDP   *..PMAP               *..*                   UDP
SNEEZY   1001  HOSTA..PMAP           *..*                  Listen
CHARMING UDP   *..2049               *..*                   UDP
READY
```

Figure 5-3. Sample Output of the NETSTAT **conn** Command

If you specify the **interval** parameter, the following information is displayed, using the full screen, for each connection:

- User ID
- Bytes received on the connection
- Bytes sent on the connection
- Local port number
- Foreign internet address - port number pair
- Connection state
- Number of Transmission Control Blocks (TCBs) in use
- Screen refresh interval
- The idle time
- Number of connections not displayed (optional).

Idle time refers to the time the connection was idle within TCP. By default, the screen is updated every 20 seconds. If you want to change this default, specify a value for *time* in seconds. Press ENTER to end the screen updates.

A sample output of the NETSTAT **interval** command is shown below:

```
09/12/89            MVS TCP/IP Real Time Network Monitor              11:29:48

User Id     B Out    B In  L Port Foreign Socket          State          Idle
--------    -----    ----- ------ --------------          -----------    --------
INTCLIEN        0        0 TELNET *..*                    Listen         2:46:37
INTCLIEN     5765       82 TELNET HOSTA..1026             Established    0:00:54
INTCLIEN     5828       82 TELNET HOSTB..1028             Established    0:11:05
SNEEZY        180      320 PMAP   *..*                    UDP            0:58:25
SNEEZY          0        0 PMAP   *..*                    Listen         0:58:25
CHARMING        0        0 2049   *..*                    UDP            1:01:03

Refresh interval: 20 Seconds.                                TCB's In Use:  4
```

Figure 5-4. Sample Output of the NETSTAT **interval** Command

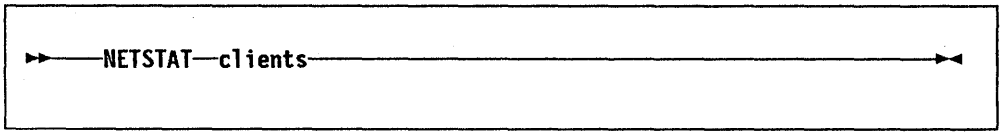
Note: The **interval** parameter can be used on an IBM 3270-type display station, or a workstation emulating an IBM 3270. Examples of supported display stations are:

IBM 3178 Display Station
IBM 3179 Display Station
IBM 3180 Display Station
IBM 3191 Display Station
IBM 3192 Display Station
IBM 3193 Display Station

IBM 3194 Display Station
IBM 3275 Display Station Model 2
IBM 3276 Control Unit Display Station Models 2, 3, and 4
IBM 3277 Display Station Model 2
IBM 3278 Display Station Models 2, 3, 4, and 5
IBM 3279 Color Display Station Models 2 and 3.

Displaying Information About TCP Clients

To display information about clients known to TCP, specify the **clients** parameter with the NETSTAT command as follows:



If you specify the **clients** parameter, the following information is displayed for each client:

- Client name
- Client's authorization
- Notes handled by the client
- Elapsed time since the client was last active
- Elapsed time since the client was last forced
- Client's Virtual Machine Communication Facility (VMCF) error count.

A sample output of the NETSTAT **clients** command is shown below:

```
READY
netstat clients
MVS TCP/IP Netstat R1.0

Current Clients:
Client: TCPMAINT        Authorization: Monitor, Informed
Notes Handled: none
Last Touched:  41:39:09 Last Forced:  41:55:49
Vmcf error count: 0

Client: SNOWHITE        Authorization: Monitor
Notes Handled: none
Last Touched:  41:39:10 Last Forced:  41:55:50
Vmcf error count: 0
```

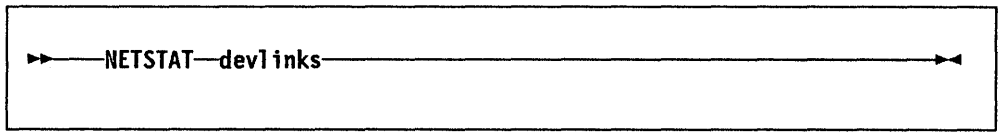
Figure 5-5 (Part 1 of 2). Sample Output of the NETSTAT **clients** Command

```
Client: INTCLIEN        Authorization: {none}
Notes Handled: Buffer space available, Connection state changed, Data delivered,
User wants attention, Timer expired, FSend response, FReceive error, RawIp
packet received
Last Touched:   0:00:07 Last Forced:  41:55:49
Vmcf error count: 0

Client: SRVRFTP         Authorization: {none}
Notes Handled: Buffer space available, Connection state changed, Data delivered
Last Touched:   3:19:10 Last Forced:  25:30:27
Vmcf error count: 0

Client: SNEEZY          Authorization: {none}
Notes Handled: none
Last Touched:   0:06:47 Last Forced:   4:41:25
Vmcf error count: 0

Client: SMTP            Authorization: {none}
Notes Handled: none
Last Touched:   0:36:08 Last Forced:   0:58:51
Vmcf error count: 0

READY
```

Figure 5-5 (Part 2 of 2). Sample Output of the NETSTAT **clients** Command

Displaying Information About Devices and Links

To display information about all devices and links in the TCPIP address space, use the **devlinks** parameter with the NETSTAT command as follows:



A sample output of the NETSTAT **devlinks** command is shown below:

```
READY
netstat devlinks
MVS TCP/IP Netstat R1.0

Device LCS1             Type: LCS         Status: Ready
Queue size: 0           Address: 0780
   Link TOKEN1          Type: IBMTR       Net number: 0**Error Status**
   Link ETHER1          Type: ETHERNET    Net number: 0
READY
```

Figure 5-6. Sample Output #1 of the NETSTAT **devlinks** Command

Field	Description
DEVICE	Is the name of the device. In the above sample output, the name of the device is LCS1.

TYPE (Device) Is the device type. In the above sample output, the device type is LCS, which means that the device is an 8232 LAN Channel Station (LCS).

STATUS Is the status of the device. Some device drivers do not provide device-specific status. For these devices, the possible values are INACTIVE, when the device is not started, and ACTIVE, when the device is started. The 8232 LCS, driver provides information on the progress of its initialization procedure. This information is useful if the initialization encounters difficulties. When initialization is complete, the STATUS field has a value of READY. If not started, the STATUS field has a value of INACTIVE for these devices. In the above sample output, the 8232 LCS has a status of READY.

QUEUE SIZE Is the number of datagrams queued for output. In the above sample output, the value of this field is 0, indicating that no datagrams are queued for output.

ADDRESS Is the base address of the device. In the above sample output, the base address is 0780.

LINK Is the name of the link defined on the device specified in the DEVICE field. In the above sample output, two links are defined on the device called LCS1; their names are TOKEN1 and ETHER1.

TYPE (Link) Is the link type for the defined link. In the above sample output, TYPE has a value of IBMTR (indicating that it is an IBM Token Ring link) and ETHERNET (indicating that it is an Ethernet[1] link).

NET NUMBER Is only meaningful for links on LCS devices. It is the network number of the link of a certain type on the LCS device. In the above sample output, this number has a value of 0 for the Token Ring link, indicating that it is the first Token Ring link for the LCS device. This number is also 0 for the Ethernet link, indicating that it is the first Ethernet link for the LCS device. The network number for each link is specified with the LINK statement for LCS devices. Refer to the *IBM TCP/IP for MVS: Installation and Maintenance* book for more information on the LINK statement.

The '**ERROR STATUS**' message appearing beside the network number for the Token Ring link indicates that this link is not operational.

The exact meaning of some fields is device dependent.

[1] Trademark of Xerox Incorporated.

Another sample output of the NETSTAT **devlinks** command is shown below:

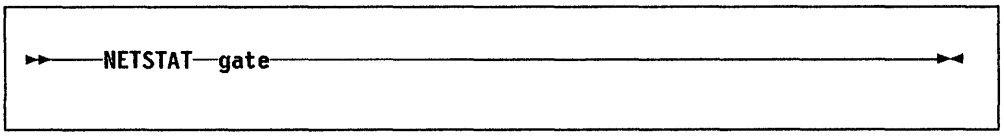
```
READY
netstat devlinks
MVS TCP/IP Netstat R1.0

Device SNALU0           Type: SNA IUCV    Status: Issued connect
Queue size: 0           Jobname: SNALINK  Pgm: SNALINK      LU: SNALNK02
   Link SNA1            Type: IUCV        Net number: 1
READY
```

Figure 5-7. Sample Output #2 of the NETSTAT **devlinks** Command

Field	Description
DEVICE	Is the name of the device. The above sample output shows the device called SNALU0.
TYPE (Device)	Is the device type. In the above sample output, the device type is SNA IUCV, which means that it is a System Network Architecture (SNA) Inter-User Communication Vehicle (IUCV) device.
STATUS	Is the status of the device. In the above sample output, the device has a status of ISSUED CONNECT, meaning that, for the passive side, SNALINK is waiting for a remote Logical Unit (LU) to establish a session. For the active side, SNALINK is trying to establish a session with a remote LU.
QUEUE SIZE	Is the number of datagrams that are queued for output. In the above sample output, the value of this field is 0, indicating that no datagrams are queued for output.
JOBNAME	Is the name of the address space the specified device is connected to. In the above sample output, the value of this field is SNALINK.
PGM	Is the name of the program that is running in the address space specified in the JOBNAME field. In the above sample output, the value of this field is SNALINK.
LU	Is the name of the LU to which the SNA session will be established. In the above sample output, the LU name is SNALNK02.
LINK	Is the name of the link defined on the device specified in the DEVICE field. In the above sample output, the name of the link defined on the device is SNA1.
TYPE (Link)	Is the link type for the defined link. In the above sample output, TYPE has a value of IUCV.
NET NUMBER	Is only meaningful for links on LCS devices (see Figure 5-6 on page 5-6).

Displaying Information About TCP Gateways

To display information about gateways known to TCP, use the **gate** parameter with the NETSTAT command as follows:



You can find the address of a particular gateway by using the **gate** parameter.

The following IP routing information is given for each network in the internet:

- Address of the network
- First "hop" address
- Link name used by the first "hop"
- Packet size used by the first "hop"
- Subnet mask and subnet value.

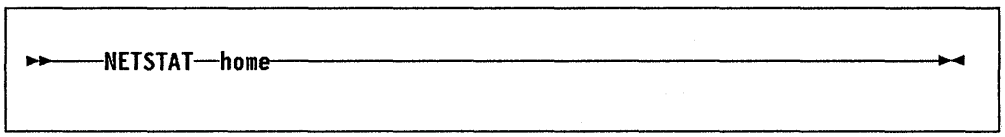
A sample output of the NETSTAT **gate** command is shown below:

