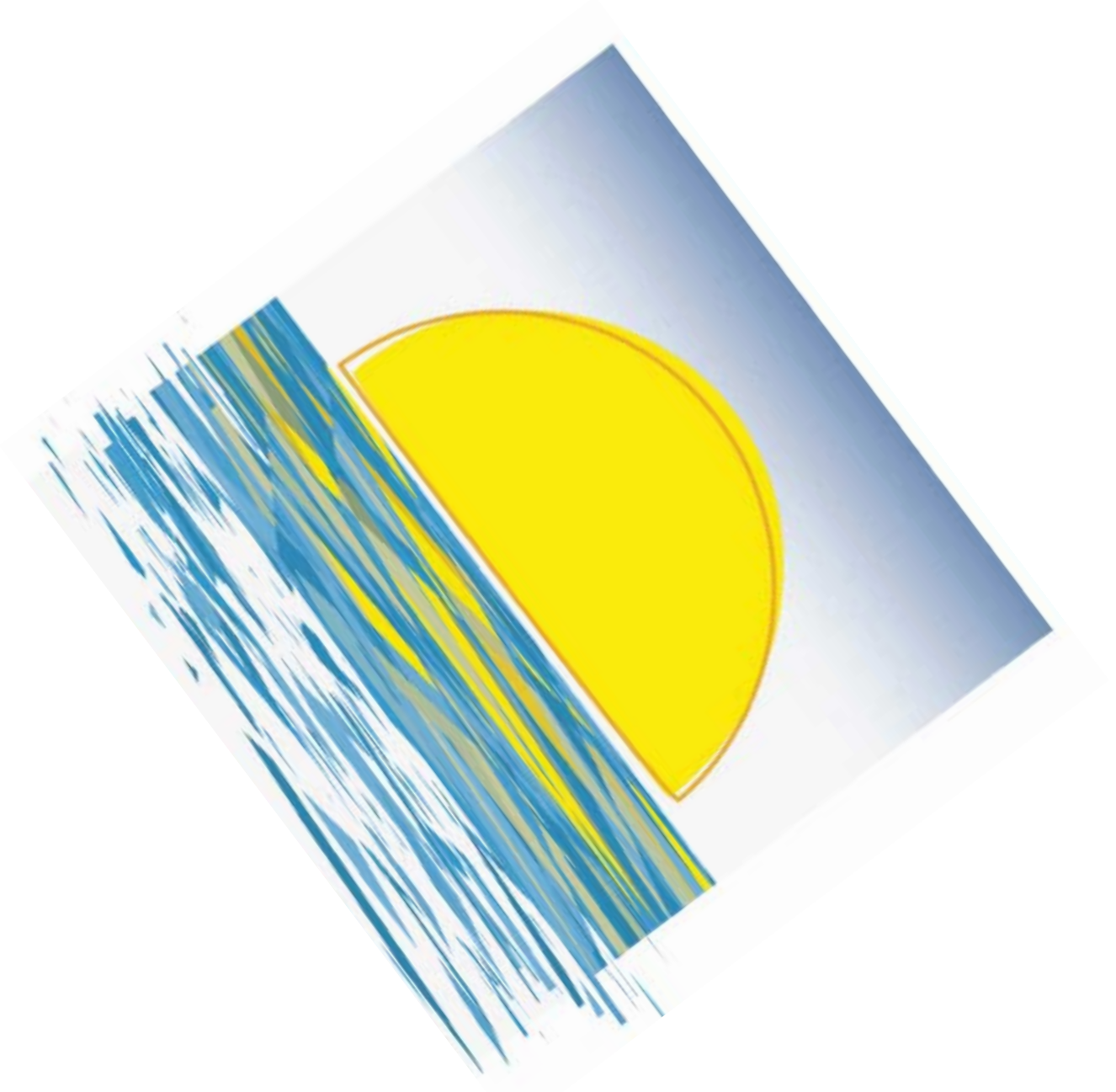
OS/390 TCP/IP OpenEdition

User's Guide



GC31-8305-00

OS/390 TCP/IP OpenEdition

User's Guide

GC31-8305-00

Note:

Before using this information and the product it supports, be sure to read the general information under Appendix H, “Notices” on page 267.

First Edition (June 1997)

This edition applies to OS/390 (5645-001) and OS/390 TCP/IP OpenEdition. See the “Summary of Changes” for a description of the changes made in this edition. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be at the back of this publication. If the form has been removed, you may send your comments to the following address:

International Business Machines Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
USA

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):	1-800-227-5088
Internet e-mail:	usib2hpd@vnet.ibm.com
World Wide Web:	http://www.s390.ibm.com/os390
IBMLink:	CIBMORCF at RALVM13
IBM Mail Exchange:	USIB2HPD at IBMMAIL

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

About This Book

This book describes how to use the applications available in OS/390 TCP/IP OpenEdition to do the following functions:

- Log on to a remote host
- Transfer data sets
- Run a command on another host
- Monitor the network
- Manage network resources

OS/390 TCP/IP OpenEdition is an integral part of the OS/390 family of products. For an overview and mapping of the documentation available for OS/390, see the *OS/390 Information Roadmap*.

Who Should Use This Book

This book is written for end users and system administrators who want to use the applications that are available in OS/390 TCP/IP OpenEdition.

Part 1 of this book provides information useful primarily to end users, while Part 2 provides information useful to system administrators. Part 3 provides reference information useful to both.

Before using this book, you should be familiar with the IBM Multiple Virtual Storage operating system (MVS) and the IBM Time Sharing Option (TSO). In addition, OS/390 TCP/IP OpenEdition should already be installed and customized for your network. For information about installing, see the Program Directory. For information about customizing, see the *OS/390 TCP/IP OpenEdition Configuration Guide*.

Where to Find Related Information on the Internet

You may find the following information helpful.

For current updates to the TCP/IP Version 3 Release 2 for MVS documentation described in “Bibliography” on page 269, check out the TCP/IP for MVS home page:

`http://www.networking.ibm.com/tcm/tcmprod.html`

To keep in close touch with OS/390, we suggest you look at the OS/390 home page:

`http://www.s390.ibm.com/os390`

To keep abreast of new products and technologies from IBM Networking, take a look at the IBM Networking home page:

`http://www.networking.ibm.com/`

The IBM Networking Software Glossary is now available in HTML format as well as PDF. You can access it directly at the following URL:

`http://www.networking.ibm.com/nsg/nsggls.htm`

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Summary of Changes

Summary of Changes
for GC31-8305-00

This is the first edition of this book. It contains information previously presented in the *TCP/IP for MVS: User's Guide*, SC31-7136-02, which supports TCP/IP Version 3 Release 2 for MVS. This book is new for OS/390 TCP/IP OpenEdition, which provides OpenEdition function for TCP/IP in the OS/390 environment. For information about previously available TCP/IP function, continue to use the TCP/IP Version 3 Release 2 for MVS library.

This book describes:

- Commands for OS/390 TCP/IP OpenEdition. Some of this information was previously found in *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, SC31-8069-00 and *TCP/IP for MVS: User's Guide*, SC31-7136-02.
- Information on using FTP and Telnet in an OE environment.
- Capability Statement.

Part 1. End-User Applications

Chapter 1. Getting Started

Transmission Control Protocol/Internet Protocol (TCP/IP) is a set of industry-standard protocols and applications that allow you to share data and computing resources with other computers, both IBM and non-IBM. By using TCP/IP commands at your workstation, you can perform tasks and communicate easily with a variety of other systems and workstations. TCP/IP allows you to perform tasks independent of the type of computer. TCP/IP for MVS lets you use the Time Sharing Option (TSO) to interactively run the TCP/IP applications (TCP/IP commands). OS/390 TCP/IP OpenEdition (OE) lets you use the OE shell to interactively run the OE TCP/IP applications (OE TCP/IP commands) described in this manual. TCP/IP commands are not OE shell commands.

Understanding TCP/IP

TCP/IP is a set of protocols and applications that allow you to perform certain computer functions in a similar manner independent of the types of computers or networks being used. When you use TCP/IP, you are using a network of computers to communicate with other users, share data with each other, and share the processing resources of the computers connected to the TCP/IP network.

A computer network is a group of computer nodes electronically connected by some communication medium. Each node has the hardware and the programs necessary to communicate with other computer nodes across this communication medium. The node can be a PC, workstation, microcomputer, departmental computer, or large computer system. The size of the computer is not important. The ability to communicate with other nodes is important.

Computer networks allow you to share the data and computing resources of many computers. Applications, such as departmental file servers, rely on networking as a way to share data and programs.

Many forms of communication media are available today. Each is designed to take advantage of the environment in which it operates. Communication media consist of a combination of the physical network used to connect the computer nodes and the language, or protocol, they use to communicate with each other.

Physical Network

A physical network consists of electrical wiring and components, such as modems, bridges, controllers, access units, telephone lines, fiber optic cables, and co-axial cables. These are used to connect the computer nodes together. The physical network can connect two nodes in a single room or thousands of nodes communicating across large geographic areas. The most common networks in use today are Local Area Networks (LANs) and Wide Area Networks (WANs). LANs cover a limited distance, generally one or two floors or buildings, while WANs, using telecommunication facilities, are used for longer distances.

Protocols

Network protocols are the rules that define how information is delivered between nodes. They describe the sequence and contents of the data exchanged between nodes on the network. Network protocols determine how a computer node functions during communication with another node, how data is enclosed to reach its destination safely, and what path it should follow. Protocols coordinate the flow of messages and can specify which node a message is destined for in the network. A variety of protocols are used to take advantage of the characteristics of each of the physical network types. The most common protocols are Ethernet, 802.3, Token-Ring, X.25, and System Network Architecture (SNA).

Network Devices

OS/390 TCP/IP OE supports the following types of network devices:

- 3172 LAN Channel Station (LCS)
- Channel-to-channel (CTC)
- Common link access to workstation (CLAW)

Addresses

A network address is a component of the communication network and is associated with both hardware and software. The address is the means by which the sending node selects the receiving node for data transfer. It is also used by the receiving node to recognize what data is destined for it. An address is a unique code assigned to every node on a network. But an address is formed differently for different protocols. The length, position, and method used to specify an address are unique for each protocol. A communication node using one protocol cannot recognize the address of another protocol.

Gateways

A network is designed and built using one physical network type and one protocol. All of the computers on that network can then communicate. Because computer nodes use different protocols based on which physical network type they are connected to, and because those using different protocols cannot communicate with each other, computers on different network types cannot share data and other computing resources.

A gateway is a network component that is physically connected to more than one network and can recognize the format of more than one network protocol. When required, a gateway translates the data transfer from the protocol of one network into the protocol of another. Computer nodes on two dissimilar networks can communicate through a gateway and share data and resources.

How TCP/IP Uses Networks

TCP/IP consists of a layered structure of protocols ranging from hardware-dependent programs to high-level applications. Each TCP/IP layer provides services to the layer above it and uses the services of the layer below it. The lowest layer, which is next to the physical layer, is not part of TCP/IP. This layer consists of existing protocols, such as Ethernet and Token Ring. TCP/IP uses the services of this layer to transport data across dissimilar networks, much like a gateway.

Local and Remote Nodes

A physical network is used by the hosts that reside on that network. Each host is a node on that network. A node is an addressable location in a communication network that provides host processing services. The intercommunication of these nodes gives rise to the concept of *local* and *remote* nodes. A *local node* pertains to a device, file, or host accessed directly from your system. A *remote node* pertains to a device, file, or host accessed by your system through the network.

Client and Server

A *server* is a computer that contains data or provides services to be used by other computers on the network. Some of the common server types are file, print, and mail servers. They allow your computer to share the data, devices, and resources of another computer. There are also servers that provide services to let you execute programs on a computer other than your own. This allows your computer to share the processing power of another computer. Servers are also known as *daemons*. Generally, a server runs continuously and can handle the requests of multiple *clients* simultaneously.

A *client* is a computer or a program that requests services or data from a server. A client could, for example, request a file that is located at the server be sent across the network to the client.

TCP/IP Addresses

An address allows data to be routed to the chosen destination. TCP/IP has an addressing scheme consisting of 32 binary bits that allows users and programs to identify a specific network and host with which to communicate. Each host in your network, as well as any other TCP/IP network you have access to, can be uniquely identified by its assigned address. A TCP/IP address is written in *dotted-decimal* notation. This scheme is numeric and consists of four groups separated by a period (.). For example 9.67.01.100 represents a single host on a single network. 193.05.86.09 represents another host on another network. Most TCP/IP commands require you to include the address of the remote host where the server you wish to access resides.

Network Names

There is an alternative to supplying a numeric address. Each host is also assigned at least one name. You can use the name, rather than the address, in TCP/IP commands. Your local host can resolve the name you supply in a command into the correct numeric address. The names are translated using either a translation file or an application known as a *name server*. Your ability to use *network names* depends on how your TCP/IP network has been designed and which features have been installed.

Ports and Port Numbers

The use of *ports* and their identifying numbers are an extension to the addressing scheme. Once the address is used to deliver data to the desired host on the network, the *port number* is used to identify the process for which the data is used. This allows one host to provide more than one service.

How you define the port number depends upon your configuration. If you are modifying your current TCP/IP for MVS so it will run concurrently with OS/390 TCP/IP

OE, you can use any combination of ports for the OE and non-OE functions, as long as all the ports are different and all the port numbers are within the acceptable range. For example, you can use the well-known ports for the non-OE functions, such as TELNET which is always *port 23*, and other port numbers for OE functions. Or you can use the well-known port numbers for OE functions, and the other port numbers as the non-OE functions.

If you are adding OS/390 TCP/IP OE image to your MVS system in addition to your non-OE TCP/IP stack, the paths are different and the port numbers on the OE and non-OE TCP/IP can be the same for the same functions.

TCP/IP assumes the well-known port number unless you specifically specify otherwise when entering a TCP/IP command. A port number is entered as a decimal number on TCP/IP commands. For those cases when you are requesting the services of a user-developed server, you need to know the port number of that server.

Understanding What You Can Do

The following table shows the commands used in the OE environment. The OE commands are case-sensitive and are described in detail in the following chapters.

In OS/390 TCP/IP OE, the OMVS commands are in lowercase. The meaning of a parameter depends on whether it is in uppercase or lowercase. The commands and parameters must be entered in the case shown in Table 1.

Using OE Commands

Table 1. OE Commands

User/Tasks	**OS/390 TCP/IP OE**	**Where Issued**
Network Administrator		
Provides status of local host	onetstat	OE shell
Determines accessibility of a foreign node (remote node)	oping	OE shell
Monitors routers and attached networks	osnmp	OE shell
Provides diagnosis information for network problems	otracert	OE shell
System Administrator		
Shows information about system resources	Display	MVS operator console
Provides device, gateway, and host status	onetstat	OE shell
Executes a command on a remote host	orexec	OE shell
Application Programmer		
Displays server information	orpcinfo	OE shell

These commands provide a set of basic functions that are available to you. These functions include:

- Logging on to other hosts
- Transferring data sets and files between hosts
- Using other hosts

- Displaying system information
- Testing network usability

Logging on to a Host Using OE TELNET

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

The MVS Telnet client does not run in the OE environment. The OE Telnet server provides access to the OE environment.

Connecting to the OE Telnet server from any client results in a session with the OMVS shell as if the user had entered OMVS from TSO in line mode or character mode. Once the OE Telnet session has been established, you can enter any OE command that can be issued from within the OMVS shell. For more information about OE commands, see the *OS/390 OpenEdition Command Reference*. The following figure illustrates what the client screen will look like after it is connected to the OE Telnet server:

```
EZYTE27I-login:user79
EZYTE28I-user79's Password:
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Developement Group, University of Waterloo,  1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

$
```

For information about initiating the OE Telnet server from an MVS client, see the *OS/390 TCP/IP OpenEdition Configuration Guide*.

Transferring Data Sets between Hosts

When data is created or stored at one host but is processed by another host, some method for transferring the data between hosts is necessary. TCP/IP provides a command for transferring data sets and files between hosts. The FTP command moves or copies data sets and files between hosts that have either similar or dissimilar file systems. The FTP command provides subcommands that allow you to change the local and remote directories, set the transmission character code, list remote files, delete remote files, and send and receive files between hosts. You can use FTP to perform SQL queries as well as submit jobs to JES for batch processing. It provides for security by requiring a user ID and password and allowing for remote logon and logoff. See Chapter 2, “Transferring Data using File Transfer Protocol (FTP) in an OE Environment” on page 13, for a complete list of FTP functions.