```
READY
netstat gate
MVS TCP/IP Netstat R1.0
Known gateways:

NetAddress      FirstHop        Link     Pkt Sz  Subnet Mask     Subnet Value
----------      --------        ------   ------  -----------     ------------
Default         129.34.50.254   TOKEN1   1500    <none>
129.34.0.0      <direct>        TOKEN1   1500    0.0.255.0       0.0.50.0
129.34.0.0      <direct>        ETHER1   1500    0.0.255.0       0.0.31.0
READY
```

Figure 5-8. Sample Output of the NETSTAT **gate** Command

Displaying Home List Internet Addresses and Link Names

To display the internet address and the corresponding link name for each entry in the home list, use the **home** parameter with the NETSTAT command as follows:



A sample output of the NETSTAT **home** command is shown below:

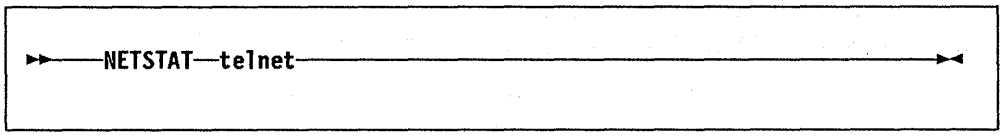
```
READY
netstat home
MVS TCP/IP Netstat R1.0
Home address list:

Address           Link
-------           ------
129.34.50.41      TOKEN1
129.34.31.41      ETHER1
READY
```

Figure 5-9. Sample Output of the NETSTAT **home** Command

For more information about the home list, refer to the *IBM TCP/IP for MVS: Installation and Maintenance* book.

Displaying the Status of the Internal Telnet Server

To display the status of the internal Telnet server, use the **telnet** parameter with the NETSTAT command as follows:



If you specify the **telnet** parameter, the following information is displayed for the internal Telnet server:

- Connection number
- Connection status
- Foreign host name
- Bytes received on the connection
- Bytes sent on the connection
- Logical device status, indicating the following:
  - Internal logical device number
  - LU name of the application
  - VTAM LU name used by the Telnet server to create the logical device (LU name of your terminal).

A sample output of the NETSTAT **telnet** command is shown below:

```
READY
netstat telnet
MVS TCP/IP Netstat R1.0
Internal Telnet server status:

Conn Status       Foreign Host   B out    B in  Logical device status
---- ------       ------------   -----    ----  ---------------------
1014 Established  SLEEPY        239091    2559  0002 RE05Q002 RE051025
1012 Established  DOPEY          72414     687  0003 RE05Q003 RE051040
1002 Established  GRUMPY        214938    2964  0001 RE05Q001 RE051019
1010 Listen       *                  0       0
READY
```

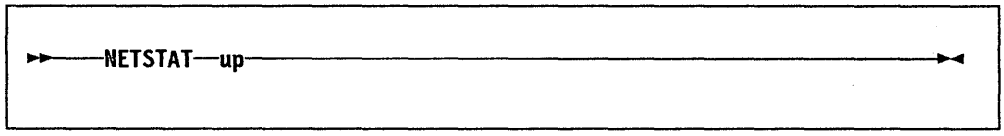
Figure 5-10. Sample Output of the NETSTAT **telnet** Command

There should always be a connection available in the listening state for an incoming open request.

To confirm the existence of a connection, display the status of the Telnet server using the **telnet** parameter, and refer to the list of connections to see if the one that you are looking for exists.

Displaying Information About When TCP/IP Was Started

To display information about when TCP/IP was started, use the **up** parameter with the NETSTAT command as follows:



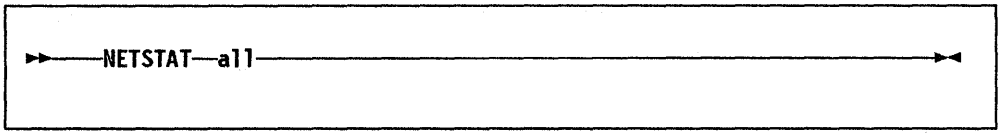
A sample output of the NETSTAT **up** command is shown below:

```
READY
netstat up
MVS TCP/IP Netstat R1.0
Tcpip started at 21:28:38 on 09/12/89
READY
```

Figure 5-11. Sample Output of the NETSTAT **up** Command

Displaying Debugging Information About TCP Connections

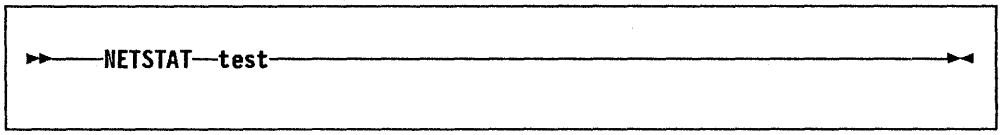
To display detailed debugging information about every TCP connection, specify the **all** parameter with the NETSTAT command as follows:



The **all** parameter is useful for debugging the TCPIP address space. For more information on maintaining the TCPIP address space, refer to the *IBM TCP/IP for MVS: Installation and Maintenance* book.

Information About the TCPIPTES Address Space

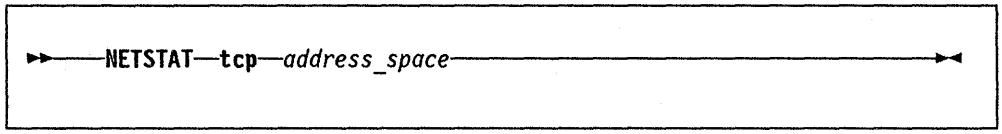
To display information about the TCPIPTES address space, specify the **test** parameter with the NETSTAT command. With the **test** parameter, the NETSTAT request is serviced by the TCPIPTES address space. The following is the format of the NETSTAT command:



If you specify the **test** parameter, information about the TCPIPTES address space is displayed instead of information about the TCPIP address space. The **test** parameter is useful for system testing.

Information About a Specified Address Space

To display information about a specified address space, specify the **tcp** parameter with the NETSTAT command. With the **tcp** parameter, the NETSTAT request is serviced by the specified address space. The format of the NETSTAT command is as follows:



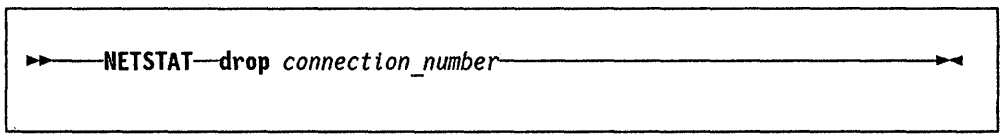
where *address_space* is the name of the address space.

If you specify the **tcp** parameter, information about the *address_space* address space is displayed instead of information about the TCPIP address space. The **tcp** parameter is useful for system testing.

The NETSTAT **tcp tcpiptes** is equivalent to the NETSTAT **test** command (see "Information About the TCPIPTES Address Space").

Dropping a Specified TCP Connection

To drop a specified TCP connection, use the **drop** parameter with the NETSTAT command as follows:



The *connection_number* specifies the number of the TCP connection. Determine the *connection_number* from the CONN column in the output from the NETSTAT **conn** (see "Displaying Information About TCP Connections and UDP Ports" on page 5-3) or the NETSTAT **telnet** (see "Displaying the Status of the Internal Telnet Server" on page 5-10) command.

Note: If you drop a connection that is a server's passive open, the server immediately re-issues the open request.

The **drop** parameter can only be used by users who are specified in the obey list. For more information about the obey list, refer to the *IBM TCP/IP for MVS: Installation and Maintenance* book.

Chapter 6. PING Command

Use the PING command to send an Internet Control Message Protocol (ICMP) echo request to a foreign host. When the foreign host receives the echo request, it must reply to the host that sent the request (your local host). If the foreign host does not respond, it indicates that the foreign host is unavailable, that a network or a gateway leading from your local host to the foreign host is unavailable, or that the foreign host does not implement the ICMP echo request protocol.

Use additional PING commands to determine where the failure is most likely located. Issue the PING commands in the following order until the failure is located:

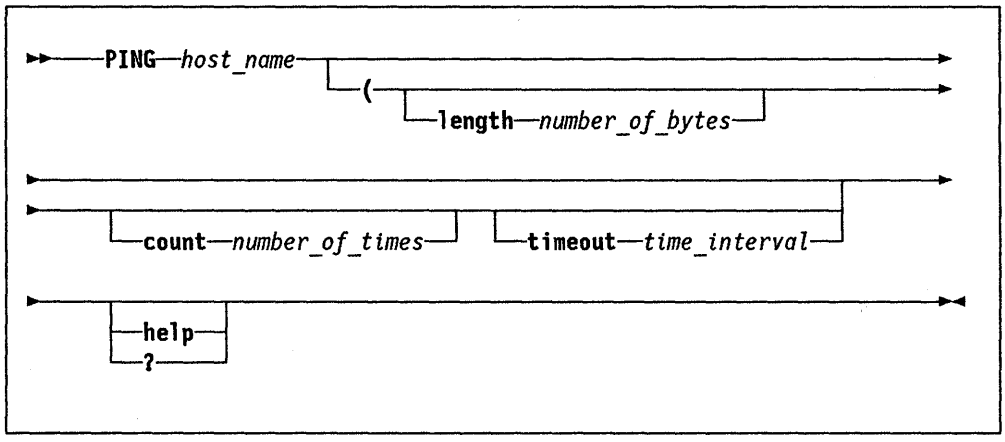
1. Direct a PING command at your local host.
2. Direct a PING command at a host (other than your local host) on your local network.
3. Direct PING commands at each intermediate node leading from your local host to the foreign host. Start with the node closest to your local host.

When a response to a PING command is received, the elapsed time is displayed. This time does not include the time spent in Virtual Machine Communication Facility (VMCF) communication with the TCPIP address space.

The echo request sent by the PING command is dispatched using a low-level protocol that does not guarantee delivery. As a result, the possibility exists that any one echo request may be accidentally lost. Do not assume that a network or gateway has failed until several PING commands fail to get beyond a point.

PING Command Format

The format of the PING command is as follows:



Parameter	Description
host_name	Is either the name or the internet address of the foreign host being pinged. If *host_name* is omitted, you are prompted for it.
length *number_of_bytes*	Is used to specify the number of bytes to be sent by each PING command. *number_of_bytes* is the number of bytes. The default is 256 bytes, the maximum value is 512 bytes, and the minimum value is 8 bytes. Abbreviations for the **length** parameter are accepted (**l** is the shortest abbreviation).
count *number_of_times*	Is used to specify the number of times that the foreign host is to be pinged. *number_of_times* is the number of times. If you do not specify the **count** parameter, the foreign host is pinged only once. If you specify the *number_of_times* as 0, the foreign host is pinged continually. Abbreviations for the **count** parameter are accepted (**c** is the shortest abbreviation).
timeout *time_interval*	Is used to specify the number of seconds that the PING command is to wait for a response from the foreign host. *time_interval* is the number of seconds. The default is 10 seconds, the maximum value is 100 seconds, and the minimum value is 1 second. Abbreviations for the **timeout** parameter are accepted (**t** is the shortest abbreviation).
help or **?**	Is used to obtain information about the PING command. If you specify this parameter, you cannot specify other parameters.

The order in which you enter the **length**, **count** and **timeout** parameters is not significant.

Return Codes

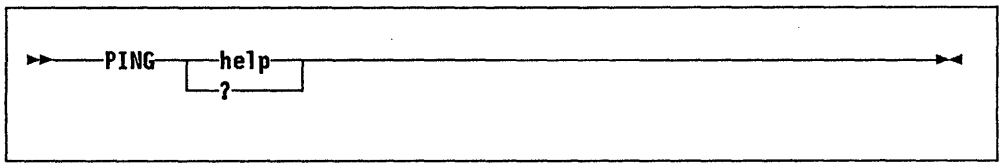
Return Code	Explanation
0	Response on at least one attempt.
4	No response.
8	TCPIP address space not operational.
100	Bad Parameter.

The return codes are not displayed on the screen, but messages giving the status of the PING command are displayed.

Note: Return codes can be tested in a CLIST using the CC variable.

Obtaining Assistance

To receive assistance with the PING command, specify the **help** or **?** parameter with the PING command as follows:



Abbreviations for the **help** parameter are accepted (**h** being the shortest abbreviation).

Terminating the PING Command

Enter PAI to end the PING command, regardless of the value of the *number_of_times* variable specified with the **count** parameter.

Chapter 7. Using the Network File System

This chapter provides the workstation user and the workstation administrator with information about using the Network File System server feature of the TCP/IP for MVS program.

This chapter contains the following information:

- An introduction to the TCP/IP for MVS Network File System feature
- A description of the commands used to determine the attributes of a data set, as well as a description of the commands used to access a data set on the MVS mainframe (examples are included)
- A description of how to create your own data set on the mainframe, with descriptions of the required parameters and attributes (examples are provided).

Related Publications

For a listing of the error messages sent to the client session, see Appendix B, "Network File System Messages for the Client" on page B-1.

The *IBM TCP/IP for MVS: Programmer's Reference* book contains information about the four exit routines provided with the Network File System server code: Logon, Security, Accounting and Archive.

The *IBM TCP/IP for MVS: Installation and Maintenance* book contains a listing of the Network File System server messages appearing at the MVS host. It also contains information about installing and configuring the Network File System server.

For information on topics related to the allocation and characteristics of MVS data sets, refer to the *IBM TSO Extensions User's Guide* (SC28-1333).

Syntax Conventions and Guidelines

The various terms and conventions that are particular to this chapter are defined below. They are used in addition to those described in "How to Read the Syntax Diagrams" on page iv.

, Commas are used as separators when specifying data set attributes.

() Parentheses in data set attributes must be passed to MVS, but the UNIX[1] client systems (such as AIX™) shells recognize parentheses and strip them off. To stop this action, use one of the following methods:

- Include the entire parameter in double (") or single (') quotes.
- Include each parenthetical expression in double (") or single (') quotes.
- Precede each parenthesis with a back slash (\).

---

[1] Trademark of American Telephone and Telegraph Company.

AIX is a trademark of the International Business Machines Corporation.

$ The dollar sign may be used in the file name of an MVS mounted data set on UNIX. To prevent UNIX from attempting to interpret $ as part of a variable, use one of the following methods:

- Include the entire parameter in double (") or single (') quotes.
- Include each parenthetical expression in double (") or single (') quotes.
- Precede each parenthesis with a back slash (\).

% Character in an AIX prompt indicating any user is allowed to perform the command.

Note:

1. Prompts for user and superuser may vary depending on the original AIX or UNIX implementation, or on site defaults. The % convention outlined above is followed in this chapter for illustrational purposes.
2. Throughout this book, a railroad track syntax convention is used to display the syntax of commands. To clarify the explanations for AIX and like users who intend to use only the Network File System feature, without reading the rest of this book, both railroad track diagrams and AIX screen samples are used to illustrate new commands.
3. Commands are not shown capitalized as is the convention everywhere else in the book. This is because AIX and UNIX implementations are case sensitive, and commands must be issued in lower case at the client. Network File System commands issued at the client are shown in the **bold lowercase font**.

Introducing the Network File System Server

The Network File System feature of TCP/IP for MVS provides access to data sets for users of workstations that support the client function of Sun Microsystems NFS[2] protocol. This access depends on the communication services provided by TCP/IP for MVS.

The client function of the NFS protocol is supported on the IBM RT™ computer by the AIX RT 2.2.1 Network File System (program number 5601-159) product, on the System/370 by the AIX/370 (program number 5688-046) product, and on the IBM PS/2® computer by the AIX PS/2 Network File System product.

The Network File System server on MVS handles requests from authorized clients to access MVS data sets as if they were local to the client system. The Network File System server also provides experienced mainframe users with the capability of specifying data set creation and processing attributes for the client.

---

[2] Trademark of Sun Microsystems Incorporated.

RT is a trademark of the International Business Machines Corporation.

PS/2 is a registered trademark of the International Business Machines Corporation.

MVS programs, as well as data, are maintained on direct access storage devices (DASD). To access any of this information, a set of routines called access methods is used. The Network File System server acts as an intermediary, dealing with the access methods to read, write, create or delete MVS data sets.

Reading and Writing

The Network File System server reads or writes binary and text data. In a file that contains text, record boundaries are translated into line terminators (new line characters, for example). A file that contains binary data is stored in the mainframe and retrieved with no text translation at all. Access to MVS data sets is transparent to the users and the files appear as extensions to the local disk.

Obtaining Access

To access a data set, a mount point on the client workstation must be specified for each MVS catalog entry or high-level index you plan to address. Attributes are used to specify processing and data set characteristics, with default attributes being specified when the Network File System server is installed. You can modify the attributes when you issue the **mount** command or when you access the data set with commands from your workstation. Based on the attribute information supplied, the Network File System server locates the data set and makes the data available to you.

Creating Data Sets

Using the Network File System server, you can create data sets. When you create a data set you will be concerned with how data sets are structured, located, and stored.

Using Mainframe Data Sets

After the Network File System server is installed and you have a path to the mainframe, use regular data access commands to bring data from the mainframe to your application.

This section describes the instructions you use to:

- Create a mount point
- Login
- Display and override attributes.

Preparing to Use the Network File System

Once the Network File System server is installed, you need to do some preparation.

A default set of attributes was defined during the installation of the Network File System server. You can override the installation default settings with a **mount** command. When you access the file with a data access command, you can override any setting created by a **mount** command, or by installation default, or both.

There are two different kinds of attributes:

- Processing attributes provide information to the Network File System server about how to handle the data set (for example, whether the contents are text or binary, or how much time to allow for processing).
- Data Set attributes provide information to the Network File System server, such as the type of data set or, how the data set is allocated: blocks, cylinders, tracks.

The attributes are provided in tables at the end of this chapter.

Know Your Environment's Commands

Depending on the client environment, there may be specific syntactical requirements for commanding the remote mount function. Examples for three possible client implementations are enclosed in the box below (the list is not complete, but serves as an illustration).

```
Client Environment    Sample Command

AIX RT                mount -v nfs -n hostname... /mnt
PC NFS                net use d:\\hostname...
Sun OS                mount hostname:...        /mnt
```

Figure 7-1. Example of Remote Mounts for Clients

Network File System client implementations differ with respect to the privilege level required to issue **mount** and related commands. Many limit these commands to the root user. Unless another Network File System client implementation is explicitly indicated, examples in this chapter are appropriate for use in IBM's AIX RT environment, which permits a **mount** to be issued by any member of the system group with write access to the directory over which the **mount** is to be made.

The mount Command

Use the **mount** command to make a connection between a mount point and a data set on the mainframe. Check with your mainframe system administrator to get a user ID, a password, and permission to access the data set you want to access. To display the current attribute settings, use the **showattr** command, which is described later in this chapter.

For each **mount** command you must follow the format below:

```
station% mount [clnt_opt][mvs_cat,]mvs_filespec[,attributes] localpath
```

Figure 7-2. mount Command Syntax Example

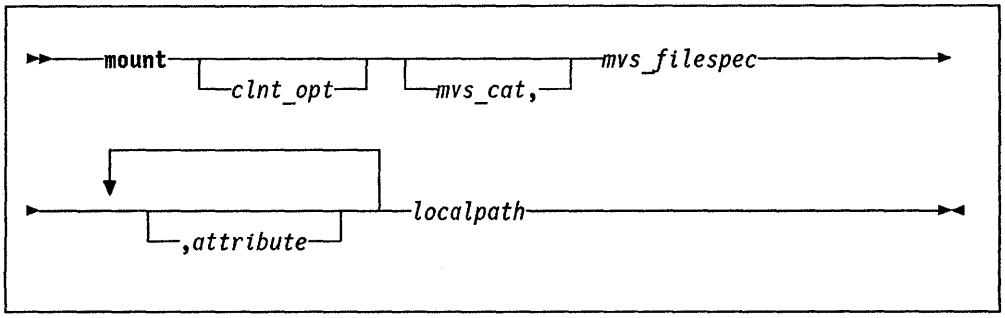


Figure 7-3. mount Command Syntax Diagram

Parameter	Description
clnt_opt	Are the client **mount** command options.
mvs_cat	Is the name of the catalog on the mainframe.
mvs_filespec	Is the name of the high level qualifier of the data set being accessed. It may include a complete data set.
attributes	Is a data set attribute.
localpath	Is the mount point on your machine.

Note: The specified high level index, in this case *mvs_cat*, must already exist in the mainframe's master catalog.

Once a **mount** command is executed, you are ready to login to MVS and start accessing data sets. If you plan to use data sets that already exist, continue reading this section. If you plan to create a data set, refer to "Creating Data Sets" on page 7-15 first.

Getting Access to the Mainframe

Once the Network File System server is installed and you have created the mount point (or points), you are ready to set up a connection to the mainframe.

Depending on the security procedures in use at your installation (check with the system administrator):

- You may need a password to access data sets on the mainframe.
- You may have to use the **mvslogin** command to establish a session between your workstation and the mainframe. Use regular data access commands to do your work and, when you are done, use the **mvslogout** command to break the connection.

  Note: If you cannot use the **mvslogin** or **mvslogout** commands, it is probably because they are incorrectly installed. Ensure your system administrator has made the executable code for the two commands available to the client.

The mvslogin Command

Issue the **mvslogin** command once for each session following the syntax below:

```
station% mvslogin  [-pn] [-g group] [-a account] hostname [mvs_username]
```

Figure 7-4. mvslogin Command Syntax Example

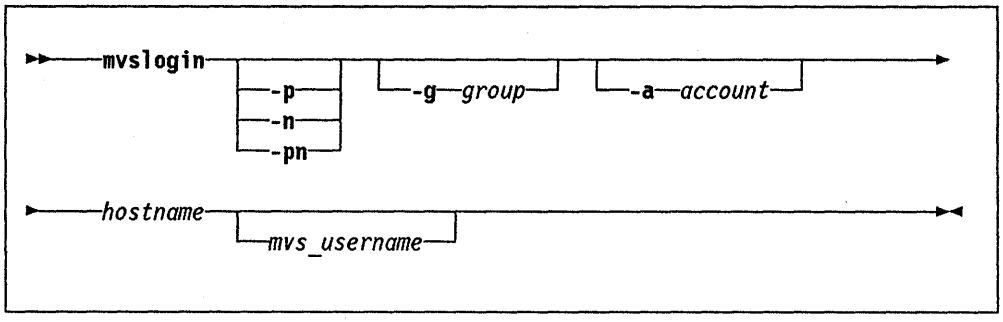


Figure 7-5. mvslogin Command Syntax Diagram

Parameter	Description
-p	It causes a prompt for the user's password. This password is passed to the mainframe to validate the user logging in. The type of security procedures required by the installation determine whether this parameter is required.
-n	It causes a prompt for a new password. The type of security procedures required by the installation determine whether this parameter is used.
group	A group name string passed to the mainframe for accounting purposes.
account	An account string passed to the mainframe for accounting purposes.
hostname	Is the machine name of the mainframe server.
mvs_username	Is a user ID that the mainframe recognizes as a valid login. If this parameter is not specified, your workstation user name is used. Check with your system administrator to find out what you should use.

If you get authentication errors while attempting to access data, your session may have been automatically logged out by the mainframe. Re-execute **mvslogin**, and start your processes again.

Once you have established the session, you can begin accessing a data set. Note that each user on a Network File System client can only establish one **mvslogin** per TCP/IP for MVS Network File System server.

The mvslogout Command

Use this command to break the connection between the workstation and the mainframe. You need to issue this command at the close of each session.

```
station% mvslogout hostname
```

Figure 7-6. mvslogout Command Syntax Example

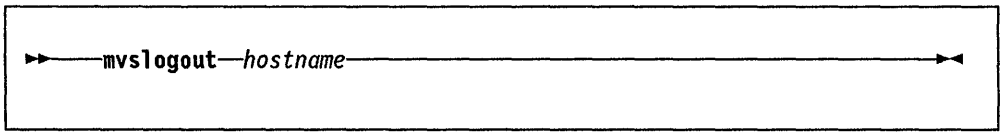


Figure 7-7. mvslogout Command Syntax Diagram

Parameter	Description
hostname	Is the name of the server mainframe.

The umount Command

Use the **umount** command to break the connection between the mount point on your system and the server.

```
station% umount /mnt
```

Figure 7-8. umount Command Syntax Example

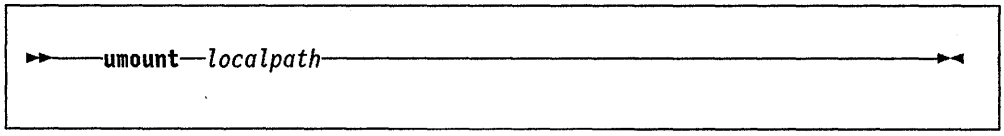


Figure 7-9. umount Command Syntax Diagram

Parameter	Description
localpath	Is the mount point on your machine.

Reading and Writing a Data Set

The following information provides clarification about handling files when using the Network File System server on the mainframe to access files.

When reading a data set, its entire contents are handled by the Network File System server as a sequential byte stream. Binary files are read and transferred with no translation. The contents of files that are defined as text are translated from EBCDIC to ASCII, and end of line characters are inserted at the end of each record. (Record boundaries are assumed to be line terminators.)

Writing a data set on a mainframe is straightforward. If the data set exists, the data set's existing attributes are used. The attributes are not modified in any way during the write operation. When a file is created, the attributes used are those specified for the site default, and any overriding attributes specified with the file name or mount point.

When you write records that have binary data, any padding will be done with null characters. When writing fixed-length records that have text data, the correct size will occur if blank stripping is enabled. Records are padded with blanks.

Notes:

1. If you are writing a text data file with blank stripping enabled and you specify blanks explicitly at the end of the line, there may be a problem when you read it back. All blanks at the end of the record are lost and the file size will be smaller. No non-terminating blank data, however, is lost.
2. A line written that is longer than the maximum record length for the file type will cause an I/O error.

Using an AIX or DOS File

When using an AIX or DOS client, specify the attributes for a binary file if the processing does not require text file sharing with MVS applications. For a DOS client, use CRLF as the line terminator sequence. If you are using AIX use LF as a terminator.

The showattr Command

Use the **showattr** command to display the attributes that have been established for a site or a mount point. The attributes displayed pertain to a mount point if the local path is specified. The format for the command is:

```
station% showattr [-t] hostname [localpath]
```

Figure 7-10. showattr Command Syntax Example

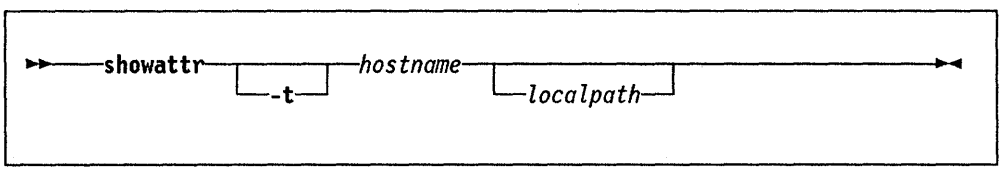


Figure 7-11. showattr Command Syntax Diagram

Parameter	Description
hostname	Is the host name of the server mainframe.
-t	Used to specify terse output.
localpath	Is the path to the directory on your machine that you have created to use as a mount point. This directory is used as part of the path to the data set you want to use.

The **showattr** command can provide attributes for the site only, or use **showattr** to display the site plus the mount point attributes. Before you use a data set for the first time, you might want to check which attributes have been established.

Note: If you cannot use the **showattr** command it is probably because it is incorrectly installed on the client. Ensure your system administrator has made the executable code for the command available to the client.

Examples

The following examples show different ways you can use the **showattr** and **mount** commands.

Example One: Displaying Site Defaults

This example shows a **showattr** command with just the host name (savant) specified. The attributes for the site will be displayed.

```
station%  showattr savant

MVS/NFS Data Set Creation Attributes:

lrecl(8196)     recfm(vb)       blksize(16396)
space(100,200)  blks            dsorg(ps)
dir(27)         recordsize(512,4096) keys(64,0)
nonspanned      shareoptions(1,3)
model()

MVS/NFS Processing Attributes:

text            lf              blankstrip
nofastfilesize  retrieve        maplower        executebitoff
attrtimeout(120)
readtimeout(90)
writetimeout(30)

MVS/NFS Site Attributes (not modifiable):

mintimeout(1)    maxtimeout(1800)    logout(1800)
nfstasks(4)
bufhigh(1048576) readaheadmax(16384) cachewindow(16)
percentsteal(20) maxrdforszleft(32)
```

Figure 7-12. Displaying Site Defaults

If the terse option is used, you get:

```
station%  showattr savant
lrecl(8196),recfm(vb),blksize(16396),space(100,200),blks,dsorg(ps),
dir(27),recordsize(512,4096),keys(64,0),nonspanned,shareoptions(1,3),
model(),text,lf,blankstrip,nofastfilesize,retrieve,maplower,
executebitoff,attrtimeout(120),readtimeout(90),writetimeout(30),$
```

Figure 7-13. Displaying Site Defaults - Terse Option

Example Two: Displaying Site and Mount Point Attributes

This example shows the **showattr** command being used to display the attributes for the site yktvsh as well as the mount point, /x. Note the changes in the second output showing specified options.

```
station%  mount -v nfs -n yktvsh -o soft,time=3 "tcpsrc,binary,space(5,0),trks" x
station%  showattr yktvsh

MVS/NFS Data Set Creation Attributes:

lrecl(8196)     recfm(vb)       blksize(16396)
space(100,200)  blks            dsorg(ps)
dir(27)         recordsize(512,4096) keys(64,0)
nonspanned      shareoptions(1,3)
model()

MVS/NFS Processing Attributes:

text            lf              blankstrip
nofastfilesize  retrieve        maplower        executebitoff
attrtimeout(120)
readtomeout(90)
writetimeout(30)

MVS/NFS Site Attributes (not modifiable):

mintimeout(1)    maxtimeout(1800)    logout(1800)
nfstasks(4)
bufhigh(1048576) readaheadmax(16384) cachewindow(16)
percentsteal(20) maxrdforszleft(32)
```

```
station%  showattr yktvsh /u2/ryniker/x

MVS/NFS Data Set Creation Attributes:

lrecl(8196)     recfm(vb)       blksize(16396)
space(5,0)      trks            dsorg(ps)
dir(27)         recordsize(512,4096) keys(64,0)
nonspanned      shareoptions(1,3)
model()

MVS/NFS Processing Attributes:

binary          lf              blankstrip
nofastfilesize  retrieve        maplower        executebitoff
attrtimeout(120)
readtomeout(90)
writetimeout(30)

MVS/NFS Site Attributes (not modifiable):

mintimeout(1)    maxtimeout(1800)    logout(1800)
nfstasks(4)
bufhigh(1048576) readaheadmax(16384) cachewindow(16)
percentsteal(20) maxrdforszleft(32)
```

Figure 7-14. Displaying Site and Mount Point Attributes

Example Three: Changing Attributes

In this example, the **mount** command is used to modify some of the attributes displayed above.