Using Other Hosts

Just as there are occasions when you want to transfer data to a host where it can be processed, there are also occasions when you want to process the data where it exists and send the processing results to another host. The data sets or files could be too large to transfer efficiently or all the data might be kept at one host for security reasons. The computing power necessary to perform some task could be more than your host is capable of or the only licensed copy of a required program might reside at some other host. TCP/IP provides a command that allow you to use the processing resources of other hosts. The orexec command allows you to send any command that is valid on the remote host and receive the results at the local host. A user ID and password provide security checking at the remote host. The command sent to the remote host must not require user interaction to complete. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179, for more information about orexec.

Displaying System Information

The Display command displays the status of the current TCP/IP images. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179, for more information about the Display command.

Testing Network Usability

A set of tools is provided to assist in the operation and management of TCP/IP segments of the network. Some of these tools are complex and are used by those responsible for network operation. Others, such as the oping command, are available to, and are useful for, any user on the network. The following commands are available to let you test the networks availability, usability, and responsiveness.

Command	Description
oping	Tests connectivity of the network or routing to a remote host. It also measures the response time for a message to travel through the network to a remote host and return. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179 , for more information about the oping command.
onetstat	Shows local and remote addresses, routing tables, and statistics for hardware and software. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179 , for more information about the onetstat command.
orpcinfo	Determines which remote procedure call (RPC) servers are registered at different hosts in the network. It is used primarily by system managers. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179 , for more information about the orpcinfo command.
osnmp	Provides a means for managing a multiprotocol, multivendor network environment. It allows management of network components, such as gateways, routers, and hosts. OE SNMP uses the OE shell to provide the end-user interface. See Chapter 5, “Managing TCP/IP Network Resources with OE SNMP” on page 199, for more detail on OE SNMP.

otracert
: Displays the route that a packet takes to reach the reach the requested host. See Chapter 4, “Monitoring the TCP/IP Network in an OE Environment” on page 179 , for more information about the otracert command.

What You Need to Get Started

TCP/IP is a part of your MVS system. To use it you need a TSO user ID and password. If you are already a TSO user, you can begin using TCP/IP.

In order to use OS/390 TCP/IP OE, you must be authorized to use the OE shell. For information about the OE shell, refer to the *OS/390 OpenEdition User's Guide*.

Ensure that you have the following before proceeding:

User IDs and passwords
: You should have a user ID and password for each host you intend to use that requires user authorization and authentication. This includes most hosts you use. Some hosts on a TCP/IP network use a user ID of `anonymous` and a password of `guest` to permit all interested parties access to data sets contained at that host, but that is the exception and not the rule.

Host names
: TCP/IP commands require that you know the name or dotted decimal TCP/IP address of the remote host you want to use.

Authorizations for data and programs
: Your ability to access data sets and programs on remote hosts depends on the data security system used by that host. You might require authorization by the Resource Access Control Facility (RACF) or other security programs before you can gain access to data sets, commands, or other resources on remote hosts.

Testing Commands with Loopback

In order to test your local machine, an address is reserved that always refers to your local host rather than any other hosts on a network. This class A network address is 127.0.0.1. You can also specify LOOPBACK as the host name.

You can use the loopback address with any TCP/IP command that accepts IP addresses, for example, oping. When you issue a command with the loopback address, the command is sent out from your local host’s client and continues until it reaches the IP layer on your local host. The command is then sent back to your local host’s server.

Note: Any command or data that you send using the loopback address never actually goes out on any network.

The information you receive indicates the state of your system and checks to ensure that the client and server code for the function you are testing is operating properly. You should see the same response as for a normal, successful command. If the client or server code is not operating properly, the same message that would be returned for an unsuccessful command is returned.

Chapter 2. Transferring Data using File Transfer Protocol (FTP) in an OE Environment

The OE File Transfer Protocol (FTP) server allows you to access files in the HFS as well as MVS data sets. You can use any FTP client, for example the MVS FTP client, to connect to the OE FTP server. The examples in this chapter use the MVS FTP client to show the interaction between a client and the OE FTP server.

The File Transfer Protocol (FTP) command allows you to transfer data sets between your local host and any host that supports TCP/IP. Using the FTP command and its subcommands, you can sequentially access multiple hosts without leaving the FTP environment.

This chapter describes how to use the FTP command and its subcommands. Specifically, FTP allows you to:

- Establish a connection to a remote host or your local host
- Identify yourself to the host
- Obtain status and system information about the host
- Work with directories on the remote host
- Work with directories on your local host
- Transfer data sets to and from the host
- Pass TSO commands to your local host
- Send information to the local and remote hosts
- Obtain assistance for the ftp subcommands
- Submit FTP jobs in batch
- Use the EXEC interface
- Interface with JES
- Perform DB2 SQL queries
- Transfer DBCS data

Using the FTP Command

Before transferring files between your local host and a remote host, or using any other FTP functions, you must use the FTP command to enter the FTP environment. For complete information about the FTP command, see “FTP Command—Enter the FTP Environment” on page 87.

Logging on to FTP

If you correctly specify a foreign host with the FTP command, you are prompted to identify yourself. The following is a sample of the information that is displayed after you successfully invoke the FTP command with *foreign_host* correctly specified.

```
MVS TCP/IP FTP V3R2
Connecting to MVS1 9.67.58.227, port 21
220-FTPD1 IBM MVS V3R3 at MVS1.TCPIPDEV.RALEIGH.IBM, 10:08:24 on 1997-01-08
220 Connection closes if idle for more than 5 minutes.
USER (identify yourself to the host):
```

After successfully identifying yourself, you are prompted for a password if the foreign host requires a password. If you enter the password correctly, you are connected to the foreign host.

Notes:

1. You can use the data set NETRC to automatically log on to a remote host. For information about using NETRC, see "NETRC Data Set."
2. Data integrity cannot be guaranteed when multiple users use the same user ID and password to access data sets contained at the target host.

For the procedure to enter the FTP environment using the FTP command, see "Establishing and Exiting a Connection" on page 15 for more information.

Allocating FTP Input and Output Data Sets

When you invoke the FTP command, a check is made to see if a data set is allocated to INPUT. If a data set is allocated, 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 OUTPUT. If so, all FTP prompts and replies are written to that data set rather than to your terminal.

The record length and block size of the output data set can be any size. If the logical record length of the output data set is less than 100 bytes, some messages could be truncated or wrapped around to the next line.

If you create INPUT and OUTPUT data sets, use the following guidelines:

- Specify the INPUT data set:
  - Record format=FB
  - Logical record length=80
  - Block size is a multiple of 80
- Specify the OUTPUT data set:
  - Record format=FB
  - Logical record length=160
  - Block size is a multiple of 160

NETRC Data Set

The *user_id*.NETRC data set provides you with an alternative to specifying your *user_id* and *password* as FTP parameters when you want to log on to a remote host. To define the *user_id*.NETRC data set, use the following format:

```
000001 MACHINE mvs1.tcp.raleigh.ibm.com LOGIN user28 PASSWORD user28
000002 MACHINE 9.67.112.25 LOGIN user28
```

The keywords **MACHINE**, **LOGIN**, and **PASSWORD** must be uppercase. The variables *user_ID* and *password* may be case sensitive, depending on the remote host.

To invoke the *user_id*.NETRC data set and automatically log on to the remote host named MVS1, enter the FTP command as shown in the following example:

```
  User:    ftp mvs1
System:
           EZA1450I MVS TCP/IP FTP V3R2
           EZA1554I Connecting to MVS1 9.67.112.25, port 21
           220-FTPD1 IBM MVS V3R3 at MVS1.TCP.RALEIGH.IBM.COM, 16:26:43 on 1997-01-08
           220 Connection will close if idle for more than 5 minutes.
           EZA1701I >>>USER user28
           331 Send password please.
           EZA1701I >>>PASS ********
           230 USER28 is logged on.  Working directory is "/u/user28".
           EZA1460I Command:
```

If you have connected to the OE server, the default working directory is the $HOME directory in the HFS.

For information about using the *user_id*.NETRC data set in a batch file, see “Submitting FTP Requests in Batch” on page 44.

Getting Help

The FTP subcommands for getting help with FTP subcommands are listed in Table 2.

Table 2. FTP Subcommands for Getting Help

Subcommand	Description	Page
?	Provides an introduction to using FTP	92
Help	Displays help information for FTP	92

Establishing and Exiting a Connection

You normally establish a connection to a foreign host when you invoke the FTP command with a *foreign_host* specified. If you are not successful in specifying a foreign host, or if you need to connect to a different foreign host, use the subcommands listed in Table 3.

Table 3. FTP Subcommands for Establishing and Exiting a Connection

Subcommand	Description	Page
ACcount	Sends host-dependent account information	71
CLose	Disconnects from the foreign host	80
Open	Opens a connection to a foreign host	123
PAss	Supplies a password to the foreign host	123
QUIt	Leaves the FTP command environment	126
User	Identifies you to a foreign host or changes your TSO user ID password	172

Example of Establishing and Exiting a Connection

This following example shows how FTP can be used between a local MVS host to a remote VM host, then the OPEN subcommand to a remote MVS host. The following internet addresses are used for the FTP examples:

1. MVS Host: 192.9.2.1, 192.9.2.2
2. VM Host: 192.9.2.4

The current host is 192.9.2.2.

```
System:   READY
  User:   ftp 192.9.2.4
System:
             MVS TCP/IP FTP V3R2
             Connecting to 192.9.2.4, port 21
             220-FTPSERVE at IBM VM V3R2 at APPLE.CORE.PIT, 09:28:02 EST MONDAY 01/06/97
             220 Connection will close if idle for more than 5 minutes.
             USER (identify yourself to the host):
  User:   vmuser
System:
             >>>USER vmuser
             331 Sent password please.
             Password:
             >>>PASS ********
             230 VMUSER logged in; working directory = VMUSER 191

             Command:
  User:   account
System:
            Usage: ACCT account-information

             Command:
  User:   acct vmuser
System:
              >>>ACCT ********
             230 You now have write permission to VMUSER 191

             Command:
  User:   close
System:
              >>>QUIT
             221 Quit command received. Goodbye.

             Command:
  User:   open 192.9.2.1
System:
            Connecting to 192.9.2.1, port 21
             220-FTPD1 IBM MVS V3R3 at TREEFROG.ABC.OZ, 09:38:06 on 1997-01-08
             220 Connection will close if idle for more than 5 minutes.

              Command:
  User:   user rtp
System:
            >>>USER rtp
              331 Send password please.
              Password:
              >>>PASS ********
```

```
              230 RTP is logged on.  Working directory is "/u/rtp".

              Command:
  User:   close
System:
            >>>QUIT
              221 Quit command received. Goodbye.

              Command:
  User:   quit
System:   READY
```

Obtaining Status and System Information

To retrieve and display status information about the local host and remote host, use the subcommands listed in Table 4.

Table 4. FTP Subcommands for Obtaining Status and System Information

Subcommand	Description	Page
DEBug	Toggles internal debug options	81
LOCSTat	Displays FTP status information for the local host	110
NOop	Checks whether the foreign host is still responding	122
STAtus	Displays status information for the foreign host	147
SYstem	Displays the name of the foreign host's operating system	166
TSO	Passes a TSO command to the local host TSO environment	167

Working with Directories on the Remote Host

To work with directories on the remote host, use the subcommands listed in Table 5.

Table 5. FTP Subcommands for Working with Directories on the Remote Host

Subcommand	Description	Page
CD	Changes the working directory	76
CDUP	Changes to the parent of the current working directory	78
CWd	Changes the working directory (Synonymous with CD)	76
DIr	Lists the directory entries for files on the foreign host	83
LS	Lists the names of files on the foreign host	113
MKdir	Creates a directory on the foreign host	118
PWd	Displays the name of the active working directory on the foreign host	125
RMDir	Removes a directory on the foreign host	129

Examples of the CD Subcommand

This example shows how to change and choose remote working directories and demonstrates how the OE server allows you to switch between the MVS and HFS environment. For more information on how to change the directory levels, see the CD, CDUP, and LCD subcommands.

```
System:     READY
  User:    ftp 9.67.113.24 621
System:
            MVS TCP/IP FTP V3R2
            Connecting to 9.67.113.24, port 621
            220-FTPD1 IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH.IBM.COM, 15:45:25
                on 1997-03-17.
            220 Connection will not timeout.
            USER(identify yourself to the host):
  User:
           user121
System:     >>>USER user121
            331 Send password please.
            Password:

            >>>PASS ********
            230 USER121 is logged on.  Working directory is "/u/user121".
            Command:
  User:   cd tcpip
System:
            >>>CWD tcpip
            250 HFS directory /u/user121/tcpip is the current working directory
            Command:

  User:    cd ..

System:     >>>CWD ..
            250 HFS directory /u/user121 is the current working directory
              Command:
  User:
           cd 'user121'
System:
           250 "'user121'" is working directory name prefix.
            Command:
```

Examples Showing the Differences between DIR and LS Output for HFS Directories

The examples in this section use the following internet addresses:

MVSXA2 : 9.67.113.25
MVSXA3 : 9.67.113.24

The current host is MVSXA2 (9.67.113.25). An FTP command is issued from 9.67.113.25 to 9.67.113.24.

```
System:     READY
  User:
           ftp 9.67.113.24
System:
           MVS TCP/IP FTP V3R2
```

```
            Connecting to 9.67.113.24, port 621
            220-FTPD1 IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH.IBM.COM, 15:45:25
                on 1997-03-17.
            220 Connection will not timeout.
            USER (identify yourself to the host):
  User:
           user121
System:
            >>>USER user121
            331 Send password please.
            Password:

            >>>PASS ********
           230 USER21 is logged on. Working directory is "/u/user121".
            Command:

  User:    dir

System:    >>>PORT 9,67,112,25,4,25
            200 Port request OK.
            >>>LIST
            125 List started OK.
            total 2736
            drwxr-xr-x    2  USER121  SYS1        0  Nov 20  18:15  IBM
            -rwxr-xr-t    2  USER121  SYS1   389120  Feb  5  16:03  ftpdka
            -rwxr-xr-t    2  USER121  SYS1   962560  Feb  5  16:04  ftpsrvka
            -rw-r-----    1  USER121  SYS1    11648  Jan 20  14:30  g.s
            drwxr-x---    3  USER121  SYS1        0  Oct 21  17:50  msg
            -rw-r-----    1  USER121  SYS1     1458  Jan 10  19:25  s.k
            drwxr-x---    2  USER121  SYS1        0  Feb  6  15:59  tcpip
            drwxr-x---    2  USER121  SYS1        0  Feb  6  17:29  test
            250 List completed successfully.

           Command:
  User:
           ls
System:
            >>>PORT 9,67,112,25,4,26
           200 Port request OK.
           >>>NLST
           125 List started OK.
           IBM
           ftpdka
           ftpsrvka
           g.s
           msg
           s.k
           tcpip
           test
           250 List completed successfully.
           Command:
```

DIR provides detailed information about the data sets under the remote working directory, while LS shows the data set names only.