```
station%  mount -v nfs -n savant -o soft "muiz,binary,
          recfm(VB),readtimeout(30)"  /mnt
```

Figure 7-15. Changing Attributes

Parameter	Description
savant	Is the name of the host server.
muiz	Is the name of the high level catalog.
binary	Is the type of data being sent from the mainframe.
recfm(vb)	Is now the format of the records.
readtimeout(30)	is the amount of time allowed before the data set is released (deallocated).
/mnt	Is the name of the mount point.

Changing the Data Set Attributes

You may have to override the default attributes established for a particular site or mount point. When this is necessary, first check the attribute settings and be sure a change is required.

To override the site default attributes set by the system administrator at your installation, specify different attributes with the **mount** command or in the data set access command.

Attribute Tables

This section contains tables for Processing and Data Set Attributes.

Processing Attributes

The Processing attributes affect how the data set is translated between the network and the mainframe. The Processing attributes set during installation can be modified by Network File System clients. The values of the following attributes are contingent on the settings of the associated site attributes:

- The attributes **attrtimeout**, **readtimeout**, and **writetimeout** must be within the ranges specified by the **maxtimeout** and **mintimeout** site attributes.
- The attributes **noattrtimeout**, **noreadtimeout**, and **nowritetimeout** are valid only if **nomaxtimeout** has been specified in the site attributes.

The processing attributes are described in Table 7-1 on page 7-12.

Table 7-1 (Page 1 of 2). Processing Attributes

Processing Attributes	Description
attrtimeout	The amount of time in seconds the data set remains allocated to the Network File System following a **lookup** or **showattr** server operation.
noattrtimeout	The data set is deallocated immediately following a **lookup** or **showattr** operation.
binary	Indicates that the contents of the data set are in binary format and no text translation occurs.
text	Translates the contents in the data set between EBCDIC and ASCII formats. Use this format if it is necessary to share text data with other MVS applications. End of line terminators can be CR, CRLF (for PC clients), LF, LFCR, or NOEOL. EOL is used to indicate the MVS logical record boundary.
blankstrip	Strips trailing blanks at the end of each record of a fixed-length text file when the file is read. Pads the end of each file or record with blanks when a text file is written.
noblankstrip	Does not strip trailing blanks at the end of fixed-length records when a fixed-length text file is read. Does not pad records when writing a text file. The file must be of the correct size or an I/O error is reported to the client.
End of Line Specifiers	These apply to text files only and are as follows: LF — Line Feed is terminator (standard UNIX) CR — Carriage Return is record terminator LFCR — Line Feed followed by Carriage Return is record terminator CRLF — Carriage Return followed by Line Feed is record terminator (standard DOS) NOEOL — No End-of-line terminator
executebiton	Turns on the execute bits in user, group and other for mount point's files. Use when storing executables or shell scripts on the MVS system on a mount point basis.
executebitoff	Turns off the execute bits in user, group and other for mount point's files. This value is normally used in the site file.
fastfilesize	Calculates only approximate file sizes during data set access. This speeds up data set access. WARNING: Only use when you are browsing through files, for example, using the **ls** command because some commands (such as **cp**) may not work correctly if **fastfilesize** is set.
nofastfilesize	Accurate file sizes are returned to the client.
maplower	Turn on mapping of lower case file names to upper case when accessing files on MVS, and back when sending to the network. This option would normally be enabled for access by UNIX clients.
nomaplower	Turn off mapping of lower case file names to upper case and back when using files on MVS.
readtimeout	The amount of time before a data set is released (in seconds).
noreadtimeout	The data set is deallocated immediately after a read operation.

Table 7-1 (Page 2 of 2). Processing Attributes	
Processing Attributes	**Description**
retrieve	Forces a call to the site archive retrieval exit for off line data sets. **retrieve** is used to access an offline (archived) data set. The archive exit determines whether or not to retrieve the data set, depending on the action of the exit as well as the user's permission.
noretrieve	An off line data set generates a "file not found" error. The archive exit is not used.
writetimeout	The amount of time in seconds before a data set is released after a write operation.
nowritetimeout	The data set is deallocated after a write operation.

Data Set Attributes

These attributes correspond to the Data Control Block (DCB) or the Job Control Language (JCL) parameters used to define a data set when it is created. They may be modified by the user through the **mount** command. Refer to the *IBM MVS/370 JCL Reference* (GC28-1350) or the *IBM MVS Extended Architecture JCL Reference* (GC28-1352) for more detailed information about the data set creation attributes.

These attributes are described in Table 7-2.

Table 7-2 (Page 1 of 2). Data Set Attributes	
Data Set Attributes	**Description**
blks	Specifies that disk space is allocated by blocks, except for VSAM data sets (see the **recs** attribute).
blksize (*quan*)	Specifies the maximum length, in bytes, of a block. *quan* is a number from 1 to 32 760.
cyls	Specifies that disk space is allocated by cylinders.
dir (*quan*)	The number of 256-byte records needed in the directory of a Partitioned Data Set (PDS). It is only used with a **mkdir** operation.
dsorg (*org*)	Specifies the organization of a data set. *org* can be a Physical Sequential (PS) data set, Direct Access (DA) data set, VSAM KSDS (INDEXED), VSAM RRDS (NUMBERED) or VSAM ESDS (NONINDEXED).
keys (*len, off*)	The length and offset of the keys for VSAM indexed KSDS data sets. They can only be specified as **dsorg**(INDEXED). *len* and *off* are specified in bytes. When you create a VSAM KSDS data set, the records must be key sequential or the write fails. Each write of the data set is treated like an initial load, and requires that the records being loaded are in ascending key sequence.

Table 7-2 (Page 2 of 2). Data Set Attributes	
Data Set Attributes	**Description**
lrecl (*quan*)	Specifies: 1. The length, in bytes, for fixed-length records. 2. The maximum length, in bytes, for variable-length records. The value must be no more than 4 bytes less than the **blksize** quantity. *quan* is a number from 1 to 32 760 in both cases.
model (*den*)	The name of the cataloged VSAM data set from which to copy data set attributes when creating a new VSAM data set. *den* is a valid MVS name.

recfm (*ccc*)	Specifies the format and characteristics of the records in the data set. *ccc* can be 1 to 3 characters, in any of the following combinations:
	`f  fb | fa | fba | fm | fbm | f`
	`u  um | ua | u`
	`v  vs | vb | vbs | va | vba | vm | vbm | v`
	• **a** is ASA,
	• **b** is blocked,
	• **f** is fixed format,
	• **m** is machine,
	• **s** is spanned,
	• **u** is undefined, and
	• **v** is variable format.
	In **recfm** the codes **a** and **m** are mutually exclusive, that is, they cannot be specified in the same data set definition.

recordsize(*ave,max*)	The average and maximum record size for VSAM data sets. *ave* and *max* are specified in bytes. These values must be equal for VSAM RRDS.
recs	Disk space is allocated by records for VSAM data sets.
shareoptions(*xreg,xsys*)	Cross region and cross system share options for a VSAM data set.
spanned	Data sets may contain spanned records.
nonspanned	Data sets do not have spanned records.
space (*prim*[,*sec*])	The amount of space allocated for a new data set on a direct access volume. *prim* is the number (from 0 to 16 777 215) of tracks, cylinders, or data blocks in the data set. *sec* (optional) is the number (from 0 to 16 777 215) of additional tracks, cylinders, or blocks allocated if more space is needed. If this parameter is not specified, the default is used.
trks	Specifies that records (files) are allocated by tracks.
vol (*volser*)	The volume name on which to create a data set.

Notes About Data Set Attributes

Specifying an attribute several times on a line does not cause an error. The line is read from left to right, and the last one of any duplicate is used. For example:

```
station %  vi file,recfm"(vb)",recfm"(fb)"
```

Figure 7-16. Specifying Multiple Attributes

results in a file created with a fixed-blocked format.

Creating Data Sets

This section provides general descriptions of the types of data sets supported by the Network File System server and provides the information you need to create a data set. When you create a data set, you need to specify the type of data set. As the data set is created, default attributes will be associated with it. With these attributes, the Network File System server specifies to MVS the required data set characteristics.

Preparing

If you are going to create a new data set, be sure you have the following information:

1. The kind of data that will be stored in it (binary or text)
2. The record layout
3. The type of data set to create.

   The following types of data sets are supported by the TCP/IP for MVS Network File System server feature:
   - Physical Sequential (PS)
   - Direct Access (DA)
   - Partitioned Data Sets (PDS)
   - Virtual Sequential (VSAM) data sets of type KSDS, ESDS, RRDS.

   Keyed access to files is not supported.

Physical Sequential (PS) Data Sets

When creating a PS data set, you must specify at least the following attribute: **dsorg(ps)**.

If you do not specify any other parameters, the MVS site defaults set up at installation time are used. Use the **showattr** command to display the site defaults if you have any doubts.

Direct Access (DA) Data Sets

When creating a DA data set, you must specify: **dsorg(da)**

If you do not specify any other parameters, the defaults set up at installation time are used. Use the **showattr** command to display the site defaults if you have any doubts.

Partitioned Data Sets (PDS)

When creating a PDS, remember that MVS only supports one level of directory. You cannot create a PDS within a PDS.

Creating a PDS

To create a PDS do the following:

1. Become root and login to the mainframe.

```
station%  mount -v nfs -n savant muiz /mnt
```

Figure 7-17. Login to Mainframe

2. Now use **mkdir** to create a PDS:

```
station%  mkdir /mnt/newone,dir(20)
```

Figure 7-18. Create a PDS Using mkdir

You now have a PDS with twenty 256-byte records for its directory.

Removing a PDS

To remove a PDS use the **rmdir** utility. The following example removes the newone directory, and confirms its removal by a failed attempt to query it after the fact.

To try this, enter:

```
station%  ls -F /mnt/newone
  ...
station%  rmdir /mnt/newone
station%  ls -F /mnt/newone
/mnt/newone not found
```

Figure 7-19. Removing a PDS Using rmdir

Note that a PDS must be empty before it can be removed.

Example of Creating a Mount Point for a PDS

Support for PDS in the Network File System server is provided, but only for one level of directory. This is an MVS limitation. For example, if you have a PDS in a high level index called peony, then you can mount the PDS by entering:

```
station%  mkdir /source
station%  mount -v nfs -n hostname:peony.source,text /source
```

Figure 7-20. Creating a Mount Point for a PDS

The above example associates the PDS named peony.source on MVS with the directory named /source on your machine. There is no difference between mounting a PDS in a high level index or an index with a prefix.

Another Example: Assume that there is a member bigblue in the library peony.source. This member may be displayed on an AIX RT client system by either of the following command sequences:

```
station%  mount -v nfs -n hostname -o soft,timeo=20 peony,text /mvs
station%  cat /mvs/source/bigblue
```

Figure 7-21. Creating a Mount Point for a PDS (1st instance)

OR

```
station%  mount -v nfs -n hostname -o soft,timeo=20 peony.source,text /mvs
station%  cat /mvs/bigblue
```

Figure 7-22. Creating a Mount Point for a PDS (2nd instance)

If there is another library, peony.oldsourc on MVS with a member bigblue, it could be displayed in the first instance by using the command