Examples Showing the Differences between DIR and LS Output With DIRECTORYMODE and DATASETMODE for MVS

This section gives examples on issuing a DIR and LS command in both DIRECTORYMODE and DATASETMODE:

```
System:     READY
  User:    ftp 1.1.2.3
System:
           MVS TCP/IP FTP V3R2
            Connecting to 1.1.2.3, port 21
            220-FTPD1 IBM MVS V3R3 at EMU.ABC.OZ, 15:00:17 on 01/08/97
            220 Connection will close if idle for more than 5 minutes.
            USER (identify yourself to the host):
  User:
           mvsuser
System:
            >>>USER mvsuser
            331 Send password please.
            Password:
            >>>PASS ********
            230 MVSUSER is logged on.   Working directory is "/u/mvsuser"
            Command:
  User:
           cd 'tcpv3'
System:
            >>>CWD 'tcpv3'
            257 "'TCPV3.'" is working directory name prefix.
            Command:
  User:
           site directorymode
System:
            >>>SITE directorymode
            200 Site command was accepted
            Command:
  User:
           dir
System:
            >>>PORT 1,1,2,2,4,39
            200 Port request OK.
            >>>LIST  125 List started OK.
          Volume Unit       Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
          Pseudo Directory                                            ETC
          Pseudo Directory                                            FTP
          Pseudo Directory                                            HOSTS
          Pseudo Directory                                            NSMAIN
          Pseudo Directory                                            PROFILE
          Pseudo Directory                                            STANDARD
          Pseudo Directory                                            TCPIP
          Pseudo Directory                                            TCPIPL62
          Pseudo Directory                                            TELNET
          250 List completed successfully.

          Command:
  User:   site datasetmode
System:
            >>>SITE datasetmode
```

```
           200 Site command was accepted

          Command:
  User:   dir
System:
           >>>PORT 1,1,2,2,4,40
           200 Port request OK.
           >>>LIST
           125 List started OK.
           Volume Unit     Date  Ext Used Recfm Lrecl BlkSz Dsorg Dsname
           APCSPL 3380D  07/16/93  1    1  FB      80  8800  PS  ETC.RPC
           APCSPL 3380D  08/03/93  1    1  FB      80  3200  PS  ETC.SERVICES
           APCSPL 3380D  08/03/93  1    1  FB      80  3120  PS  FTP.DATA
           APCSPL 3380D  08/02/93  1    1  F      158   158  PS  HOSTS.ADDRINFO
           APCSPL 3380D  08/03/93  1    1  FB      80  3120  PS  HOSTS.LOCAL
           APCSPL 3380D  07/30/93  1    1  F       56    56  PS  HOSTS.SITEINFO
           APCSPL 3380D  07/15/93  1    1  FB      80  8800  PS  NSMAIN.CACHE
           APCSPL 3380D  07/28/93  1    1  FB      80  8800  PS  NSMAIN.DATA
           APCSPL 3380D  08/03/93  1    2  FB      80  3200  PS  PROFILE.TCPIP
           APCSPL 3380D  07/26/93  1    2  FB      80  3200  PS  PROFILE.TCPIP.XA2
           APCSPL 3380D  08/03/93  1    1  VB    5124  6160  PS  STANDARD.TCPKJBIN
           APCSPL 3380D  08/03/93  1   15  F      256   256  PS  STANDARD.TCPXLBIN
           APCSPL 3380D  08/03/93  1    1  FB      80  3120  PS  TCPIP.DATA
           APCSPL 3380D  06/29/93  1    2  FB      80  3200  PS  TCPIPL62.CONFIG
           APCSPL 3380D  07/29/93  1   15  F      256   256  PS  TELNET.TCPXLBIN
           250 List completed successfully.

           Command:
  User:   site directorymode
System:
           >>>SITE directorymode
           200 Site command was accepted

           Command:
  User:   ls
System:
           >>>PORT 1,1,2,2,4,41
           200 Port request OK.
           >>>NLST
           125 List started OK.
           ETC
           FTP
           HOSTS
           NSMAIN
           PROFILE
           STANDARD
           TCPIP
           TCPIPL62
           TELNET
           250 List completed successfully.

           Command:
  User:   site datasetmode
System:
           >>>SITE datasetmode
           200 Site command was accepted

           Command:
```

```
  User:   ls
System:
           >>>PORT 1,1,2,2,4,42
           200 Port request OK.
           >>>NLST
           125 List started OK.
           ETC.RPC
           ETC.SERVICES
           FTP.DATA
           HOSTS.ADDRINFO
           HOSTS.LOCAL
           HOSTS.SITEINFO
           NSMAIN.CACHE
           NSMAIN.DATA
           PROFILE.TCPIP
           PROFILE.TCPIP.XA2
           STANDARD.TCPKJBIN
           STANDARD.TCPXLBIN
           TCPIP.DATA
           TCPIPL62.CONFIG
           TELNET.TCPXLBIN
           250 List completed successfully.
           Command:
```

Working with Directories on the Local Host

To work with directories on the local host, use the subcommands listed in Table 6.

Table 6. FTP Subcommands for Working with Directories on the Local Host

Subcommand	Description	Page
LCd	Changes the current directory on the local host	97
LMkdir	Creates a PDS on the local host	98
LPwd	Displays the name of the active working directory on the local host	112

This section shows how to choose local working directories:

```
System:    READY
  User:   ftp 1.1.2.3
System:
           MVS TCP/IP FTP V3R2
           Connecting to 1.1.2.3, port 21
           220-FTPD1 IBM MVS V3R3 at EMU.ABC.OZ, 16:15:54 on 1997-01-08
           220 Connection will close if idle for more than 5 minutes.
           USER (identify yourself to the host):
  User:   mvsuser
System:
           >>>USER mvsuser
           331 Send password please.
           Password:

           >>>PASS ********
           230 MVSUSER is logged on.  Working directory is "/u/mvsuser".
```

```
           Command:
  User:   lpwd
System:
           Local directory is MVSUSER.

           Command:
  User:   lcd tcpip
System:
           Local directory name set to MVSUSER.TCPIP.

           Command:
  User:   lpwd
System:
           Local directory is MVSUSER.TCPIP.

           Command:
  User:   lcd 'ftp.test'
System:
           Local directory name set to FTP.TEST.

           Command:
  User:   lpwd
System:
           Local directory is FTP.TEST.

           Command:
  User:   lcd ..
System:
           Local directory name set to FTP.

           Command:
  User:   lpwd
System:
           Local directory is FTP.
           Command:
```

Preparing the Environment

You can use the subcommands listed in Table 7 to prepare the environment before working with data.

Table 7 (Page 1 of 2). FTP Subcommands for Preparing the Environment

Subcommand	Description	Page
AScii	Sets the transfer type to ASCII	73
Binary	Sets the transfer type to IMAGE	75
EBcdic	Sets the transfer type to EBCDIC	86
LOCSIte	Specifies information that is used by the local host to provide service specific to that host system	100
MOde	Specifies the mode or data format of the transfer	120
QUOte	Sends an uninterpreted string of data	127

Table 7 (Page 2 of 2). FTP Subcommands for Preparing the Environment

Subcommand	Description	Page
SENDPort	Enables or disables automatic transmission of the FTP server PORT subcommand	131
SENDSite	Enables or disables automatic transmission of the SITE subcommand	132
SIte	Sends information to the foreign host using site-specific commands	132
STRuct	Sets the file transfer structure	165
SUnique	Toggles the storage methods	165
TYpe	Specifies the transfer type	168

Transferring Data

You can use the subcommands listed in Table 8 to work with and transfer data.

Table 8. FTP Subcommands for Transferring Data

Subcommand	Description	Page
APpend	Appends a data set on your local host to a file on the foreign host	72
DELEte	Deletes a single file on the foreign host	83
DELImit	Displays the delimiter character between the *file_name* and *file_type*	83
Get	Copies a file from the foreign host to your local host	90
MDelete	Deletes multiple files on the foreign host	114
MGet	Copies multiple files from the foreign host to your local host	116
MPut	Copies multiple files on your local host to the foreign host	121
PUt	Copies a file on your local host to the foreign host	124
REName	Renames a file on the foreign host	128
REStart	Restarts a checkpointed data transfer	129

How to Transfer Data

Since information could be lost or altered during transmission if you use an incorrect transfer, using the recommended methods ensures efficient data transfer. TCP/IP supports only the data transfer of a data set or file structured as a continuous sequence of data bytes. This ensures that the correct record format is preserved across MVS hosts.

Table 9 shows how to set the transmission attributes for different host systems. IBM host systems (VM or MVS) use EBCDIC for internal character representation. A text file of ASCII data type contains displayable characters; a carriage return (X'0D') and line feed (X'0A') are used to delimit a line. A text file of EBCDIC data type contains displayable characters; the newline character (X'15') is used to

delimit a line. A binary file contains a contiguous stream of bits with no line delimiters.

Table 9. Recommended Methods for Data Transfer

Transfer Between Host Types	Data Transfer Type	Data Transfer Mode
EBCDIC and EBCDIC — DBCS text data	**IBMKANJI (EBCDIC)**	**Stream**
EBCDIC and EBCDIC — text data	**EBCDIC**	**Stream**
EBCDIC and EBCDIC — DBCS binary data	**IBMKANJI (EBCDIC)**	**Block**
EBCDIC and EBCDIC — binary data	**EBCDIC**	**Block**
EBCDIC and ASCII — DBCS text data	**SJISKANJI, EUCKANJI, JIS78KJ, JIS83KJ, HANGEUL, KSC5601, TCHINESE, BIG5, SCHINESE (ASCII)**	**Stream**
ASCII and EBCDIC — text data	**ASCII**	**Stream**
ASCII and EBCDIC — DBCS binary data	**Image (binary)**	**Stream**
ASCII and EBCDIC — binary data	**Image (binary)**	**Stream**
ASCII to EBCDIC to ASCII — all data **Note:** The EBCDIC host is used for storage only. Data remains encoded in ASCII, therefore, the data cannot be used on the EBCDIC host.	**Image (binary)**	**Stream**

For more information about the DBCS data type keywords and examples, see “FTP with DBCS Support” on page 66.

For information about setting data transfer type, see TYPE subcommand. For information about setting data transfer mode, see MODE subcommand.

Examples of GET and MGET Subcommands

GET and MGET allow you to obtain files from a remote host and send them to the local host. In this example, FTP subcommands are issued from MVSXA2 to MVSVIC03

The following members exist in the data set USER121.FTP.EXAMPLE on MVSVIC03:

FILE1
FILE2
FILE3
FILE4
FILE5

Entering the FTP environment:

```
System:     READY
  User:
            ftp 9.67.113.24 621
System:
             MVS TCP/IP FTP V3R2
             Connecting to 9.67.113.24, port 621
             220-FTPD1 IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH.IBM.COM, 15:45:25
                 on 1997-03-17
             220 Connection will not timeout.
             USER (identify yourself to the host):

  User:     user121

System:      >>>USER user121
             331 Send password please.
             Password:

              >>>PASS ********
             230 USER121 is logged on.  Working directory is "/u/user121".

               Command:
  User:
            get 'user121.ftp.example(file1)' 'user121.ftp.example(file1)'
           'USER121.FTP.EXAMPLE(FILE1)' IS AN non-EXISTENT PARTITIONED DATASET.  USE
           LMKDIR TO CREATE IT. LOCAL FILE NOT FOUND
           COMMAND:
           lmkdir 'user121.ftp.example'
           USER121.FTP.EXAMPLE CREATED.
           COMMAND:
  User:
            get 'user121.ftp.example(file1)' 'user121.ftp.example(file1)'
System:      >>>PORT 9,67,112,25,4,9
             200 Port request OK.
             >>>RETR 'USER121.ftp.example(file1)'
             125 Sending data set USER121.FTP.EXAMPLE(FILE1) FIXrecfm 128
             250 Transfer completed successfully.
             3464 bytes transferred in 0.754 seconds. Transfer rate 4.59 Kbytes/sec.

             Command:

  User:
            get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)'
System:
             >>>PORT 9,67,112,25,4,34
             200 Port request OK.
             >>>RETR 'USER121.ftp.example(file2)'
             125 Sending data set USER121.FTP.EXAMPLE(FILE2) FIXrecfm 128
             250 Transfer completed successfully.
             3464 bytes transferred in 1.483 seconds. Transfer rate 2.34 Kbytes/sec.

              Command:
  User:
            get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)'

System:     Data set 'USER121.FTP.EXAMPLE(FILE2)' was not replaced.
```

```
            Local file already exists
            To replace it, use GET with the (REPLACE option
            Command:

  User:    get 'user121.ftp.example(file2)' 'user121.ftp.example(file2)' (replace

System:    >>>PORT 9,67,112,25,4,35
            200 Port request OK.
            >>>RETR 'user121.ftp.example(file2)'
            125 Sending data set USER121.FTP.EXAMPLE(FILE2)
            250 Transfer completed successfully.
            3464 bytes transferred in 0.767 seconds. Transfer rate 0.50 Kbytes/sec.
            Command:
  User:
           lpwd

System:    Local directory is USER121
            COMMAND:
  User:
           mget 'user121.ftp.example(file3)' 'user121.ftp.example(file4)'
System:
            >>>PORT 9,67,112,25,4,10
            200 Port request OK.
            >>>NLST 'user121.ftp.example(file3)'
           125 List started OK.
            250 List completed successfully.
            >>>PORT 9,67,112,25,4,11
           200 Port request OK.
            >>>NLST 'user121.ftp.example(file4)'
            125 List started OK.
            250 List completed successfully.
            >>>PORT 9,67,112,25,4,12
            200 Port request OK.
            >>>RETR 'USER121.FTP.EXAMPLE(FILE3)'
            125 Sending data set USER121.FTP.EXAMPLE(FILE3)
            250 Transfer completed successfully.
            3993 bytes transferred in 0.745 seconds.  Transfer rate 0.51 Kbytes/sec.
            >>>PORT 9,67,112,25,4,13
            200 Port request OK.
            >>>RETR 'USER121.FTP.EXAMPLE(FILE4)'
            125 Sending data set USER121.FTP.EXAMPLE(FILE4)
            250 Transfer completed successfully.
            7367 bytes transferred in 0.818 seconds. Transfer rate 9.01 Kbytes/sec.
            Command:
  User:
           lpwd
System:
            Local directory is USER121.
            Command:
  User:   cd 'user121.ftp.example'
System:
            >>>CWD 'user121.ftp.example'
            250 "USER121.FTP.EXAMPLE" partitioned data set is working directory.
            Command:
  User:
           pwd
System:
            >>>PWD
```

```
                257 "USER121.FTP.EXAMPLE" partitioned data set is working directory.
                Command:
  User:
               mget file3 file4
System:
                >>>PORT 9,67,112,25,4,20
                200 Port request OK.
                >>>NLST file3
                125 List started OK.
                250 List completed successfully.
                >>>PORT 9,67,112,25,4,21
                200 Port request OK.
                >>>NLST file4
                125 List started OK.
                250 List completed successfully.
                >>>PORT 9,67,112,25,4,22
                200 Port request OK.
                >>>RETR FILE3
                125 Sending data set USER121.FTP.EXAMPLE(FILE3)
                250 Transfer completed successfully.
                3993 bytes transferred in 0.549 seconds. Transfer rate 0.46 Kbytes/sec.
                >>>PORT 9,67,112,25,4,23
                200 Port request OK.
                >>>RETR FILE4
                125 Sending data set USER121.FTP.EXAMPLE(FILE4)
                250 Transfer completed successfully.
                7367 bytes transferred in 0.936 seconds. Transfer rate 0.23 Kbytes/sec.
                Command:
  User:
               quit
System:
               >>>QUIT
                221 Quit command received.  Goodbye.
                READY
  User:
               ftp 9.67.113.24 621
System:
                MVS TCP/IP FTP V3R2
                Connecting to 9.67.113.24, port 621
                220-FTPD1 IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH.IBM.COM, 15:45:25
                    on 1997-03-17
                220 Connection will not timeout.
                USER (identify yourself to the host):

  User:    user121

System:     >>>USER user121
                331 Send password please.
                Password:

                 >>>PASS ********
                230 USER121 is logged on.  Working directory is "/u/user121".
               Command:

  User:
               get '/u/user121/ftp.example/file1' 'user121.ftp.example(file1)'
System:
                >>>PORT 9,67,112,25,4,24
```

```
            200 Port request OK.
            >>>RETR '/u/user121/ftp.example/file1'
            125 Sending data set /u/user121/ftp.example/file1
            250 Transfer completed successfully.
            3464 bytes transferred in 1.391 seconds. Transfer rate 2.49 Kbytes/sec.

            Command:
  User:
           lcd 'user121.ftp.example'
System:
            Local directory name set to partitioned data set USER121.FTP.EXAMPLE.
            Command:
  User:
           lpwd
System:
            Local directory is partitioned data set USER121.FTP.EXAMPLE.
            Command:
  User:
           cd '/u/user121/ftp.example'
System:
            >>>CWD '/u/user121.ftp.example'
           250 HFS directory /u/user121/ftp.example is the current working directory
            Command:
  User:
           pwd
System:
            >>>PWD
           257 "/u/user121.ftp.example" is the HFS working directory.
           Command:
  User:
           get file1
System:
            >>>PORT 9,67,112,25,4,26
           200 Port request OK.
            >>>RETR file1
            125 Sending data set /u/user121/ftp.example/file1
            250 Transfer completed successfully.
            3464 bytes transferred in 1.059 seconds.  Transfer rate 3.27 kbytes/sec.
            Command:

  User:
           mget '/u/user121/ftp.example/file4' '/u/user121/ftp.example/file5'
System:
           >>>PORT 9,67,112,25,4,33
            200 Port request OK.
            >>>NLST '/u/user121/ftp.example/file4'
            125 List started OK
            250 List completed successfully.
            >>>PORT 9,67,112,25,4,34
            200 Port request OK.
            >>>NLST '/u/user121/ftp.example/file5'
            125 List started OK
            250 List completed successfully.
            >>>PORT 9,67,112,25,4,35
            200 Port request OK.
            >>>RETR /u/user121/ftp.example/file4
            125 Sending data set /u/user121/ftp.example/file4
            250 Transfer completed successfully.
```

```
7367 bytes transferred in 1.324 seconds.  Transfer rate 5.56 kbytes/sec.
200 Port request OK.
>>>RETR /u/user121/ftp.example/file5
125 Sending data set /u/user121/ftp.example/file5
250 Transfer completed successfully.
3464 bytes transferred in 0.951 seconds.  Transfer rate 3.64 kbytes/sec.
Command:
```

The data set USER121.FTP.EXAMPLE on MVSXA2 now contains the following members:

- FILE1
- FILE2
- FILE3
- FILE4
- FILE5

Note: You do not have a choice of names for the local file as a result of the MGET command.