```
station%  cat /mvs/oldsourc/bigblue
```

Figure 7-23. Creating a Mount Point for a PDS (other information)

The file is not accessible through the **mount** used in the second instance; another **mount** is required to access the oldsourc library.

There is no intrinsic advantage to using either technique over the other. You should choose based on personal convenience and at your discretion.

Note: The PDS support in the Network File System server adheres to the locking conventions used in MVS. For example, you cannot have more than one member of a PDS open for writing at a time.

A PDS member will stay open for the applicable timeout period (set by the site defaults, in the **mount** command or when the data set is accessed). Until the file times out, you cannot access another member of the PDS for writing. Keep your timeouts relatively short (< 30 seconds) if you need to create and write into several members of a PDS.

If you attempt to write to another member of a PDS while one is open for writing and time out has not occurred, you get the error message:

```
Permission denied
```

Note that this restriction does not exist when writing multiple data sets that are non-PDS or when writing members in different partitioned data sets. You will be able to write to a new file when the time expires for the member open for writing.

A file name is restricted to eight characters and it must be a valid name for a PDS member.

VSAM Data Sets

There are three different types of VSAM data sets: key-sequenced, entry-sequenced, and relative record. Each type allows both sequential and direct access. Before choosing a data set format, consider the following:

1. Record access method (by sequence or random)
2. Length of records (all equal? does the length vary between records?)
3. Frequency of record access, and deletion
4. Spanned vs. nonspanned
5. Data order requirements: by the contents of the record, or by an alternate index.

The following table contains a quick summary of what each data set format offers.

Key-Sequenced Data Set	Entry-Sequenced Data Set	Relative Record Data Set
Records are in collating sequence by key field. The key must contain a unique value and be in the same position in each record.	Records remain in the order in which they are entered.	Records are in relative record number, in ascending order.
Direct access by key or by RBA.	Direct access by RBA.	Direct access by relative record number.
Alternate indexes allowed.	Alternate indexes allowed.	No alternate indexes.
A record's RBA can change.	A record's RBA cannot change.	A record's relative record number cannot change.
Free space is used for inserting and lengthening records.	Space at the end of the data set is used for adding records.	A slot given by a deleted record can be re-used.
Spanned records allowed.	Spanned records allowed.	No spanned records.

How to Create a VSAM Data Set

This example creates a new VSAM KSDS data set by defining the attributes and copying the data from an existing VSAM KSDS data set.

```
station%  cp ksds.old "ksds.new1,model(ksds.old)"
```

Figure 7-24. Creating a VSAM KSDS Data Set

In the following example, the attributes indicate that

- Spanned records are allowed
- Organization is key-sequenced
- Keys are 8 bytes long and start in position 0 of each record
- Average record size is 1028
- Maximum record size is 4096
- Space is allocated for 50 records with a secondary allocation of 10
- Cross region and cross system share options are provided
- Data set is to be created on a volume named D80CAT.

```
station%  cp ksds.old "ksds.new2,spanned,dsorg(indexed),keys(8,0),
          recordsize(1028,4096),space(50,10),shareoptions(1,3),
          vol(D80CAT)"
```

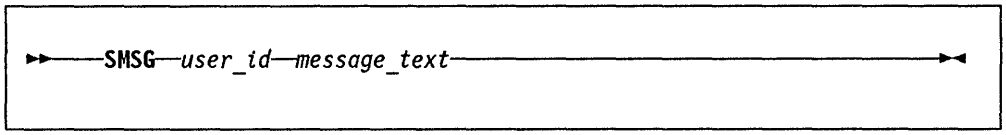
Figure 7-25. Creating a VSAM KSDS Data Set (2nd example)

Appendix A. Miscellaneous Commands

This appendix describes miscellaneous TCP/IP for MVS commands.

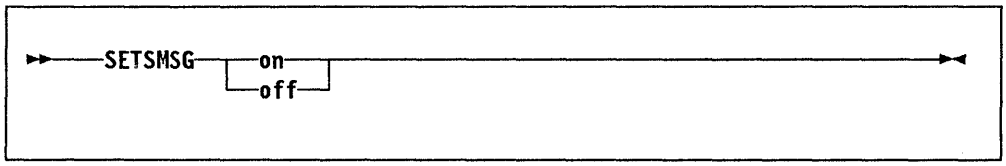
Sending and Receiving Special Messages

The SMSG command is used to send a special message (SMSG) to a user program, a started task, or a batch job. The format of the SMSG command is as follows:



Parameter	Description
user_id	Is the name of the user program, the started task, or the batch job to receive the SMSG.
message_text	Is text of the SMSG to be sent.

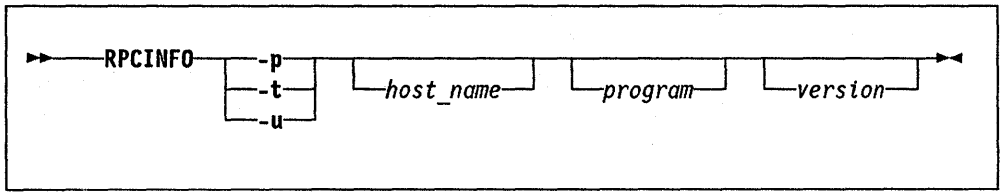
If your application program accepts and processes SMSGs, you must enable the reception of SMSGs before using the services provided by the application program. By default, reception of SMSGs is disabled. The SETSMSG command is used to enable and disable the reception of SMSGs. The format of the SETSMSG command is as follows:



Parameter	Description
on	Enables the reception of SMSGs.
off	Disables the reception of SMSGs. This is the default.

Obtaining Remote Procedure Call Status Information

Use the RPCINFO command to obtain status information about Remote Procedure Call (RPC) services provided by a specified host. The format of the RPCINFO command is as follows:



Parameter	Description
-p	Is used to query a host's PORTMAP service. A list of the registered RPC programs that reside on the host is displayed. If a host is not specified, RPCINFO queries your local host.
-t	Is used to determine the status of procedure 0 of an RPC program on a specified host. The status of the procedure is displayed. TCP is used as the transport layer.
-u	Is used to determine the status of procedure 0 of an RPC program on a specified host. The status of the procedure is displayed. UDP is used as the transport layer.
host_name	Is the host name of the RPC server.
program	Is the number of the program used by the remote procedure.
version	Is the program version number. If you do not specify this parameter, RPCINFO searches for version 1 only.

Appendix B. Network File System Messages for the Client

This appendix contains the listing of the error messages that the Network File System server feature of IBM TCP/IP for MVS sends to the clients.

Table B-1 (Page 1 of 3). Network File System Server Messages for the Client

Error Message	Possible Causes	Action
1 NFSERR_PERM - Not Owner	1. Data set not in use, but has not timed out yet. 2. Data set is busy. Another client (maybe even same client) has the data set open for writing. 3. The client attempted to change the mode in a nfsattr Network File System procedure call.	• For (1), follow the steps in waiting and retrying described in the section of NFSERR_ACCES. • For (3), don't do that.
2 NFSERR_NOENT - No such file or directory	A locate (SVC26) failed for this data set. The data set is not cataloged.	Check your spelling. If it is correct, contact your system administrator.
5 NFSERR_IO - I/O error	1. Unexpected error from SVC26. 2. SVC99 failed during action other than read or write. 3. A data set could not be opened during action other than a read or write. 4. An error occurred while reading a PDS directory. 5. An attempt has been made to nest PDSs. 6. No space available in TIOT. 7. TIOT resource unavailable. 8. Unable to release enough resources. 9. Maximum number of file allocations exceeded. 10. Insufficient units available. 11. The server could not get enough memory to perform this function.	This action is undertaken by the MVS system administrator. For (11), stop the server and change the region field in the JCL before restarting, or modify parameters in the attributes file.
6 NFSERR_NXIO - No such device or address	A locate succeeded for this data set but an obtain failed. The data set is cataloged but could not be found on the disc.	The data set may be archived. Alert the system administrator.

Table B-1 (Page 2 of 3). Network File System Server Messages for the Client

Error Message	Possible Causes	Action
13 NFSERR_ACCES - Permission denied	1. Data set not in use, but has not timed out yet (occurs most often when writing PDS members). 2. Data set in use by an MVS user, or another client. 3. Authorized function requested. 4. RACF not active. 5. Request denied by operator. 6. Installation supplied exit rejected; SVC99 attempt. 7. Access privilege refused - returned from security exit. 8. Mount processing is suspended (only if seen after a mount attempt). 9. The data set/prefix is not exported to your client (only if seen after a mount attempt). 10. IDCAMS failed during rename or remove. Usually this happens because the data set is busy. The output from IDCAMS is printed to the system log.	1. To check (1), retry your action after p seconds (p is the shortest timeout value). 2. If you get this error again, retry after p' seconds (p' is the next shortest timeout value). 3. If you get this error again, retry after p" seconds (p" is the longest timeout value).
17 NFSERR_EXIST - File exists	An attempt has been made to rename a PDS and the new name is the name of an existing file.	Delete the file with the new name before renaming a PDS. This is not required for a regular file or a PDS member.
18 EXDEV - Cross device link	This error is seen when a rename is attempted using one of the forbidden combinations: • MEM of PDS, MEM of different PDS • MEM of PDS, other kind of data set • other kind of data set, MEM of PDS.	Try a copy and remove instead of rename.
19 NFSERR_NODEV - No such device	No catalog mounted.	Point this out to your system administrator.
20 NFSERR_NOTDIR - Not a directory	A directory operation has been attempted on a file that is not a PDS.	Don't do that.

Table B-1 (Page 3 of 3). Network File System Server Messages for the Client

Error Message	Possible Causes	Action
21 NFSERR_ISDIR - Is a directory	A non-directory operation has been attempted on a PDS.	Don't do that.
28 NFSERR_NOSPC - No space left on device	1. The data set has exceeded the space allocated to it. 2. The PDS has exceeded the space allocated to it.	1. For (1), save this file into a new data set and then rename it to the old name. 2. For (2), create a new PDS and store this member there. You can use create attributes to give your new PDS a larger directory.
30 NFSERR_ROFS - Read only file system	One of the following Network File System procedures was attempted on a read only file system: link, write, rename, remove, mkdir, create.	See the documentation on export control and the exports file to see how a file system is designated read-only (see *IBM TCP/IP for MVS: Installation and Maintenance* book). You may either be doing things wrong or the exports file needs to be changed.
63 NFSERR_NAMETOOLONG - File name too long	The name is not a legal data set or member name.	
66 NFSERR_NOTEMPTY - Directory not empty	An attempt has been made to remove a PDS that has members.	
70 NFSERR_STALE - Stale NFS file handle	A file handle is used by the client and server sides of the Network File System to specify a particular file or prefix. A stale file handle occurs when the name is no longer valid, possibly due to: 1. The file or prefix has been removed. 2. The server has been stopped and brought back up. This affects data sets and members below mount points.	Unmount and re-mount. If your client maintains that the device is busy though it is not, you may have to re-boot your client.
EWEAKAUTH - Weak authorization	The authorization data in the RPC message was invalid. This is a client side error.	UNIX style authorization is required.

Other Possible Reasons for I/O Errors

1. The file is being written in text mode, and the beginning offset is determined to fall within the end-of-record (EOR) sequence. The user supplied data does not contain the correct EOR characters.
2. The file is being written in text mode, with a non-zero EOR (LF, CR, LFCR, CRLF). The number of bytes of data in the line written is larger than the maximum record size of the file.
3. The file is being written in text mode, with non-zero record indicator, stripblanks not set (no padding blanks on the write), and the number of bytes of data in the line written is less than the record size of the file.
4. The file is being written is in text mode, fixed records, with stripblanks set, and the line of data written contains trailing blanks as part of the data.

Appendix C. Other Related Publications

The following publications are used as the protocol specifications:

User Datagram Protocol, RFC 768, J. Postel

Trivial File Transfer Protocol, RFC 783, K.R. Sollins

Internet Protocol, RFC 791, J. Postel

Internet Control Message Protocol, RFC 792, J. Postel

Transmission Control Protocol, RFC 793, J. Postel

Simple Mail Transfer Protocol, RFC 821, J. Postel

Standard for the Format of ARPA Internet Text Messages, RFC 822, David H. Crocker

The DARPA Internet Gateway, RFC 823, R. Hinden, A. Sheltzer

An Ethernet Address Resolution Protocol, RFC 826, D. Plummer

Telnet Protocol Specification, RFC 854, J. Postel, J. Reynolds

Telnet Binary Transmission, RFC 856, J. Postel, J. Reynolds

Telnet Echo Option, RFC 857, J. Postel, J. Reynolds

A Standard for the Transmission of IP Datagrams over Public Data Networks, RFC 877, J.T. Korb

Telnet End Of Record Option, RFC 885, J. Postel

Telnet Terminal Type Option, RFC 930, M. Solomon, E. Wimmers

Internet Standard Subnetting Procedure, RFC 950, J. Mogul, J. Postel

DOD Internet Host Table Specification, RFC 952, K. Harrenstien, M. Stahl, E. Feinler

File Transfer Protocol, RFC 959, J. Postel

Mail Routing And The Domain System, RFC 974, C. Partridge

Assigned Numbers, RFC 1010, J. Reynolds, J. Postel

Official ARPA Internet Protocols, RFC 1011, J. Reynolds, J. Postel

X Window System Protocol, Version 11, RFC 1013, R. Scheifler

XDR: External Data Representation Standard, RFC 1014, SUN Microsystems Incorporated

Domain Names - Concepts and Facilities, RFC 1034, P. Mockapetris

Domain Names - Implementation and Specification, RFC 1035, P. Mockapetris

Internet Protocol on Network Systems HYPERchannel Protocol Specification, RFC 1044, K. Hardwick, J. Leckashaman

Remote Procedure Call Protocol Specification, RFC 1057, SUN Microsystems Incorporated

Network File System Protocol Specification, RFC 1094, SUN Microsystems Incorporated.

These RFCs are all part of a distribution package contained in the **tcpip.arfc(rfc*nnnn*)** data set; where the *nnnn* in the member name is the number of the RFC. Other documents may be obtained from:

SRI International
DDN Network Information Center
Room EJ291
333 Ravenswood Avenue
Menlo Park, CA. 94025
1-800-235-3155

You may also obtain the RFCs from the Internet host with a domain name of **sri-nic.arpa**. Use the FTP command and the appropriate FTP subcommands to retrieve the files. Use a user ID of **guest** and supply a password of **anonymous** to the FTP server on that host. The files to receive are:

< rfc > rfc-index.txt

< rfc > rfc*nnnn*.txt

Glossary

This glossary defines the most common terms associated with TCP/IP communication in an internet environment.

A

access method. A mainframe data management routine that moves data between storage and an I/O device in response to requests made by a program.

active open. The state of a connection that is actively seeking a service.

address. The unique identifier assigned to each device or workstation connected to a network.

address space. The complete range of addresses in memory available to a computer program.

AIX. Acronym for Advanced Interactive Executive, IBM's licensed version of the UNIX operating system.

API. Acronym for Application Program Interface, the formally-defined programming language interface between an IBM system control program or licensed program and the user of the program.

ARP. Acronym for Address Resolution Protocol, a protocol used to dynamically bind an internet address to a hardware address. ARP is implemented on a single physical network, and is limited to networks that support broadcast addressing.

ARPA. Acronym for Advanced Research Projects Agency, the former name for DARPA. See DARPA.

ARPANET. A proprietary TCP/IP-based internetwork funded by United States Department of Defense.

ASCII. Acronym for American National Standard Code for Information Interchange, the standard code, using a coded character set consisting of 7-bit coded characters, used for information exchange among data processing systems, data communication systems, and associated equipment.

B

backbone. In a wide area network, a high speed link to which nodes or data switching exchanges are connected.

baseband. A frequency band that uses the complete bandwidth of a transmission. All the stations on the network must participate in every transmission. See also *broadband.*

block. A string of data elements recorded, processed or transmitted as a unit. The element may be characters, words or physical records.

bridge. A functional unit that connects two LANs that use the same logical link control procedure, but may use different medium access control procedures.

broadband. A frequency band divisible into several narrower bands that uses analog signals, carrier frequencies and multiplexing techniques to allow simultaneous communication by more than one process through a single connection.

broadcast. The transmission of data packets to all nodes on a network or subnetwork simultaneously.

broadcast address. An address that is recognized by all nodes on a network.

bus topology. A network configuration in which only one path is maintained between stations and any data transmitted by a station is available concurrently to all other stations on the link.

C

checksum. The sum of a group of data associated with the group and used for error-checking purposes.

Class A network. An internet network in which the high-order bit of the address is 0. The host number occupies the 3 low-order octets, allowing for 128 class A networks with 16 777 216 host numbers on each network.

Class B network. An internet network in which the high-order bit of the address is 1 and the next high-order bit is 0. The host number occupies the 2 low-order octets, allowing for 16 384 class B networks with 65 536 host numbers on each network.

Class C network. An internet network in which the 2 high-order bits of the address are 1 and the next high-order bit is 0. The host number occupies the low-order octets, allowing for 2 097 152 class C networks with 256 host numbers on each network.

client. (1) A function that requests services from a server, and makes them available to the user. (2) An address space in MVS that is using TCPIP services. (3) A term used in an environment to identify a machine that uses the resources of the network.

client-server relationship. Any process that provides resources to other processes on a network is a *server*. Any process that employs these resources is a *client*. A machine can run client and server processes at the same time.

connection. An association established between functional units for conveying information.

D

daemon. A background process usually started at system initialization that runs continuously and performs a function required by other processes.

DARPA. Acronym for Defense Advanced Projects Research Agency, the United States Department of Defense agency responsible for creating ARPANET. Formerly called ARPA.

datagram. The basic unit of information that is passed across an internet. It consists of one or more data packets.

data set. The major unit of data storage and retrieval in MVS, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set organization. The way in which data is arranged within a data set on a mainframe. Only sequential, direct, partitioned, and VSAM data set organizations are supported by the Network File System server feature of TCP/IP for MVS.

DDN. Acronym for Defense Data Network. It is sometimes used to refer to the collection of X.25 networks that include MILNET and ARPANET, but more accurately refers to MILNET and its interconnected military networks.

direct data set. A type of data set used in a mainframe environment for storing data on a random access device that is accessed using a record address.

Distributed Services. A facility that provides a stateful architecture for transparent file sharing, file-level remote mounts, inherited mounts, cross-system file locking, and for local/remote process transparency of Inter-Process Communications message queues. It is supported only on AIX RT and AIX PS/2 systems.

domain name. Part of the naming hierarchy used in an internet. It contains a sequence of names (labels) separated by periods (dots).

domain naming. A hierarchical system for naming network resources.

dotted-decimal notation. A representation for a 32-bit integer consisting of four 8-bit numbers, written in base 10, and separated by periods (dots). Dotted decimal notation is accepted by many Internet application programs (instead of machine names).

DS. See Distributed Services.

E