Examples of PUT and MPUT Subcommands

PUT and MPUT allow you to send files from a local host to a remote host. In this example, FTP subcommands are issued from MVSXA2 to MVSVIC03. The data set USER121.FTP.EXAMPLE on MVSXA2 contains the following members:

- APPEND01
- XA2FILE1
- XA2FILE2
- XA2FILE3

The data set USER121.FTP.EXAMPLE on MVSVIC03 contains the following members:

- XA3FILE1
- XA3FILE2
- XA3FILE3

Entering the FTP environment:

```
System:     READY
  User:
           ftp 1.1.2.3
System:
           MVS TCP/IP FTP V3R2
            Connecting to 1.1.2.3, port 21
            220-FTPD1 IBM MVS V3R3 at MVSVIC03, 15:46:59 on 1997-03-17
            220 Connection will close if idle for more than 5 minutes.
            USER (identify yourself to the host):
  User:
           user121
System:
           >>>USER user121
            331 Send password please.
            Password:
             >>>PASS ********
            230 user121 is logged on.  Working directory is "/u/user121"
```

```
            Command:

  User:    put 'user121.ftp.example(xa2file1)' 'user121.ftp.example(f1from2)'

System:      >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
             200 Site command was accepted
             >>>PORT 1,1,2,2,4,48
             200 Port request OK.
             >>>STOR 'user121.ftp.example(f1from2)'
             125 Storing data set USER121.FTP.EXAMPLE(F1FROM2)
             250 Transfer completed successfully.
             390 bytes transferred in 1.117 seconds.
             Transfer rate 0.35 Kbytes/sec.
             Command:
  User:
            put 'user121.ftp.example(xa2file1)' 'user121.ftp.example(f1from2)'
System:
             >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
             200 Site command was accepted
             >>>PORT 1,1,2,2,4,49
             200 Port request OK.
             >>>STOR 'user121.ftp.example(f1from2)'
             125 Storing data set USER121.FTP.EXAMPLE(F1FROM2)
             250 Transfer completed successfully.
             390 bytes transferred in 0.680 seconds.
             Transfer rate 0.57 Kbytes/sec.
             Command:
  User:
            sunique
System:
             Store unique is ON
             Command:
  User:
            put 'user121.ftp.example(xa2file1)' 'user121.ftp.example(f1from2)'
System:
             >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
             200 Site command was accepted
             >>>PORT 1,1,2,2,4,50
             200 Port request OK.
             >>>STOU 'user121.ftp.example(f1from2)'
             125 Storing data set USER121.FTP.EXAMPLE(F1FROM21) ( unique name )
             250 Transfer completed successfully.
             390 bytes transferred in 1.085 seconds.
             Transfer rate 0.36 Kbytes/sec.

             Command:
  User:
            sunique
System:
             Store unique is OFF
             Command:
  User:
            cd 'user121.ftp.example.'
System:
            >>>CWD 'user121.ftp.example.'
             257 "'USER121.FTP.EXAMPLE.'" is working directory name prefix.
             Command:
```

```
  User:
            lpwd
System:
            Local directory is USER121.
             Command:
  User:
            lcd 'user121.ftp.example'
System:
            Local directory name set to PDS USER121.FTP.EXAMPLE.
             Command:
  User:
            lpwd
System:
            Local directory is partitioned data set USER121.FTP.EXAMPLE.
             Command:
  User:
            mput xa2file2 xa2file3
System:
             >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
             200 Site command was accepted
             >>>PORT 1,1,2,2,4,51
             200 Port request OK.
             >>>STOR XA2FILE2
             125 Storing data set USER121.FTP.EXAMPLE.XA2FILE2
             250 Transfer completed successfully.
             390 bytes transferred in 1.437 seconds.
             Transfer rate 0.27 Kbytes/sec.
             >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
             200 Site command was accepted
             >>>PORT 1,1,2,2,4,52
             200 Port request OK.
             >>>STOR XA2FILE3
             125 Storing data set USER121.FTP.EXAMPLE.XA2FILE3
             250 Transfer completed successfully.
             390 bytes transferred in 1.091 seconds.
             Transfer rate 0.36 Kbytes/sec.
             Command:
  User:
            quit
System:
            >>>QUIT
             221 Quit command received. Goodbye.
             READY
```

The data set USER121.FTP.EXAMPLE on MVSVIC03 now contains the following members:

- F1FROM2
- F1FROM21
- XA3FILE1
- XA3FILE2
- XA3FILE3

MVSVIC03 now also has the following data sets:

- USER121.FTP.EXAMPLE.XA2FILE2
- USER121.FTP.EXAMPLE.XA2FILE3

Changing Local Site Defaults Using FTP.DATA

The default values for the local site parameters are hard-coded in the FTP client module. You can change these default values by creating a FTP.DATA configuration data set.

The FTP.DATA configuration data set is optional. FTP client uses the following search order to obtain the local site parameter values:

1. *userid*.FTP.DATA
2. *hlq*.FTP.DATA

If neither data set exists, the FTP client uses hard-coded defaults.

Table 10 describes the statements you can code in the FTP.DATA data set. It is not necessary to include all statements in the FTP.DATA data set, only those statements whose default values are to be changed. The hard-coded default is used for any statement not included in the FTP.DATA data set.

You can change several of the FTP local site parameters during the FTP session by using the LOCSITE subcommand. See “LOCSITE Subcommand—Specify Site Information to the Local Host” on page 100 for more information about using the LOCSITE subcommand to change the local site parameters.

Table 10 (Page 1 of 5). FTP.DATA Data Set Statements for the FTP client

Parameter	Value	Description
AUTOMOUNT	true	Permits automatic mounting of volumes for data sets on volumes that are not mounted. This is the hard-coded default.
	false	Prevents automatic mounting of volumes for data sets on volumes that are not mounted.
AUTORECALL	true	Permits automatic recall of migrated data sets. This is the hard-coded default.
	false	Prevents automatic recall of migrated data sets.
BLKSIZE	*size*	Specifies the block size of a newly allocated data set. The hard–coded default is 6233. Valid range is 0 to 32760. BLKSIZE is functionally equivalent to BLOCKSIZE. The BLOCKSIZE parameter is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS.
BLOCKSIZE	*size*	Specifies the block size of a newly allocated data set. The hard-coded default is 6233. Valid range is 0 to 32760. BLOCKSIZE is functionally equivalent to BLKSIZE. The BLOCKSIZE parameter is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS.

Table 10 (Page 2 of 5). FTP.DATA Data Set Statements for the FTP client

Parameter	Value	Description
CCTRANS	*dsname*	Specifies the SBCS translation table to be used for the control connection. FTP uses the translation table in the *user_id.dsname.*TCPXLBIN data set. If the *user_id.dsname.*TCPXLBIN data set does not exist, FTP uses the *hlq.dsname.*TCPXLBIN data set. FTP resolves the translation table to be used for the control connection by: 1. data set specified in FTP subcommand with TRANSLATE option 2. data set specified in CCTRANS parameter in FTP.DATA data set 3. default search order a. *user_id*.FTP.TCPXLBIN b. *hlq*.FTP.TCPXLBIN c. *user_id*.STANDARD.TCPXLBIN d. *hlq*.STANDARD.TCPXLBIN 4. FTP's internal tables
CHKPTInt	*interval*	Specifies the checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted. The actual frequency of marker blocks is a function of the checkpoint interval value and the size of the buffers being used. If the checkpoint interval is 0, no checkpointing occurs and no marker blocks are transmitted. The default value is 0. **Note:** Use this parameter only if you are sending to a server that supports checkpointing.
CCONNTIME	*seconds*	Defines the amount of time to wait after attempting to close a control connection before terminating it and reporting an error. The default is 30.
DATACLASS	*Class*	Specifies the data class for SMS created files as defined by your organization for the FTP client. To cancel the specification, use LOCSITE with the DATAC=* parameter. See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.
DATACTTIME	*seconds*	Defines the amount of time to wait after attempting to send or receive data before terminating the connection and reporting an error to the user. The default is 120.
DB2	*name*	Specifies the name of the DB2 subsystem. The default name is DB2.
DCBDSN	*name*	Specifies the name of the data set to be used as a model for allocation of new data sets. There is no hard-coded default.
DCONNTIME	*seconds*	Defines the amount of time to wait after attempting to close a data transfer before terminating the connection and reporting an error. The default is 120.
DIRECTORY	*size*	Specifies the number of directory blocks to be allocated for the directory of a PDS. The hard-coded default is 27.

Table 10 (Page 3 of 5). FTP.DATA Data Set Statements for the FTP client

Parameter	Value	Description
DIRECTORYMODE	true	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.
	false	Specifies that all the data set qualifiers below the current directory are treated as entries in the directory. This is the hard-coded default.
FILETYPE	*type*	Specifies the file type of the data set. The hard-coded default is SEQ. The description of each file type is: **Type** **Description** SEQ Sequential or partitioned data sets SQL SQL query function JES Remote job submission
INACTTIME	*seconds*	Specifies the amount of time to wait for an expected response from the server, on either the control or the data connection, before closing the session. Data transfer times that exceed this value will not cause session termination unless the time between data packet arrivals exceeds this value. The hard-coded default is 300.
LRECL	*length*	Specifies the record length of a newly allocated data set. Valid values are 1 through 32756. The default is 256. **Note:** You can also use a value of x to indicate the maximum possible value.
MGMTCLASS	*class*	Specifies the SMS management class to be assigned to newly allocated data sets. To cancel the specification, use LOCSITE subcommand with the MG=* parameter. See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.
MIGRATEVOL	*volid*	Indicates the volume ID (volid) for migrated data sets using non-IBM storage management systems. If you do not specify MIGRATEVOL, the default volid is MIGRAT.
MYOPENTIME	*seconds*	Defines the amount of time to wait when waiting for a session to open before terminating the attempt and reporting an error. The default is 60.
NCP	*number*	Specifies the number of disk I/O buffers to be used during data transfer. Valid values are 2 through 20. The hard-coded default is 3.
PRIMARY	*value*	Specifies the amount of direct access storage for primary allocation of new data sets. The hard-coded default is 1.

Table 10 (Page 4 of 5). FTP.DATA Data Set Statements for the FTP client

Parameter	Value	Description
RDW	true	Specifies that RDWs are treated as if they were part of the record and not discarded during FTP transmission of VB or VBS data sets in other than block mode. **Note:** RDW information is stored in a binary halfword. You should transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.
	false	Specifies that RDWs are discarded during FTP transmission of VB or VBS data sets in other than block mode. This is the hard-coded default.
RECFM	*format*	Specifies the record format of a data set. Valid record formats are: F, FA, FB, FBA, FBM, FBS FM, U, V, VA, VB, VBA, VBM, VBS, VM, and VS. The default is VB. The characters used to specify these record formats have the following meanings: **Format** — **Description** F — Fixed record length V — Variable record length U — Undefined record length B — Blocked records S — Spanned records A — Records contain ISO/ANSI control characters M — Records contain machine code control characters
RETPD	*days*	Specifies the number of days that a newly allocated data set should be retained. The default is 0, which means no retention period. If you do not specify a retention period, no retention period is assigned to newly allocated data sets. The maximum is 9999.
SBTRANS	*dsname*	Specifies the SBCS translation table to be used for the data connection. This table will be used for SBCS and DBCS data transfers. FTP uses the translation table in the *user_id.dsname.*TCPXLBIN data set. If the *user_id.dsname.*TCPXLBIN data set does not exist, FTP uses the *hlq.dsname.*TCPXLBIN data set. FTP resolves the translation table to be used for the data connection by: 1. data set specified in FTP subcommand with TRANSLATE option 2. data set specified in SBTRANS parameter in FTP.DATA data set 3. default search order a. *user_id*.FTP.TCPXLBIN b. *hlq*.FTP.TCPXLBIN c. *user_id*.STANDARD.TCPXLBIN d. *hlq*.STANDARD.TCPXLBIN 4. same tables that are to be used for the control connection.
SECONDARY	*value*	Specifies the amount of direct storage for secondary allocation of new data sets. The hard-coded default is 1.

Table 10 (Page 5 of 5). FTP.DATA Data Set Statements for the FTP client

Parameter	Value	Description
SPACETYPE	*type*	Specifies whether newly allocated data sets are allocated in blocks, cylinders, or tracks. Valid values for *type* are BLOCK, CYLINDER, or TRACK. The hard-coded default is TRACK.
SPRead	true	Specifies that the output is in spreadsheet format when the file type is SQL.
	false	Specifies that the output is in report format rather than spreadsheet format when the file type is SQL. This is the hard-coded default.
SQLCOL	*type*	Specifies what the column headings of the SQL output file are. The following list describes each type: **Type** — **Description** Names — Database column names are used for the column headings. The labels are ignored. This is the hard-coded default. Labels — Database column names are used for the column headings. If any of the columns do not have labels, the corresponding column heading in the output file is left blank. Any — The database column heading is the first choice for column heading. If there is no label, the database column name becomes the column heading.
STORCLASS	*Class*	Specifies the storage class for the SMS-created files as defined by your organization for the FTP client. To cancel the specification, use LOCSITE with the STO=* parameter. See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.
UNITNAME	*type*	Specifies the unit *type* (for example, 3380) for allocation of new data sets. You can use the UNITNAME parameter to specify either direct access or tape device types. If UNITNAME is not specified, the unit type used for allocation is the system default. For tape output, only standard label (SL) and file sequence number 1 is supported. For input, multiple file volumes are supported if the data set is cataloged.
VOLUME	*name*	Specifies the volume serial number for allocation of new data sets. The value specified for *name* is case sensitive. If VOLUME is not specified, the volume serial number used for allocation is the system default.
WRAPRECORD	true	Indicates that data is wrapped to the next record if there is no new line character.
	false	Indicates that data is truncated if there is no new line character. This is the default.

Sample FTP.DATA Data Set (FTCDATA)

The following is a sample of the contents of the FTP.DATA data set in the FTCDATA member of the *hlq*.SEZAINST data set.

```
;*********************************************************************
;                                                                    *
;   Name of File:            tcpip.SEZAINST(FTCDATA)                 *
;                                                                    *
;   Descriptive Name:         FTP.DATA  (for FTP Client)             *
;                                                                    *
;   SMP/E Distribution Name:  EZAFTCDA                               *
;                                                                    *
;   COPYRIGHT = NONE.                                                *
;                                                                    *
;   This FTP.DATA file is used to specify default file and disk      *
;   parameters used by the FTP client.                               *
;                                                                    *
;   Note: For an example of an FTP.DATA file for the FTP server,     *
;   see the FTSDATA example.                                         *
;                                                                    *
;   Syntax Rules for the FTP.DATA Configuration File:                *
;                                                                    *
;   (a) All characters to the right of and including a ; will be     *
;       treated as a comment.                                        *
;                                                                    *
;   (b) Blanks and <end-of-line> are used to delimit tokens.         *
;                                                                    *
;   (c) The format for each statement is:                            *
;                                                                    *
;       parameter value                                              *
;                                                                    *
;   (d) abbreviations are not permitted in the FTP.DATA file for     *
;       the FTP client.                                              *
;                                                                    *
;*********************************************************************
;
;
; File and disk parameters
;
AUTOMOUNT       TRUE       ; Nonmounted volumes mounted automatically
AUTORECALL      TRUE       ; Migrated HSM files recalled automatically
BLOCKSIZE       6144       ; Block size is 6144 bytes
;CCTRANS         CTRL       ; mlq for control connection translate table
;DCBDSN          MODEL.DCB  ; Data set name used as model for allocation
DIRECTORY       15         ; PDS allocated with 15 directory blocks
DIRECTORYMODE   FALSE      ; Use all qualifiers (Datasetmode)
FILETYPE        SEQ        ; File Type = SEQ (default)
LRECL           128        ; Logical record length is 128 bytes
;MGMTCLASS       TCPMGMT    ; SMS management class for new data sets
NCP             15         ; 15 I/O buffers
PRIMARY         5          ; Primary allocation is 5 tracks
RDW             FALSE      ; Do not retain RDWs as data
RECFM           FB         ; Fixed blocked record format
RETPD           30         ; New data set expiration date is 30 days
;RETPD           0          ; No retention period
SECONDARY       2          ; Secondary allocation is 2 tracks
;SBTRANS         DATA       ; mlq for data connection translate table
```

```
SPACETYPE       TRACK      ; Data sets allocated in tracks
;UNITNAME       SYSDA      ; Unit name used for allocation
;VOLUME          WRKLB2    ; Volume serial number for allocation
```

Using Different SBCS Translation Tables for the Control and Data Connections

If the SBCS table you need for SBCS or DBCS data transfer does not support the standard encodings for the portable character set, you should establish different translation tables for the control and data connections. This will avoid unrecognized ASCII conversion of FTP commands.

You can specify distinct tables by using the

`CCTRANS`

statement for the control connection and the

`SBTRANS`

statement for the data connection in the *userid*.FTP.DATA or the *hlq*.FTP.DATA data set. For example, you could create the following data sets:

- *userid*.FTP.DATA containing:

  ```
  CCTRANS ccdsname
  SBTRANS sbdsname
  ```

- *userid.ccdsname*.TCPXLBIN containing the SBCS binary tables for the control connection
- *userid.sbdsname*.TCPXLBIN containing the SBCS binary tables for the data transfers

Specifying Values for New Data Sets

When allocating new data sets, there are two methods you can use to specify the data set attributes. You can individually use the storage attribute parameters with the SITE and LOCSITE commands or the *hlq*.FTP.DATA data set. Or, if your system programmer has used the Storage Management System to group together default attributes into named classes, you can specify those class names on the DATACLASS, STORCLASS, and MGMTCLASS parameters.

Dynamic Allocation

FTP allows you to dynamically allocate a new physical sequential data set or a partitioned data set (PDS) for the purpose of transferring data to be written to that data set. The following optional allocation variables can be used by the client to override and turn off the hard-coded defaults that affect the allocation of the data set.

Variable	**FTP.DATA parameter**
allocation units	SPACETYPE
blocksize	BLOCKSIZE
data class	DATACLASS
directory blocks	DIRECTORY
logical record length	LRECL
management class	MGMTCLASS
model DCB values	DCBDSN

primary space	PRIMARY
record format	RECFM
retention period	RETPD
secondary space	SECONDARY
storage class	STORCLASS
unit	UNITNAME
volume serial number	VOLUME

Some of these allocation variables might provide duplicate information. For example, the model DCB may have a record format (RECFM) that differs from the record format specified by a data class and from the one explicitly specified by the client. FTP passes all variables that are specified to dynamic allocation and lets it determine which of the specifications take precedence. The following list describes the exceptions to that policy:

- If neither the primary nor secondary space quantity is specified, then the allocation units value is not sent.
- If the data set organization is physical sequential, then directory blocks specification is not sent.
- Otherwise, all variables are sent to dynamic allocation where the order of precedence is:
  1. Any FTP.DATA, SITE, or LOCSITE parameters explicitly specified or defaulted
  2. Any attributes picked up from the model DCB and not otherwise explicitly specified
  3. Any attributes picked up from the data class and not previously derived from 1 or 2
  4. Any allocation defaults

Storage Management Subsystem (SMS)

An FTP client can specify one or more of the following SMS classes to manage characteristics that are associated with or assigned to data sets.

- Data class is an SMS construct that determines data set allocation attributes used by SMS for creation of data sets. The fields listed are available attributes that serve as a template for allocation. Each is **optional** and is overridden by any explicit specification of FTP allocation variables or by a model DCB (DCBDSN).

Variable	**FTP.DATA parameter**
directory blocks	DIRECTORY
logical record length	LRECL
primary space	PRIMARY
record format	RECFM
retention period	RETPD
secondary space	SECONDARY

  Note: If either primary or secondary space is explicitly specified, then the primary and secondary values from data class are not used.

- Management class is an SMS construct that determines DFHSM action for data set retention, migration, backup, and release of allocated but unused space. Management class replaces and expands attributes that otherwise would be

specified. That is, management class may override any other specification of retention period.

- Storage class is a list of storage performance and availability services requests for an SMS-managed data set that SMS attempts to honor when selecting a volume or volumes for the data set. It may conflict with a explicit specification of volume and unit. If storage class is used, then volume and unit parameters should be unspecified.

Using a DCBDSN Model to Create a New Data Set

To use a DCBDSN model to create a data set, first issue the following command:

```
SITE DCBDSN=data_set_name
```

where *data_set_name* is the name of the data set to be used as a model to set the values of the logical record length (LRecl), the blocksize (BLKSIze), and the record format (RECfm) of a new data set.

Then issue the following command to set the LRecl, BLKSIze, and RECfm values to 0.

```
SITE LRECL BLKSIZE RECFM
```

Then issue the following command to create the new data set with the values specified by the DCBDSN model:

```
PUT data_set_name
```

where *data_set_name* is the name of the new data set.

Note: If your client does not support the SITE command, you may enter the command using the QUOTE command. For example:

```
QUOTE SITE DCBDSN=data_set_name
```

Generation Data Group Support

Generation data groups (GDGs) allow you to store multiple data sets, called generation data sets (GDS) as versions of the GDG. You cannot use FTP to create a new GDG, but you can use it to create a new version (that is, a new GDS) or to transfer an existing version of an existing GDG.

Before you specify a (+nnn) value to create a new GDS, issue the following command:

```
SITE DCBDSN=model
```

This subcommand specifies an MVS data set to be used as a model. The model must have a DSORG of PS. The other DCB characteristics of the data set are not checked.

Notes:

1. Failure to have a valid DCBDSN before trying to create a new GDS may cause FTP or ALLOCATION to fail or to return unpredictable results.
2. If you issue a SITE DCBDSN LRECL BLKSIZE command before the creation of a new data set, the LRECL and BLKSIZE parameters on the SITE command will overide the LRECL and BLKSIZE parameters on the DCBDSN command.

For more information about GDGs, see *MVS/DFP Version 3 Release 3: Using Data Sets*.

The following are sample FTP commands that access a GDG called JIMKEO.GDG.

Notes:

1. In the following examples, gdg (0), gdg (-1), and gdg (+1) specify which copy of the GDG you are using. 0 indicates the latest version, -1 indicates the previous version, and +1 indicates that a new version is created.
2. GDGALL is not supported by the C-FTP server. GDGALL processing occurs when the base name for the GDG is specified without a relative index value.

The following example illustrates a PUT to the latest existing GDS. The working directory at the server is JIMKEO.

```
Command:
put my.gdg gdg(0)
>>>SITE FIXrecfm 150
200-Blocksize must be a multiple of lrecl for FB  data sets. Blocksize set to
6150.
200 Site command was accepted
>>>PORT 129,34,128,245,126,229
200 Port request OK.
>>>STOR gdg(0)
125 Storing data set JIMKEO.GDG.G0055V00
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.24 Kbytes/sec.
```

The following example illustrates a GET of the latest GDG:

```
Command:
get gdg(0) my.gdg2
>>>PORT 129,34,128,245,126,233
200 Port request OK.
>>>RETR gdg(0)
125 Sending data set JIMKEO.GDG.G0055V00 FIXrecfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.04 Kbytes/sec.
```

The following example illustrates a PUT to a new GDS (After the STOR is complete, this new version is referenced by (0)).

```
Command:
put my.gdg gdg(+1)
>>>SITE FIXrecfm 150
200 Site command was accepted
>>>PORT 129,34,128,245,126,234
200 Port request OK.
>>>STOR gdg(+1)
125 Storing data set JIMKEO.GDG.G0056V00
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 1.16 Kbytes/sec.
```

The following example illustrates a GET of the previous GDS into the local file called my.gdg3:

```
Command:
get gdg(-1) my.gdg3
>>>PORT 129,34,128,245,126,239
200 Port request OK.
>>>RETR gdg(-1)
125 Sending data set JIMKEO.GDG.G0055V00 FIXrecfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 2.77 Kbytes/sec.
```

The following example illustrates a GET that replaces the contents of my.gdg3 with the most recent GDS:

```
Command:
get gdg(0) my.gdg3 (replace
>>>PORT 129,34,128,245,126,243
200 Port request OK.
>>>RETR gdg(0)
125 Sending data set JIMKEO.GDG.G0056V00 FIXrecfm 150
250 Transfer completed successfully.
612 bytes transferred. Transfer rate 3.36 Kbytes/sec.
```

The following example illustrates changing the working directory:

```
Command:
cd gdg
>>>CWD gdg
257 "'JIMKEO.GDG.'" is working directory name prefix.
```

The following example shows the files created:

```
Command:
dir
>>>MODE s
200 Data transfer mode is Stream.
>>>PORT 129,34,128,245,127,12
200 Port request OK.
>>>LIST
125 List started OK.
Volume Unit    Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
STRG73 3380K   04/30/92   1    5 FB      150 32700  PS   G0003V00
STRG65 3380K   04/30/92   1    5 FB      150 32700  PS   G0006V00
STRG61 3380K   04/30/92   1    5 FB      150 32700  PS   G0010V00
STRG47 3380K   04/30/92   1    5 FB      150 32700  PS   G0015V00
STRG47 3380K   04/30/92   1    5 FB      150 32700  PS   G0021V00
STRG66 3380K   04/30/92   1    5 FB      150 32700  PS   G0028V00
STRG47 3380K   04/30/92   1    5 FB      150 32700  PS   G0036V00
STRG01 3380K   04/30/92   1    5 FB      150 32700  PS   G0045V00
STRG53 3380K   04/30/92   1    5 FB      150 32700  PS   G0055V00
STRG59 3380K   04/30/92   1    5 FB      150 32700  PS   G0056V00
250 List completed successfully.
>>>MODE b
200 Data transfer mode is Block.
Command:
```

Submitting FTP Requests in Batch

FTP is usually run interactively by starting and entering commands from your terminal. You can also run FTP as a batch job, but you must supply the job control language (JCL) file. You can use batch when you know what functions you want to perform, when you want a hard copy of the results, or when you want to perform an FTP function many times.

Notes:

1. The file containing the FTP commands cannot have sequence numbers in it. It must be saved *un*numbered. Input streams containing sequence numbers cause unpredictable results because the FTP client parses the job stream until the end of record (EOR).
2. A plus sign (+) is used as a continuation character at the end of an FTP subcommand line to indicate the line following is a continuation of the subcommand.
3. Do not attempt to use server parameters and options on this JCL, because it is acting as a client.
4. To have the FTP client perform DB/2 queries in a batch job, the DSNLOAD library must be in the link list or appear on a STEPLIB DD statement for the job.
5. Use the (EXIT parameter if you want FTP to display an error return code and then exit when certain errors are detected. See “FTP EXIT Return Codes” on page 48 for more information.

   Attention

   Because MVS batch condition codes range from 0 to 4095 (X'FFF'), the step condition code displayed by the JCL will not match the exit return code that FTP displays in its message. What is displayed is the remainder of a division of the FTP exit return code by 4096. When viewed in hex, this is the same as the 3 last hex digits of the exit return code.

   For example, a PUT that fails and returns the message: "550 Open failed for ...", would get an exit return code of 27550. If you divide 27770 by 4096, the remainder of 2794 would be the JCL STEP condition code.

   It is easier to visualize this in hex, where decimal 27550 becomes hex 6B9E; the last 3 hex digits, B9E, is the decimal number 2794.

Figure 1 on page 45 shows an example of the JCL required to submit a batch job.

```
//USER28F  JOB ,CARTER,MSGLEVEL=(1,1)
//FTPSTP1  EXEC PGM=FTP,REGION=2048K,
//              PARM='9.67.112.25 21 (timeout 20'
//STEPLIB  DD  DSN=TCP3.SEZALINK,DISP=SHR
//SYSTCPD  DD  DSN=ANYHLQ.TCPIP.DATA,DISP=SHR
//SYSFTPD  DD  DSN=ANYHLQ.FTP.DATA,DISP=SHR
//NETRC    DD  DSN=ANYHLQ.NETRC,DISP=SHR
//SYSMDUMP DD  SYSOUT=A
//SYSPRINT DD  SYSOUT=H
//OUTPUT   DD  SYSOUT=H
//INPUT    DD  *
type e
mode b
put idss.parts
/*
```

Figure 1. JCL to Run FTP in Batch

Note: In Figure 1, REGION=2048K is a minimum requirement. The requirement could increase depending on the block size of the data set being transmitted.

In Figure 1, the first JCL statement is a standard job statement. The next JCL statement is an EXEC statement that specifies `PGM=FTP`, and a region parameter, because FTP might use more storage than your default region size.

Note: For `PARM=`, you can specify any parameter that is valid when invoking FTP from your terminal. See “FTP Command—Enter the FTP Environment” on page 87 for more information. These parameters are only supported on the `PARM=` field of the EXEC card.

To run FTP in batch, you must have the following 3 DD statements:

SYSPRINT DD	Specifies the data set where you want messages to be returned.
INPUT DD	Specifies the data set where the FTP subcommands to be performed are located.
OUTPUT DD	Specifies the data set where FTP is to place the output of the FTP functions performed.

Note: The data set specified on the OUTPUT DD statement should have an LRECL of 160 with any block size that is a multiple of the LRECL. The data set specified on the INPUT DD statement should have an LRECL of 80 with any block size that is a multiple of the LRECL.

You can use the *user_id*.NETRC data set, as defined by the NETRC DD statement in Figure 1, to identify the userid and password for a batch-processed remote login. You can also specify the userid and password in the INPUT DD data set.

Figure 2 on page 46 shows the records in an INPUT DD data set that contains the FTP commands to be executed.

```
HOSTNAME
USERID PASSWD
DIR
PUT MYFILE.LISTING
QUIT
```

Figure 2. Contents of an INPUT DD Data Set

The first line of Figure 2 contains the name of the host that you want FTP to use. The second line contains the user ID to which you want to connect, followed by its password. The next 3 lines contain the FTP commands that you want FTP to perform. In this example, FTP is doing a directory listing of the server to which you are connecting. The example then instructs FTP to send a file to the server. The last line ends the connection.

The results of the session and any commands you execute appear in the OUTPUT DD data set. The SYSPRINT can contain some additional messages that relate to the execution of your FTP session.

Note: If you do not want your password to be copied to the output file, specify your user ID and password on separate input lines. See Figure 3 for an example.

Submitting Requests without Input and Output Data Sets

Figure 3 shows an easier way to submit a batch job, because you can avoid referring to data sets for input and output.

```
//USERIDX  JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID,MSGCLASS=H,TIME=9
//FTP   EXEC PGM=FTP,REGION=4096K
//INPUT DD *
nodeid
userid
password
DIR
GET hostfile.name locfile.name
QUIT
//OUTPUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Figure 3. Another Job to Run FTP in Batch

Figure 4 on page 47 shows step1 creating a new GDS in batch and FTP getting the data set.

```
//USERIDX  JOB USERID,MSGLEVEL=(1,1),NOTIFY=USERID,MSGCLASS=H,TIME=9
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=USER31.SOURCE.DATA,DISP=SHR     (MYDGD.G0008V00)
//SYSUT2   DD DSN=MYDGD(+1),DISP=(,CATLG),
             UNIT=SYSDA, SPACE=(TRK,(1,1)),DCB=(MODEL)
//FTP   EXEC PGM=FTP,REGION=4096K
//INPUT DD *
nodeid
userid
password
DIR
GET MYDGD(0) A.DATA.SET
QUIT
//OUTPUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Figure 4. Job to Create a new GDS in Batch

Note: All the GDG allocation in batch must be complete before the start of FTP.

Using the EXEC Interface

The FTP EXEC interface allows you to execute FTP commands from an EXEC rather than interactively from a terminal. The FTP subcommands to be performed can be in a data set, or you can code them directly in the EXEC.

By default, the results of a user FTP dialog are printed on the terminal. If you want the dialog results sent to a data set rather than the terminal, specify an OUTPUT data set as part of the ALLOC statement.

Issuing FTP Subcommands from a Data Set

Figure 5 is an example of an EXEC that issues FTP subcommands from a data set. In this example, the FTPIN1 data set is used for the FTP subcommands, and FTPOUT1 is used to store the FTP session dialog. The EXEC is written in REXX.

```
/*REXX*/
"ALLOC DA(FTPIN1) DD(INPUT) SHR REU"   /* Input will be from FTPIN1 */
if rc ¬= 0 then do
   say 'Error in ALLOC INPUT, rc = ' rc
   exit
end
"ALLOC DA(FTPOUT1) DD(OUTPUT) SHR REU" /* Output goes to FTPOUT1     */
if rc ¬= 0 then do
   say 'Error in ALLOC OUTPUT, rc = ' rc
   exit
end
"FTP YKTVSH"                           /* FTP to the YKTVSH host       */
"FREE DD(INPUT)"
"FREE DD(OUTPUT)"
EXIT
```

Figure 5. How to Issue the FTP Subcommands from a Data Set

The following is an example of the INPUT data set (in Figure 5, the INPUT data set name is FTPIN1).

```
krasik mvsftp
cd examples
put t.info t1.info
get t1.info t2.info (r
quit
```

Where:

`krasik`	Is the user ID.
`mvsftp`	Is the password.
`t.info`	Is the file to be transferred.