EBCDIC. Acronym for Extended Binary Coded Decimal Interchange Code, a coded character set consisting of 8-bit coded characters.

entry-sequenced data set. A type of data set used in a mainframe environment. The format consists of logical records sequenced by the time of their arrival. A particular record is located by using the relative byte address (RBA).

ESDS. Acronym for Entry-Sequenced Data Set.

Ethernet. The name given to a local area packet-switched network technology invented in the early 1970s by the Xerox Corporation.

exit. A mechanism that provides an interface from a server application into a function. Exits are used in the Network File System server feature of TCP/IP for MVS to provide RPC services.

F

foreign host. Any host on the network other than the local host.

foreign network. In an internet, any other network interconnected to the local network by one or more intermediate gateways.

foreign node. See *foreign host.*

FTP. Acronym for File Transfer Protocol, a TCP/IP protocol used for transferring files to and from foreign hosts. FTP also provides the capability to access directories. Password protection is provided as part of the protocol.

G

gateway. A functional unit that interconnects computer networks of different architectures and protocols (at the IP layer).

H

hop count. The number of networks through which a datagram passes on the way to its destination node.

host. A computer connected to a network, and providing an access method to that network. A host provides end-user services.

I

ICMP. Acronym for Internet Control Message Protocol. It is included in IP, and handles error and control messages.

IEEE. Acronym for Institute of Electrical and Electronics Engineers.

internet. See *internetwork.*

internetwork. A collection of packet-switched networks that are connected by gateways. They function as single network.

Internet. A specific internetwork that includes ARPANET, MILNET and NSFnet. These networks use the TCP/IP protocol suite.

internet address. The unique 32-bit address identifying a node on an internetwork.

interoperability. The ability of hardware and software from multiple vendors to communicate on a network.

IP. Acronym for Internet Protocol, the TCP/IP layer between the higher-level host-to-host protocol and the local network protocols. IP uses local area network protocols to carry packets, in the form of datagrams, to the next gateway or destination host.

ISO. Acronym for International Standards Organization, an organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

IUCV. Acronym for Inter-User Communication Vehicle, a communication mechanism between address spaces.

J

JES. Acronym for Job Entry Subsystem, a system facility for spooling, job queuing, and managing I/O.

K

KB. Kilobyte; 1024 bytes.

kernel. A master program that manages all the physical resources of the computer, including file system management, virtual memory, reading and writing files to disks and tapes, scheduling of processes, printing, and communicating over a network.

key-sequenced data set. A type of data set used in a mainframe environment for sorting data on a random access device. The format consists of an index followed by one or more logical records.

KSDS. Acronym for Key-Sequenced Data Set.

L

LAN. Acronym for Local Area Network, a data network located on the user's premises in which serial transmission is used for direct data communication among data stations.

local host. The computer to which a user's terminal is directly connected.

local network. That portion of a network physically connected to the host without intermediate gateways.

M

MCH. Acronym for Multichannel Link.

mount. (1) The process of accessing a directory from a disk attached to the machine making the mount request (4.2 mount), or to the remote disk on a network (Network File System mount). (2) An operation that associates a group of files on a server with a directory (mount point) on a client to provide transparent access to the files through that directory. The files must be in a hierarchical arrangement.

mount point. A place established in a workstation or server local directory that is used during the transparent accessing of a remote file. Two entries must be created; first, an entry in the /ETC/FSTAB file and, second, an empty directory must be created in a local directory, quite often the /USR directory.

multichannel link. A means of enabling a data terminal equipment (DTE) to have several access channels to the data network over the single circuit.

N

name server. The server used for cross-referencing a name with its corresponding internet address.

national characters. The characters $, #, and @.

NCP. Acronym for Network Control Program, an IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

network. An arrangement of nodes and connecting branches.

Network File System. (1) A generic term for a system based on the NFS 3.2 protocol. (2) A facility for sharing files in a heterogeneous environment of machines, operating systems, and networks.

NFS 3.2. A protocol developed by SUN Microsystems Incorporated. It allows computers on a network to access each other's file systems. Once accessed, the file system appears to reside on the local host. NFS 3.2 uses IP.

NJE. Acronym for Network Job Entry, a batch networking application that transmits data between IBM operating systems.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (2) In a network topology, the point at an end of a branch.

NPSI. Acronym for NCP Packet Switching Interface, an IBM program product that provides NCP users with the capability of attaching IBM communications controllers to data transmission services that support X.25 interfaces.

O

obey list. The list of user IDs that is authorized to perform privileged functions in the TCPIP address space.

octet. A byte composed of eight binary elements.

OSI. (1) Acronym for Open Systems Interconnection, the interconnection of open systems in accordance with specific ISO standards. (2) The use of standardized procedures to enable the interconnection of data processing systems.

P

packet. A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

passive open. The state of a connection that is prepared to provide a service on demand.

PDN. Acronym for Public Data Network, a network established and operated by a telecommunication administration or by a Recognized Private Operating Agency (RPOA) for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public.

partitioned data set. See PDS.

PDS. A type of data set used in the mainframe environment. It must be on a direct access volume and consists of members. It has a directory that points to the locations of the various files stored in this data set. Often used to store libraries of programs and macro instructions.

peer. In network architecture, any functional unit that resides in the same layer as another entity.

PING. The process of sending an ICMP Echo Request packet to a host or gateway, with the expectation of receiving a reply.

POP. Acronym for Post Office Protocol, a protocol that allows an AIX RT or AIX PS/2 host to act as the receiver for mail destined for a user of TCP for the PS/2 computer.

portmapper. A server that converts RPC program numbers into port numbers acceptable to the protocol. This server must be running to make RPC calls.

port. (1) An endpoint for communication between devices, generally referring to a physical connection.

(2) A 16-bit number identifying a particular TCP or UDP resource within a given TCP/IP node.

process. (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. (2) Any operation or combination of operations on data. (3) A function being performed or waiting to be performed. (4) A program in operation. For example, a daemon is a system process that is always running on the system. (If it stops running, you have to start it up.)

PROFS. Acronym for Professional Office Systems, IBM's proprietary integrated office management system used for sending, receiving, and filing electronic mail, and a variety of other office tasks.

protocol. A set of semantic and syntactic rules that defines the behavior of functional units in achieving communication.

R

RACF. Acronym for Resource Access Control Facility, a facility that allows access to data and system components based on authorization levels.

RBA. Acronym for Relative Byte Address, an address which may be used in accessing key sequenced or entry sequenced VSAM data sets.

relative record data set. A type of data set used in the mainframe environment. It must be on a direct access volume and the format consists of one logical record in a fixed-length slot. Each slot has a unique relative record number. Data is placed in a specific slot based on a user-supplied relative record number.

remote host. See *foreign host.*

remote spooling communications subsystem. A VM networking component that provides telecommunication facilities for the transmission of bulk files between VM users and remote stations.

resolver. A program or subroutine that obtains information from a name server for use by the calling program.

RFC. Acronym for Request For Comments, a series of documents that address a broad range of topics affecting internetwork communication. Some RFCs are established as internet standards.

ring topology. A network configuration in which devices are connected by unidirectional transmission links to form a closed path.

router. A device that connects networks at the physical network layer. It is protocol-dependent and connects only networks operating the same protocol. Routers do more than transmit data; they also select the best transmission paths and optimum sizes for packets.

routing table. A list of network numbers and the information needed to route packets to each.

RPC. Acronym for Remote Procedure Call, a facility that a client uses to have a server execute a procedure call. This facility is composed of a library of procedures plus an XDR.

RRDS. Acronym for Relative Record Data Set.

RSCS. Acronym for Remote Spooling Communications Subsystem.

S

segmentation. The process of dividing a unit of data into smaller units in order to send it across a network. Usually this is done at a gateway when the incoming data buffer is too large to be transmitted to the next network.

sequential data set. A type of data set used in the mainframe environment. It must be on a direct access volume and has the records stored and retrieved according to their physical order within the data set.

server. (1) A function that provides services for users. A machine may run client and server processes at the same time. (2) A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service.

sharing. A term used in a computing environment to refer to utilizing a file on a remote system. It is done by mounting the remote file system, then reading or writing files in that remote system.

SMF. Acronym for System Management Facility, a facility used on the mainframe to log accounting information, which includes processor time, data transfer statistics, as well as user information.

SMTP. Acronym for Simple Mail Transfer Protocol, a TCP/IP application protocol used for transferring mail between users on different systems.

SNA. Acronym for Systems Network Architecture, the description of a logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

socket. (1) An endpoint for communication between processes or applications in the C Socket API of TCP/IP for MVS. (2) A pair consisting of TCP port and IP address, or UDP port and IP address.

star topology. A network configuration in which all nodes are connected to a central controller or computer that transfers data between nodes.

stream. A continuous sequence of data elements transmitted in character or binary-digit form using a defined format.

subnet. A networking scheme that divides a single logical network into smaller physical networks to simplify routing.

subnet address. The portion of the host address that identifies a subnetwork.

subnet mask. A mask used in the IP protocol layer to separate the subnet address from the host address.

SVC. Acronym for Supervisor Call, the macro instruction used by the mainframe to generate a software interrupt. Control is then transferred to a routine that will handle the interrupt processing.

switched virtual circuit. A virtual circuit that is requested by a virtual call. It is released when the virtual circuit is cleared.

system catalog. The highest level catalog on a mainframe that must exist so that the operating system can find data files. It is a VSAM data set and can contain pointers to VSAM data sets, VSAM user catalogs, OS data sets, and OS user catalogs.

T

TCB. Acronym for Transmission Control Block, an internal control block within the TCPIP address space.

TCF. Acronym for Transparent Computing Facility, a facility that allows a cluster of AIX/370 and AIX PS/2 systems to be constructed, allowing centralized administration of user logons, passwords, and system resources across the entire cluster.

TCP. Acronym for Transmission Control Protocol, a stream communication protocol that includes error recovery and flow control.

TCP/IP. Acronym for Transmission Control Protocol/Internet Protocol, a suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network.

Telnet. Terminal Emulation Protocol, a TCP/IP application protocol that allows interactive access to foreign hosts.

token. (1) In a local network, the symbol of authority passed among data stations to indicate the station temporarily in control of the transmission medium. (2) In programming languages, a language construct that by convention represents an elemental unit of meaning.

TFTP. Acronym for Trivial File Transfer Protocol, the TCP/IP standard protocol for file transfer used primarily for communications among PS/2 computers. TFTP allows sending and receiving of files, but does not provide any password protection or directory capability.

TN3270. An informally defined protocol for transmitting 3270 data streams over Telnet.

Token-Ring network. A ring network that allows unidirectional data transmission between data stations by a token passing procedure over one transmission medium so that the transmitted data returns to the transmitting station.

U

UDP. Acronym for User Datagram Protocol, a connectionless datagram protocol that requires minimal overhead, but does not guarantee delivery.

user. Anyone who requires the services of a computing system.

USS. Acronym for Unformatted System Services.

V

VMCF. Acronym for Virtual Machine Communication Facility, a connectionless mechanism for communication between address spaces.

VSAM. Acronym for Virtual Storage Access Method, an access method used on a mainframe to organize data and maintain information about that data in a catalog. VSAM data sets cannot be accessed by any other access method.

Virtual Telecommunications Access Method. An IBM program product that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. VTAM runs under MVS, VSE, and VM.

virtual circuit. (1) In packet switching, the facilities provided by a network that give the appearance to the

user of an actual connection. (2) A logical connection established between two DTEs.

VTAM. Acronym for Virtual Telecommunications Access Method.

W

WAN. Acronym for Wide Area Network, a network that provides communication services to a geographic area larger than that served by a local area network.

widget. (1) The fundamental data type of the X-Windows Toolkit. (2) An object providing a user-interface abstraction; for example, a Scrollbar widget. It is the combination of an X-Windows window (or subwindow) and its associated semantics.

working directory. A collection of files to be manipulated by an FTP operation.

X

X.25. A recommendation of the Consultative Committee on International Telephony and Telegraphy (CCITT) that defines the interface between data terminal equipment and packet switching networks.

XDR. Acronym for External Data Representation, a standard developed by SUN Microsystems Incorporated for representing data in machine independent format.

X-Windows API. An application program interface designed as a distributed, network-transparent, device independent, multitasking windowing and graphics system.

X Window System. An application developed by the Massachusetts Institute of Technology that incorporates the protocol also used in IBM's X-Windows API.

Index

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Special Characters

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office. You may use this form to communicate your comments about this publication, its organization, or subject matter **with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.**

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name _____

Company or Organization _____

Address _____

_____

SC09-1255-00

Cut or Fold Along Line

Reader's Comment Form

Fold and tape | **Please Do Not Staple** | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1T
180 Kost Road
Mechanicsburg, Pennsylvania 17055

Fold and tape | **Please Do Not Staple** | Fold and tape

IBM®

SC09-1255-00

Printed in U.S.A.