Issuing FTP Subcommands Directly from the EXEC Interface

Figure 6 is an example of how to issue FTP subcommands directly from a REXX EXEC.

```
/*REXX*/
Address TSO
"LISTDS FTP.TEMP.FILE"
if rc = 0 then  "DELETE FTP.TEMP.FILE"
"ALLOC DA(FTP.TEMP.FILE) DD(INPUT) NEW BLKSIZE(800) REUSE LRECL(80) ,
       CATALOG RECFM(F B) DSORG(PS) "
if rc ¬= 0 then do
  say "error in ALLOC, rc = " rc
  exit
end
"NEWSTACK"
QUEUE 'krasik mvsftp'
QUEUE 'cd examples'
QUEUE 'put t.info t1.info'
QUEUE 'get t1.info 'user14.t2.info' (r'
QUEUE 'cd ..'
QUEUE 'cd dummy'
QUEUE 'quit'
QUEUE '
"EXECIO * DISKW INPUT (FINIS"
"FTP YKTVSH (EXIT "            /* FTP to the YKTVSH host       */
"DELSTACK"
"FREE DD(INPUT)"
"DELETE FTP.TEMP.FILE"
EXIT
```

Figure 6. How FTP Subcommands Can Be Issued from an EXEC

Note: If data set DUMMY does not exist, FTP exits with a return code.

FTP EXIT Return Codes

FTP EXIT return codes are displayed (in message EZA1735I) when the (EXIT parameter is used on the FTP command and an error is detected. The return codes are composed of a subcommand code and a reply code. FTP EXIT return codes have the following format:

yyxxx

Where:

yy Represents the subcommand code, which is a number from 1 to 99. Each subcommand has an EXIT_IF_ERROR flag that determines whether FTP is exited when an error occurs if you specified EXIT on the FTP command. See Table 11 on page 49 for a description of the possible FTP subcommand codes.

xxx Represents the reply code that is sent from the server. The reply code is a 3-digit number. See Table 12 on page 51 for a description of the possible reply codes.

For example, the FTP EXIT return code 16550 indicates the following:

16 The GET command failed.

550 The reply code from the FTP server. The requested action was not taken; the file was not found or could not be accessed.

The FTP EXIT return code 4532 indicates the following:

4 The APPEND command failed.

532 The reply code from the FTP server. Need an account for storing files.

FTP Subcommand Codes

Table 11 lists the valid FTP subcommand codes. The “EXIT_IF_ERROR” column specifies whether an error causes FTP to end if you specified the EXIT parameter on the FTP command.

Table 11 (Page 1 of 2). FTP Subcommand Codes

Code Number	Subcommand	EXIT_IF_ERROR
1	AMBIGUOUS	true
2	?	false
3	ACCOUNT	true
4	APPEND	true
5	ASCII	true
6	BINARY	true
7	CD	true
8	CLOSE	true
9	TSO	true
10	OPEN	true
11	DEBUG	false
12	DELIMIT	false
13	DELETE	true
14	DIR	true
15	EBCDIC	true
16	GET	true
17	HELP	false
18	LOCSTAT	true
19	USER	true
20	LS	true

Table 11 (Page 2 of 2). FTP Subcommand Codes

Code Number	Subcommand	EXIT_IF_ERROR
21	MDELETE	true
22	MGET	true
23	MODE	true
24	MPUT	true
25	NOOP	true
26	PASS	true
27	PUT	true
28	PWD	true
29	QUIT	true
30	QUOTE	true
31	RENAME	true
32	SENDPORT	true
33	SENDSITE	false
34	SITE	false
35	STATUS	true
36	STRUCT	true
37	SUNIQUE	true
38	SYSTEM	true
39	TRACE	false
40	TYPE	true
41	LCD	true
42	LOCSITE	true
43	LPWD	true
44	MKDIR	true
45	LMKDIR	true
46	EUCKANJI	true
47	IBMKANJI	true
48	JIS78KJ	true
49	JIS83KJ	true
50	SJISKANJI	true
51	CDUP	true
52	RMDIR	true
53	HANGEUL	true
54	KSC5601	true
55	TCHINESE	true
56	RESTART	false
99	UNKNOWN	true

FTP Reply Codes

When you enter an FTP command, TCP/IP displays the sequence of subcommands, if any, that are sent to the foreign host's FTP server. In addition, the response from the FTP server is also displayed as a reply code. These replies ensure the synchronization of requests and actions during data transfer, and guarantee that the client always know the state of the foreign host's FTP server.

Table 12 on page 51 lists the descriptions of the possible reply codes.

Note: The description of the reply codes is not the exact message that is displayed on your screen; the description describes the content of the reply code.

Table 12 (Page 1 of 2). FTP Reply Codes

Code	Description
000	FTP subcommand contains an incorrect parameter **Note:** A reply code of 000 is returned from the FTP client when it detects an incorrect parameter. The FTP client in this case does not send the command to the FTP server.
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
208	Unable to delete data set because expiration date has not passed
211	System status, or system help reply
212	Directory status
213	File status
214	Help message
215	MVS is the operating system of this server
220	Service ready for new user
221	QUIT command received
226	Closing data connection; requested file action successful
230	User logged on; proceed
250	Requested file action okay, completed
257	PATH NAME created
331	Send password please
332	Supply minidisk password using account
421	Service not available
425	Cannot open data connection
426	Connection closed; transfer ended abnormally
450	Requested file action not taken; file busy
451	Requested action abended; local error in processing
452	Requested action not taken; insufficient storage space in system

Table 12 (Page 2 of 2). FTP Reply Codes

Code	Description
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 abended; page type unknown
552	Requested file action ended abnormally; exceeded storage allocation
553	Requested action not taken; file name not allowed
554	Transfer aborted; unsupported SQL statement

FTP Internal Error Codes

If an internal error occurs when you enter an FTP subcommand, the reply code is an internal error code rather than an FTP reply code. Table 13 lists the valid internal error codes.

Table 13. Internal Error Codes

Code	Error	EXIT_IF_ERROR
01	NOmoreSLOTS	false
02	TOOfewDOTS	false
03	WOULDclobberFILE	false
04	CMSfileERROR	false
05	CMSfileNOTfound	false
06	CMSdiskFILEid	false
07	CMSinvalidCHAR	false
08	CMSdiskNOTaccessed	false
09	CMSdiskREADonly	false
10	CMSfileNOTaccessed	false
11	BLOCKbutNOTebcdcic	false
12	INITemulationERROR	true
13	OPENwasNOTissued	true
14	NOinputFILE	false
15	CANTwriteTOoutput	false
16	USERwasNOTissued	true

For example, the internal error code 13 indicates the following:

13 The internal error `OPENwasNOTissued` occurred.

Interfacing with JES

MVS's Job Entry System (JES) allows you to perform the following functions:

- Submit jobs, consisting of JCL and data, to the Job Scheduler for execution
- Spool JCL messages and SYSOUT during execution
- Print the output
- View the output
- Delete job output

FTP server provides the following functions in its JES interface:

- Submitting a job
- Displaying the status of all the user's jobs
- Receiving the spool output of the job (JCL messages and SYSOUT)
- Deleting a job
- Submitting a job and automatically receiving output
- Terminating access to JES

Submitting a Job

A job consists of job control language (JCL) and data. To submit a job using FTP, you must do the following:

1. Create the JCL and data that you want to submit, using the editor on your client. The job name in the JCL must be USERID*x*, where *x* is a 1-character letter or number and USERID must be the user ID you will use to log into the FTP server to submit the job. For example: `MYUSRIDA`. The output class for MSGCLASS and SYSOUT files contained in your JCL must specify a JES HOLD output class.

   Note: The maximum LRECL for the submitted job is 254 characters. JES scans only the first 72 characters of JCL.

2. Start a session with the FTP server on the MVS system to which you want to submit the job.
3. After you have entered your user ID and password, specify that you want to interface to JES with a site parameter by entering the following:

   ```
   SITE FILETYPE=JES
   ```

4. To submit the JCL file you have created, enter the following:

   ```
   PUT filename.filetype
   ```

5. The JCL is then submitted to the JES internal reader and waits for an initiator to start the job. The job is submitted under the user ID that you used when you logged on to the system.
6. The default for *filetype* is SEQ, and when you want to go back to normal FTP file transfer mode, enter the following:

   ```
   SITE FILETYPE=SEQ
   ```

See *JCL Reference Manual* for more information about using JCL.

Displaying the Status of a Job

After you have submitted your job, you can determine whether it is waiting for execution, running, or finished. You do this with the FTP subcommands DIR or LS while in the FILETYPE=JES mode. These subcommands display the status of all the jobs that are on the JES spool for your user ID. A display for the DIR subcommand might look like the following:

```
MYUSRIDA JOB05444 OUTPUT 3 spool Files
MYUSRIDB JOB05766 OUTPUT 6 spool Files
MYUSRIDC JOB05832 OUTPUT 6 spool Files
MYUSRIDD JOB05946 ACTIVE
MYUSRIDE JOB06021 INPUT                 —HELD—
```

The first column displays the job name. The second column displays the job ID, assigned by JES. This 8-character job ID, consisting of JOB followed by a 5-digit number assigned by JES, is the way that JES identifies your job. The third column displays the status of the job.

The following is a description of each status:

Status	Description
INPUT	The job was received, but not run yet.
HELD	The JCL specified that the job is to be put on hold.
ACTIVE	The job is running.
OUTPUT	The job has finished and has output to be printed or retrieved. For each OUTPUT job, there are spool files that consist of JCL messages, JES messages, initiator and terminator messages, and SYSOUT. For jobs with a status of OUTPUT, you are told the number of spool files for each job.

Note: The LS subcommand gives the same results as the DIR subcommand, but does not provide the number of spool files. Providing this spool information consumes a lot of computer resources. The server provides the job id only as the result of an LS subcommand in order to support an MGET subcommand. Use the LS subcommand rather than the DIR subcommand when possible.

Receiving Spool Output

You can retrieve spool files individually or in a group.

Receiving Individual Spool Files

Retrieving the spool files one at a time enables you to see whether a job ran correctly before you retrieve the rest of the output, or to retrieve a dump but not the rest of the output.

To retrieve the spool output, use the FTP GET subcommand while in the FILETYPE=JES mode, specifying the job ID and the number of the spool file you want.

You can specify a short form of the job ID by entering the letter `J` followed by a 4- or 5-digit job number. For example:

```
GET JOB05444.1 JOB05444.FILE1 (REPLACE
GET JOB05766.6 ASSEMBLY.FILE6
GET JOB06235.2 (REPLACE
GET JOB00275.4
GET J7438.3
```

In these examples, *foreign_file* is specified first, followed by *local_file* (on your client machine) with the appropriate options, such as REPLACE. The first example requests that the first spool file for JOB05444 be transmitted and replace the file on your client named JOB05444.FILE1. The second command requests that the sixth spool file for JOB05766 be transmitted to your client with the name ASSEMBLY.FILE6.

If you have specified FILETYPE=JES, you can use the MGET subcommand to receive output from multiple jobs without specifying them one at a time. For example you can enter:

`MGET` *`parameter`*

The FTP client requires a parameter of the MGET subcommand. The parameter is passed to the FTP server but is not used. The server returns all of the SYSOUT files for all of the jobs in the HELD queue for your user ID.

Notes:

1. On an MVS FTP server, *local_file* must be specified.
2. Truncation can cause a loss of data.
3. A GET command performed on an empty data set erases the contents of the existing local data set.
4. Receiving the output of a job does not remove the job output from the queue. To remove the job output from the queue, you must issue a DELETE command.

Receiving a Group of Spool Files

To retrieve all the spool files associated with the same job simultaneously into the same destination file, specify:

`GET` *`jobid`*`.x`

where *x* can be either upper or lowercase. All the spool files are transferred together and put into file `jobid.x`. The following line appears between each retrieved JES spool file:

```
!! END OF JES SPOOL FILE !!
```

This allows you to easily find the end of each spool file.

You can also specify a data set name to send the files to, such as:

`GET` *`jobid`*`.x` *`data_set_name`*

All the spool files are put into the file named `file.name`. This eliminates the need to retrieve each spool file separately. For example:

```
GET J3456.X
```

retrieves all the spool files for JOB03456 and puts them in a file named J3456.X.

`MGET` *`parameter`*

The command MGET with any parameter produces the same results as issuing GET *jobid.x* commands for each job that is associated with your user ID.

Notes:

1. In JES2, the spool files retrieved by GET and tallied by DIR must be in a hold queue (commonly class=H).
2. In JES3, the spool files must be in a hold queue reserved for external writers. Ask your system programmer for the class that says (HOLD=EXTWTR) in the JES3 installation stream.
3. The maximum record length that can be received at the server is 254 characters before the record is truncated.
4. Receiving the output of a job does not remove the job output from the queue. To remove the job output from the queue, you must issue a DELETE command.

Deleting a Job

You can delete a job before or during execution, or you can delete the output of a job before you have retrieved it. You do this by using the DELETE subcommand while in the FILETYPE=JES mode, along with the job ID. You can specify either the 8-character job ID or a short form of the job ID by entering the letter J followed by a 4- or 5-digit job number. For example:

```
DELETE JOB05444
DELETE J3672
```

When you issue the DELETE command, all spool output related to a job is deleted.

The host returns the message `CANCEL SUCCESSFUL` after it deletes the job.

Submitting a Job and Automatically Receiving Output

You can submit a job using FTP and automatically receive your output. Rather than using the JCL you built on the FTP client, this function uses the JCL you have built on the FTP server. To submit a job using FTP, perform the following steps:

1. Create the JCL and data that you want to submit and save it on the MVS host where the FTP server resides. The JCL can reside in a sequential or partitioned data set. The job name in the JCL must be USERID*x*, were *x* is a 1-character letter or number. The output class for MSGCLASS and SYSOUT files contained in your JCL must specify a JES HOLD output class.
2. Start a session with the FTP server on the MVS system to which you want to submit the job.
3. After you have entered your user ID and password, specify that you want to interface to JES with a site parameter by entering the following:

   ```
   SITE FILETYPE=JES
   ```

4. To submit the JCL file you have created, enter the following command:

   ```
   GET jclfilename.jclfiletype outputfilename.outputfiletype
   ```

   The *outputfilename.outputfiletype* defines the data set at the FTP client site that is to contain the HELD job output when the job completes.

The MVS FTP server reads the data set *jclfilename.jclfiletype* and submits it to the JES internal reader. It then sends the client the following 2 messages:

```
125 Submitting job outputfilename.outputfiletype FIXrecfm 80
125 When JOB05125 is done, will retrieve its output
```

Note: When submitting a job and automatically receiving the output, remember that your session is suspended. You should use care, based on the anticipated run time of your job, when using this function. If your session times out, you must restart FTP and manually retrieve your output. Normally, there is a 10 minute time out. You can change the time out based on what you specify on the JESPUTGETTO statement in the FTP.DATA data set.

Terminating Access to JES

The default for FILETYPE is SEQ. When you want to end access to JES and return to FTP in its normal file transfer mode, specify the following:

```
SITE FILETYPE=SEQ
```

JES Examples

The following example shows the JCL file USER121.JCL.CNTL(SMFALL) being submitted to the JES. Before FTP commands are issued, only the data set USER121.FTP.EXAMPLE exists on MVSXA2.

```
System:    READY
  User:   ftp 9.67.113.24 621
System:   MVS TCP/IP FTP V3R2
          FTP.DATA FILE NOT FOUND.  USING HARDCODED DEFAULT VALUES.
           Connecting to 9.67.113.24, port 621
           220-FTPSERVE IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH.IBM.COM, 19:03:08
            on 1997-03-17
           220 Connection will close if idle for more than 5 minutes.
           USER (identify yourself to the host):
  User:   user121
System:   >>>USER user121
            331 Send password please.
            Password:
            >>>PASS ********
            230 user121 is logged on. Working directory is "/u/user121".

            Command:
  User:   site file=jes
System:   >>>SITE file=jes
            200 Site command was accepted

            Command:
  User:
           put 'user121.jcl.cntl(mvsjob)'
System:
           >>>SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=27920
           200 Site command was accepted
           >>>PORT 9,67,112,25,4,37
           200 Port request OK.
           >>>STOR 'user121.jcl.cntl(mvsjob)'
           125 Sending Job to JES Internal Reader FIXrecfm 80
           250-It is known to JES as JOB02189.
           250 Transfer completed successfully.
```

```
           1066 bytes transferred in 3.118 seconds. Transfer rate 0.34 Kbytes/sec.
           Command:
  User:    dir
System:    >>>PORT 9,67,112,25,4,38
           200 Port request OK.
           >>>LIST
           125 List started OK.
           USER121A  JOB00067  INPUT
           250 List completed successfully.
           Command:
  User:    dir
System:    >>>PORT 9,67,112,25,4,39
           200 Port request OK.
           >>>LIST
           125 List started OK.
           USER121A  JOB00067  ACTIVE
           250 List completed successfully.
           Command:
  User:    dir
System:    >>>PORT 9,67,112,25,4,40
           200 Port request OK.
           >>>LIST
           125 List started OK.
           USER121A  JOB00067  OUTPUT    4 Spool Files
           250 List completed successfully.
           Command:
  User:    lcd 'user121.ftp.example.'
System:    Local directory name set to USER121.FTP.EXAMPLE.
           Command:
  User:    lpwd
System:    Local directory is USER121.FTP.EXAMPLE.

           Command:
  User:    dir
System:    >>>PORT 9,67,112,25,4,41
           200 Port request OK.
           >>>LIST
           125 List started OK.
           USER121A  JOB00067  OUTPUT    4 Spool Files
           250 List completed successfully.

           Command:
  User:    get job00067.x spoolall
System:    >>>PORT 9,67,112,25,4,42
           200 Port request OK.
           >>>RETR job00067.x
           125 Sending all SPOOL files for requested JOBID.
           250 Transfer completed successfully.
           5935 bytes transferred in 4.755 seconds. Transfer rate 1.25 Kbytes/sec.
           Command:
  User:    get job00067.1 spool1
System:    >>>PORT 9,67,112,25,4,43
           200 Port request OK.
           >>>RETR job00067.1
           125 Sending data set USER121.USER121A.JOB00067.D000002.JESMSGLG
           250 Transfer completed successfully.
           1962 bytes transferred in 0.739 seconds. Transfer rate 2.65 Kbytes/sec.
           Command:
```

```
  User:
           get job00067.2 spool2
System:
           >>>PORT 9,67,112,25,4,44
           200 Port request OK.
           >>>RETR job00067.3
           125 Sending data set USER121.USER121A.JOB00067.D000003.JESYSMSG
           250 Transfer completed successfully.
           1982 bytes transferred in 2.123 seconds. Transfer rate 0.93 Kbytes/sec.
           Command:
  User:   get job00067.3 spool3
System:   >>>PORT 9,67,112,25,45
           200 Port request OK.
           >>>RETR job00067.3
           125 Sending data set USER121.USER121A.JOB00067.D000004.JESYSMSG
           250 Transfer completed successfully.
           1982 bytes transferred in 2.123 seconds. Transfer rate 0.93 Kbytes/sec.
           Command:
  User:   get job00067.4 spool4
System:   >>>PORT 9,67,112,25,46
           200 Port request OK.
           >>>RETR job00067.4
           125 Sending data set USER121.USER121A.JOB00067.D000103.?
           250 Transfer completed successfully.
           1227 bytes transferred in 0.380 seconds. Transfer rate 3.23 Kbytes/sec.
           Command:
  User:
           get job00067.5 spool5
System:   >>>PORT 9,67,112,25,47
           200 Port request OK.
           >>>RETR job00067.5
           550 Index 5 is greater than number of spool files for JOB00067
           Command:

  User:   dir
System:   >>>PORT 9,67,112,25,4,50
           200 Port request OK.
           >>>LIST
           125 List started OK.
           user121A  JOB00067  OUTPUT    4 Spool Files
           250 List completed successfully.

           Command:
  User:   delete job00067
System:   >>>DELE job00067
          250 Cancel Successful

           Command:
  User:   dir
System:   >>>PORT 9,67,112,25,4,51
           200 Port request OK.
           >>>LIST
           125 List started OK.
           No jobs found on Held queue
           250 List completed successfully.
           Command:
  User:   site filetype=seq
System:   >>>SITE filetype=seq
```

```
           200 Site command was accepted

           Command:
  User:    quit
System:    >>>QUIT

           221 Quit command received. Goodbye.
           READY
```

After executing the FTP commands, the following data sets now exist on MVSXA2:

USER121.FTP.EXAMPLE.SPOOLALL
USER121.FTP.EXAMPLE.SPOOL1
USER121.FTP.EXAMPLE.SPOOL2
USER121.FTP.EXAMPLE.SPOOL3
USER121.FTP.EXAMPLE.SPOOL4

Note: In most situations, the INPUT status is too fast to be captured by issuing `DIR`. However, if the ACTIVE or OUTPUT status of the job is captured, the INPUT status has been passed successfully.

Performing DB2 SQL Queries with FTP

FTP allows you to submit a Structured Query Language (SQL) SELECT query to the DB2 subsystem and receive the results of the SQL query. FTP can perform this function as either the server or the client.

For information on installing the SQL query function for the FTP client or server, refer to *TCP/IP for MVS: Customization and Administration Guide*.

SQL Data Types Supported by FTP

FTP access to SQL supports the following data types by the client and the server:

- DATE
- TIME
- TIMESTAMP
- VARCHAR (variable length, up to 254 characters)
- CHAR (fixed length, up to 254 characters)
- DECIMAL
- INTEGER (full word)
- SMALLINT (half word)

In addition, the FTP server supports the following data types:

- FLOAT (single or double precision)
- LONG VARCHAR (**VARCHAR**(*n*), where *n* is greater than 254)
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC

For the server, mixed data (double-byte character set and single-byte character set) is supported in CHAR, VARCHAR, and LONG VARCHAR data types, but column alignment in the output file might not be maintained.

Creating the Input Data Set

Before performing a DB2 SQL query using FTP, you must create an MVS data set that contains the SQL query you want to perform.

You can create queries on the client and use the FTP PUT command to send the queries to the MVS system to be processed. Or, you can prepare a group of "stock" SQL queries on the MVS system and perform them regularly.

Note: FTP can pass only one SQL query per file.

For example, a data set on an MVS system named `userid.SQL.IN` contains the following SQL query:

```
SELECT LASTNAME, EMPID, YEARS_EMPLOYED FROM EMPLOYEE_TABLE
WHERE YEARS_EMPLOYED > 25
```

You either created that data set on the MVS system with TSO, or you used the FTP PUT command to put the data set on the MVS system.

Setting the Characteristics for the SQL Query

After creating a data set to use for your query, you must log on to FTP and set the file type for the query:

```
SITE/LOCSITE FILETYPE=SQL
```

There are several commands that are relevant to the client and server in SQL mode. The server commands use SITE, and the client commands use LOCSITE. The following list describes the commands:

SITE/LOCSITE DB2=
: Specifies the name of the DB2 subsystem that you want to perform your queries. See "Specifying the DB2 Subsystem to Perform the Query" on page 62 for more information about DB2 subsystems.

SITE/LOCSITE SPRead or NOSPRead
: Specifies whether you want the output to be in spreadsheet or report format. See "Specifying the Output Format" on page 63 for more information about output format.

SITE/LOCSITE SQLCOL=
: Specifies Names, Labels, or Any signifying whether you want the column headings to use the DB2 column names or labels. See "Assigning Column Headings for the SQL Query Result Table" on page 62 for more information about column headings.

To return to normal FTP processing, specify:

```
SITE/LOCSITE FILETYPE=SEQ
```

On MVS systems, RECFM=VB is a recommended format that allows you to view the results of the SQL query. Issue the following command to specify that new data sets should be created with the RECFM=VB attribute:

```
SITE/LOCSITE RECFM=VB
```

To prevent the automatic sending of a SITE command that might override your SITE setting, toggle SENDSITE to OFF. For more information about the SENDSITE command, see "SENDSITE Subcommand—Toggle the Sending of Site Information" on page 132.

Specifying the DB2 Subsystem to Perform the Query

An MVS system can run several DB2 systems simultaneously, each known by a subsystem name of up to 4 characters. For example, you can have a DB2 test system called DB2T and a DB2 production system called DB2P.

FTP connects to a DB2 system to have it execute a DB2 query. You can specify what DB2 system FTP should connect to with the following SITE or LOCSITE parameter:

```
SITE/LOCSITE DB2=
```

For example, if you want the FTP server to have the DB2T system perform your queries, specify:

```
SITE DB2=DB2T
```

If you want the FTP client to have the DB2P system perform your queries, specify:

```
LOCSITE DB2=DB2P
```

The default DB2 system name is *DB2*. You can change the default with the DB2 parameter in the FTP.DATA data set. See “Changing Local Site Defaults Using FTP.DATA” on page 33 for more information about the FTP.DATA data set.

Assigning Column Headings for the SQL Query Result Table

The creator of a DB2 table can assign descriptive labels to the table’s columns. For example, a column name could be XCM554, but the label could be WEEKLY PAY. For information about assigning names and labels, see the DB2 DESCRIBE statement in the *DB2 SQL Reference*.

The SQLCOL parameter of the SITE command allows you to specify whether you want names or labels to appear at the top of the columns in your output file. The default is Names. Issue the following command if you want a database column name to appear at the top of each column in your output file:

```
SITE/LOCSITE SQLCol=Names
```

Issue the following command if you want a label to appear at the top of each column:

```
SITE/LOCSITE SQLCol=Labels
```

If you specify the Labels parameter and a column in your query does not have a label defined in the data base, the FTP server supplies a column heading. For more information about column headings, see “FTP-Supplied Column Headings” on page 63.

Issue the following command if you want either a label or a name to appear at the top of each column:

```
SITE/LOCSITE SQLCol=Any
```

If you specify the Any parameter, the label appears as the column heading. However, if the column does not have a label, the name appears at the top of the column.

FTP-Supplied Column Headings

The FTP server provides column headings in the result table when DB2 does not. This occurs when a result table contains expression columns or when labels are requested and a data base column that appears in the result table does not have a label defined.

The server builds a column heading for expression columns. For example,

```
Select employee, salary/52 from ABC.Staff
```

results in 2 columns. The first column gets its name from DB2, while the second column is built by the server. The server will use the heading `COL002` for the second column because it supports the SQL limit of 750 columns.

Specifying the Output Format

You have 2 choices for the format of your output data set: spreadsheet format and report format. The default is NOSPRead (report format), but you can change the default for your FTP server by changing the FTP.DATA data set. See “Changing Local Site Defaults Using FTP.DATA” on page 33 for more information.

Spreadsheet Format: You can have the output of the SQL query formatted to load directly into a spreadsheet program running on a PC or a workstation. To get the spreadsheet format, issue the following command:

```
SITE SPRead or LOCSITE SPRead
```

The SPRead format option puts a TAB character in front of the first character of each column entry, except the first column. See your spreadsheet program documentation for instructions about how to import the output of the SQL query.

Report Format: The NOSPRead format option puts one or more blank spaces between the columns, and it lists the SQL query, the column headings, and the resulting columns. Each section is separated with horizontal dashed lines. An output data set in NOSPRead format is easier to view and print.

To get the report format, issue the following command:

```
SITE NOSPRead or LOCSITE NOSPRead
```

The following is an example of the results contained in the NOSPRead format of the SQL.OUTPUT data set.

```
s---------+---------+---------+---------+---------+--------+---------+
SELECT * FROM DB2USER.PHONES
         WHERE FIRSTNAME LIKE 'BILL%'
         OR    FIRSTNAME LIKE 'WILL%'
h---------+---------+---------+---------+---------+--------+---------+
LASTNAME              FIRSTNAME       TIE  EXT  ALT  DEPT ROOM   NODE
d---------+---------+---------+---------+---------+--------+---------+
ACKERMAN              BILL            893  6266 7813 431  J2-A22 IBMABC
ADAMS                 WILLIAM J.      892  2202 1716 681  33-943 IBMABC
ASTERMAN              WILLIAM C.      893  7244 7813 222  J4-A44 IBMVM2
BENDER                WILLIAM R.      892  4217 4766 490  45-556 IBMVM2
```

A lowercase letter in the first position of each dashed line specifies what part of the output follows, allowing a program to read and interpret the contents. For example, `s` indicates that the SQL query follows, `h` indicates a header, `d` indicates that the

rest of the data set is the actual data, and `e` indicates that an error message follows.

The width of the output data set depends on the width of the results from the DB2 query.

Submitting the Query

After you have created a data set that contains an SQL query, logged on to FTP, and set the appropriate SITE or LOCSITE parameters, you are ready to execute the contents of the data set. You can do this from either an FTP client or an FTP server.

Performing an SQL Query from an FTP Client

To have the FTP client perform SQL queries and have the results sent to an FTP server, specify:

```
LOCSITE FILETYPE=SQL
```

Then perform a PUT command specifying the name of the file on the client that contains the SQL query.

For example, if the client has a file named `userid.SQL.IN` that contains an SQL query, you can specify:

```
PUT SQL.IN SQL.OUT
```

The FTP client then submits the query found in SQL.IN to the DB2 subsystem on the client and sends the resulting rows of output to the server to be put into SQL.OUT on the server.

To return to normal FTP processing, specify:

```
LOCSITE FILETYPE=SEQ
```

Performing an SQL Query from an FTP Server

To have the FTP server perform the query and have the results sent to the client, specify:

```
SITE FILETYPE=SQL
```

Then perform a GET command specifying the name of the file on the server that contains the SQL query.

For example, if the server has a file named *userid*.SQL.IN that contains an SQL query, you can specify:

```
GET SQL.IN SQL.OUT
```

The FTP server then submits the query found in SQL.IN to the DB2 subsystem on the server and sends the resulting rows of output to the client to be put into SQL.OUT on the client.

Examples of SQL Query Output

This section shows examples of SQL query output using different options.

With NOSPRead and SQLCol=Names

The following output is from a query using NOSPRead and SQLCol=Names.

```
s--------+---------+---------+---------+---------+---------+---------+---------+
SELECT EMPLOYEE,AGE
FROM   ABC.STAFF
WHERE  AGE < 60
h--------+---------+---------+---------+---------+---------+---------+---------+
EMPLOYEE                  AGE
d--------+---------+---------+---------+---------+---------+---------+---------+
Steve Jasinski             23
Alison Cook                22
```

With SPRead and SQLCol=Names

The following output is from a query with SPREAD and SQLCol=Names.

Note: The period symbol (.) represents a TAB character.

```
EMPLOYEE            .AGE
Steve Jasinski      . 23
Alison Cook         . 22
```

The following output examples are for the query:

```
SELECT DISTINCT ABC.STAFF.TLA, ABC.STAFF.SALARY
FROM   ABC.STAFF, ABC.HOURS
WHERE  (ABC.STAFF.TLA = ABC.HOURS.TLA) AND
       (ABC.HOURS.TOTAL > 40)
```

With NOSPRead and SQLCol=Names

The following is output is from a query using NOSPRead and SQLCol=Names.

```
s--------+---------+---------+---------+---------+---------+---------+---------+
SELECT DISTINCT ABC.STAFF.TLA, ABC.STAFF.SALARY
FROM   ABC.STAFF, ABC.HOURS
WHERE  (ABC.STAFF.TLA = ABC.HOURS.TLA) AND
       (ABC.HOURS.TOTAL > 40)
h--------+---------+---------+---------+---------+---------+---------+---------+
TLA    SALARY
d--------+---------+---------+---------+---------+---------+---------+---------+
ACO  20050.00
SJJ  19040.00
```

With NOSPRead and SQLCol=Labels

The following output is from a query using NOSPRead and SQLCol=Labels.

```
s--------+---------+---------+---------+---------+---------+---------+---------+
SELECT DISTINCT ABC.STAFF.TLA, ABC.STAFF.SALARY
FROM   ABC.STAFF, ABC.HOURS
WHERE  (ABC.STAFF.TLA = ABC.HOURS.TLA) AND
       (ABC.HOURS.TOTAL > 40)
h--------+---------+---------+---------+---------+---------+---------+---------+
EMPLOYEE'S INITIALS    SALARY
d--------+---------+---------+---------+---------+---------+---------+---------+
ACO                  20050.00
SJJ                  19040.00
```

With NOSPRead and SQLCol=Names

The following output is from a query using NOSPRead and SQLCol=Names.

```
s--------+---------+---------+---------+---------+---------+---------+---------+
SELECT * FROM ABC.STAFF
h--------+---------+---------+---------+---------+---------+---------+---------+
EMPLOYEE              TLA      AGE    SALARY
d--------+---------+---------+---------+---------+---------+---------+---------+
Steve Jasinski        SJJ       23  28040.00
Alison Cook           ACO       22  28040.00
Mark Ballam           MFB       63  87420.55
```

With NOSPRead and SQLCol=Any

The following output is from a query using NOSPRead and SQLCol=Any.

```
s--------+---------+---------+---------+---------+---------+---------+---------+
SELECT * FROM ABC.STAFF
h--------+---------+---------+---------+---------+---------+---------+---------+
EMPLOYEE              EMPLOYEE'S INITIALS      AGE    SALARY
d--------+---------+---------+---------+---------+---------+---------+---------+
Steve Jasinski        SJJ                       23  28040.00
Alison Cook           ACO                       22  28040.00
Mark Ballam           MFB                       63  87420.55
```

FTP with DBCS Support

This section describes how to use FTP to exchange DBCS data sets between hosts supporting DBCS file transfer.

The MVS TCP/IP FTP server and client programs access data sets containing data that is usually in EBCDIC format. To transfer these data sets to or from an ASCII based host requires translation tables. The transfer of DBCS data uses two tables—one for DBCS characters and one for SBCS characters.

Selecting a DBCS Translation Table

The LOADDBCSTABLES statement in *hlq*.TCPIP.DATA is used by both the FTP server and client to determine which DBCS translation table data sets can be loaded . See the *TCP/IP for MVS: Customization and Administration Guide* for more information about the loading and customizing of DBCS translation tables for FTP.

The FTP server and client can be configured to load a number of DBCS translation tables. These are used during data set transfers to convert MVS host DBCS characters and non-MVS DBCS characters. The command

```
TYPE B n
```

or corresponding client subcommand is used to enter DBCS transfer mode and select a DBCS table. For information on selecting a SBCS table, see “Selecting an SBCS Translation Table” on page 67 below.

Selecting an SBCS Translation Table

The SBCS table used to transfer DBCS data is the SBCS table that is established for the data connection .

SBCS tables are used by the control connection to transfer commands; they are also used by the data connection. Often the same SBCS table is used, but you may want to select a different table to be used for data transfers. How you specify the SBCS table for the data connection depends on whether the translation is to be done by the FTP server or the FTP client.

When the EBCDIC-to-ASCII translation will be done by the FTP server, you can issue

```
SITE SBDATACONN
```

to select the SBCS table to be used by the server for data transfers.

When the EBCIDIC-to-ASCII translation will be done by the FTP client, you can use the following parameters in your local FTP.DATA file to establish the SBCS tables:

CCTRANS dsname
Establishes the SBCS tables the client uses for control connections.

SBTRANS dsname
Establishes the SBCS tables the client uses for data connections.

Alternatively, you can use the TRANSLATE option of the FTP command to change the SBCS and DBCS translation table hierarchy for both the control and data connection. The TRANSLATE option will result in the same SBCS table for both the control and the data connection.

Note: The TRANSLATE option can be used as long as the table maintains the integrity of the portable character set.

DBCS Subcommands

DBCS data sets are transferred using the standard FTP subcommands PUT and GET. However, before the transfer commences, the current transfer type for the session must be set to the required DBCS type. To set the transfer type to DBCS for an FTP session, you must issue the appropriate FTP subcommand to the client or the server, depending on where the DBCS conversion is to be done. The FTP subcommands for DBCS support are listed in Table 14.

Table 14 (Page 1 of 2). FTP Subcommands for DBCS Support

Subcommand	Description	Page
BIG5	Sets the transfer type to BIG-5	74
EUckanji	Sets the transfer type to EUCKANJI	86
HAngeul	Sets the transfer type to HANGEUL	91
Ibmkanji	Sets the transfer type to IBMKANJI	93
JIS78kj	Sets the transfer type to JIS78KJ	94
JIS83kj	Sets the transfer type to JIS83KJ	95
Ksc5601	Sets the transfer type to KSC5601	96

Table 14 (Page 2 of 2). FTP Subcommands for DBCS Support

Subcom-mand	Description	Page
QUOte	Sends an uninterpreted string of data	127
SChinese	Sets the transfer type to SCHINESE	130
SJiskanji	Sets the transfer type to SJISKANJI	146
TChinese	Sets the transfer type to TCHINESE	166
TYpe	Specifies the transfer type	168

Server Commands and Client Subcommands

Table 15 shows examples of the server command that would be generated for each client subcommand alias:

Table 15 (Page 1 of 2). FTP TYPE Subcommand Aliases

Client Subcommand	Server Command	Description
BIG5	TYPE B 8	Big-5 transfer type
EUCKANJI	TYPE B 2	Extended Unix Code Kanji transfer type
HANGEUL	TYPE B 5	Hangeul transfer type
IBMKANJI	TYPE F 1	IBM (EBCDIC) Kanji transfer type
JIS78KJ	TYPE B 4 A	JIS 1978 Kanji using ASCII shift-in transfer type
JIS78KJ (ASCII	TYPE B 4 A	ASCII shift-in escape sequence
JIS78KJ (JISROMAN	TYPE B 4 R	JISROMAN shift-in escape sequence
JIS78KJ (JISROMAN NOSO	TYPE B 4 R N	Pure DBCS data transfer
JIS83KJ	TYPE B 3 A	JIS 1983 Kanji using ASCII shift-in transfer type
JIS83KJ (ASCII	TYPE B 3 A	ASCII shift-in escape sequence
JIS83KJ (JISROMAN	TYPE B 3 R	JISROMAN shift-in escape sequence
JIS83KJ (JISROMAN NOSO	TYPE B 3 R N	Pure DBCS data transfer
KSC5601	TYPE B 6	Korean Standard Code KSC-5601 transfer type
SCHINESE	TYPE B 9	Simplified Chinese transfer type
SJISKANJI	TYPE B 1	Shift JIS Kanji transfer type
SJISKANJI (SOSI	TYPE B 1 S A	Shift-out/shift-in characters X'1E'/X'1F'
SJISKANJI (SOSI ASCII	TYPE B 1 S A	Shift-out/shift-in characters X'1E'/X'1F'
SJISKANJI (SOSI EBCDIC	TYPE B 1 S E	Shift-out/shift-in characters X'0E'/X'0F'

Table 15 (Page 2 of 2). FTP TYPE Subcommand Aliases

Client Subcommand	Server Command	Description
SJISKANJI (SOSI SPACE	TYPE B 1 S S	Shift-out/shift-in characters X'20'/X'20'
SJISKANJI (NOSO	TYPE B 1 N	Pure DBCS data transfer
TCHINESE	TYPE B 7	Traditional Chinese (5550) transfer type

Mapping DBCS Aliases to CCSIDs

The code sets supported by the DBCS for FTP options conform to standard coded character set identifiers (CCSIDs). Table 16 shows how CCSIDs map to DBCS keywords.

For more information about CCSIDs, see *Character Data Representation Architecture*, (SC09-1390).

Table 16. Mapping of DBCS Keywords to CCSIDs

DBCS Keyword	CCSID	Description
BIG5	00947	IBM Big-5 DBCS
EUCKANJI	00954	Japanese EUC (G0, G1 and G2 only)
HANGEUL	00926	Korean DBCS-PC
JIS78KJ	00955	JIS X0208–1978
JIS83KJ	05048	JIS X0208–1990
KSC5601	00951	IBM Korean Standard code
SCHINESE	01380	Simplified Chinese DBCS-PC
SJISKANJI	00301	Japanese DBCS-PC
TCHINESE	00927	Traditional Chinese DBCS-PC

FTP Subcommands

The FTP subcommands are listed in Table 17. The minimum abbreviation, a description, and the page reference for each subcommand are also included. You must be in the FTP environment to use the FTP subcommands. See “FTP Command—Enter the FTP Environment” on page 87 for more information.

Table 17 (Page 1 of 3). FTP Subcommands

Subcommand	Description	Page
?	Provides information to use FTP	92
ACcount	Sends host-dependent account information	71
APpend	Appends a data set on your local host to a file on the foreign host	72
AScii	Sets the transfer type to ASCII	73
BIG5	Sets the transfer type to BIG5	74

Table 17 (Page 2 of 3). FTP Subcommands

Subcommand	Description	Page
BINary	Sets the transfer type to IMAGE	75
CD	Changes the working directory	76
CDUP	Changes to the parent of the current working directory	78
CLose	Disconnects from the foreign host	80
CWd	Changes the working directory (Synonymous with CD)	76
DEBug	Toggles internal debug options	81
DELEte	Deletes a single file on the foreign host	83
DELImit	Displays the delimiter character between the *file_name* and *file_type*	83
DIr	Lists the directory entries for files on the foreign host	83
EBcdic	Sets the transfer type to EBCDIC	86
EUckanji	Sets the transfer type to EUCKANJI	86
Get	Copies a file from the foreign host to your local host	90
HAngeul	Sets the transfer type to HANGEUL	91
Help	Displays help information for FTP	92
Ibmkanji	Sets the transfer type to IBMKANJI	93
JIS78kj	Sets the transfer type to JIS78KJ	94
JIS83kj	Sets the transfer type to JIS83KJ	95
Ksc5601	Sets the transfer type to KSC5601	96
LCd	Changes the current directory on the local host	97
LMkdir	Creates a PDS on the local host	98
LOCSIte	Specifies information that is used by the local host to provide service specific to that host system	100
LOCSTat	Displays FTP status information for the local host	110
LPwd	Displays the name of the active working directory on the local host	112
LS	Lists the names of files on the foreign host	113
MDelete	Deletes multiple files on the foreign host	114
MGet	Copies multiple files from the foreign host to your local host	116
MKdir	Creates a directory on the foreign host	118
MOde	Specifies the mode or data format of the transfer	120
MPut	Copies multiple files on your local host to the foreign host	121
NOop	Checks whether the foreign host is still responding	122
Open	Opens a connection to a foreign host	123
PAss	Supplies a password to the foreign host	123
PUt	Copies a file on your local host to the foreign host	124
PWd	Displays the name of the active working directory on the foreign host	125

Table 17 (Page 3 of 3). FTP Subcommands

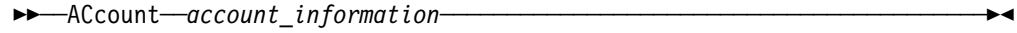
Subcommand	Description	Page
QUIt	Leaves the FTP command environment	126
QUOte	Sends an uninterpreted string of data	127
RENname	Renames a file on the foreign host	128
RESTart	Restarts a checkpointed data transfer	129
RMDir	Removes a directory	129
SCHINESE	Sets the transfer type to SCHINESE	130
SENDPort	Enables or disables automatic transmission of the FTP server PORT subcommand	131
SENDSite	Enables or disables automatic transmission of the SITE subcommand	132
SIte	Sends information to the foreign host using site-specific commands	132
SJiskanji	Sets the transfer type to SJISKANJI	146
STAtus	Displays status information for the foreign host	147
STRuct	Sets the file transfer structure	165
SUnique	Toggles the storage methods	165
SYstem	Displays the name of the foreign host's operating system	166
TChinese	Sets the transfer type to TCHINESE	166
TSO	Passes a TSO command to the local host TSO environment	167
TYpe	Specifies the transfer type	168
User	Identifies you to a foreign host or changes your TSO user ID password	172

ACCOUNT Subcommand—Supply Account Information

Purpose

Use the ACCOUNT subcommand to supply account information to a host.

Format



Parameters

account_information

Specifies the account information required by the host. See your foreign-host FTP server documentation for the information required by that host.

Usage

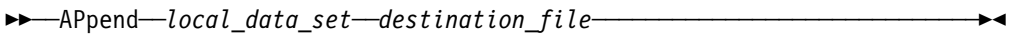
- The MVS FTP server does not require any account information.
- You may have to use the ACCOUNT subcommand when the foreign host requires passwords for read and write access to its files or data sets. If you are not prompted by the foreign host for the passwords, use the ACCOUNT subcommand to send these passwords to the foreign host.

APPEND Subcommand—Append a Local Data Set

Purpose

Use the APPEND subcommand to append a local data set to a remote host.

Format



Parameters

local_data_set
The name of the data set on your local host to be appended.

destination_file
The name of the file on the remote host to which your data set is appended. If the destination file does not already exist at the remote host, a new file is created. If the server is an OE server, the local file can be appended to as an HFS or an MVS data set.

Examples

In this example, an FTP command is issued from MVSXA2 to MVSXA3. MVSXA2 has a data set MVSUSER.FTP.EXAMPLE with one member. The member, APPEND01, contains:

```
;
; THIS FILE ORIGINALLY RESIDED IN MVSXA2, AND
; WILL BE APPENDED TO A FOREIGN FILE IN MVSXA3.
;
```

MVSXA3 has a data set, MVSUSER.FTP.EXAMPLE, with one member, APPEND02. The member contains:

```
;
; THIS FILE ORIGINALLY RESIDED IN MVSXA3, AND
; WILL BE USED TO RECEIVE ANOTHER FILE FROM MVSXA2.
;

System:   Command:

  User:   append

System:   Usage: APPEND localfile foreignfile

           Command:
```

```
  User:   lpwd
System:
           Local directory is MVSUSER.

           Command:
  User:
           append 'mvsuser.ftp.example(append01)' 'mvsuser.ftp.example(append02)'
System:   >>>SITE FIXrecfm 128 Lrecl=128 Recfm=FB BlockSize=6144
           200 Site command was accepted
          >>>PORT 1,1,2,2,4,16
           200 Port request OK.
           >>>APPE 'mvsuser.ftp.example(append02)'
           125 Appending to data set MVSUSER.FTP.EXAMPLE(APPEND02)
           250 Transfer completed successfully.
           520 bytes transferred in 1.100 seconds.
           Transfer rate 0.47 Kbytes/sec.
           Command:
```

Usage

- **Attention**

  FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the APPEND subcommand, FTP may truncate data records and you might lose data. If the data set name already exists at the receiving site and the logical record length (LRECL) of the data set at the receiving site is less than the LRECL of the transmitted data set, FTP truncates the transmitted data set.
- If the remote host is an MVS or VM host, and if the data set on the remote host has a fixed-record format, the format and record length of the data set on the remote host are always preserved.
- Records from the data set on your local host are truncated or padded with blank spaces when necessary.
- To append to a file on a remote host, you must have a defined working directory on that host, and you must have write privileges to the files in this working directory.

Context

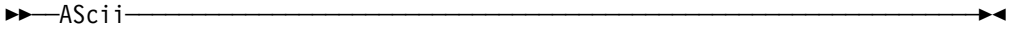
- See “CD Subcommand—Change the Directory on the Remote Host” on page 76 and “ACCOUNT Subcommand—Supply Account Information” on page 71 for more information about working with current directories.
- See Appendix A, “Specifying Data Sets and Files” on page 213, for more information about naming conventions.

ASCII Subcommand—Change the Data Transfer Type to ASCII

Purpose

Use the ASCII subcommand to change the data transfer type to ASCII.

Format



Usage

The ASCII transfer type is used to transfer data to or from an ASCII host. ASCII is the default transfer type.

Context

For more information about transfer methods, see Table 9 on page 25.

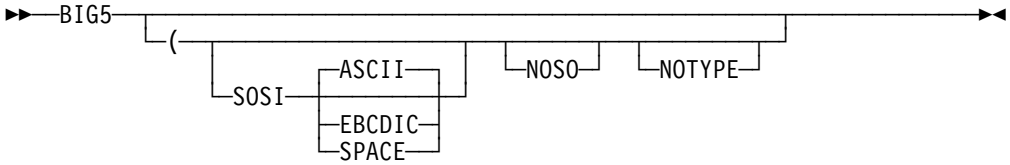
BIG5 Subcommand—Change the Data Transfer Type to BIG5

Purpose

Use the BIG5 subcommand to change the data transfer type to Big-5.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see "FTP Command—Enter the FTP Environment" on page 87 and "Changing Local Site Defaults Using FTP.DATA" on page 33.

Format



Parameters

SOSI

Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII

When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The BIG5 client subcommand is equivalent to the TYPE B 8 server command.

Context

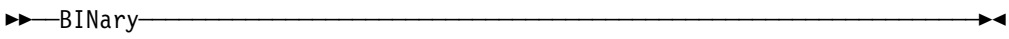
See “FTP with DBCS Support” on page 66 for more information.

BINARY Subcommand—Change the Data Transfer Type to Image

Purpose

Use the BINARY subcommand to change the data transfer type to image (binary).

Format



Parameters

None.

Usage

Use the image transfer type to transfer data from an ASCII host to an EBCDIC host. When using the image transfer type, data is sent as contiguous bits packed into 8-bit bytes. Use the image transfer type for efficient storage and retrieval of data sets or files, and for the transfer of binary data.

Context

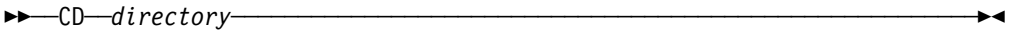
For more information about data transfer methods, see Table 9 on page 25.

CD Subcommand—Change the Directory on the Remote Host

Purpose

Use the CD subcommand to change the working directory or file group on the remote host.

Format



Parameters

directory

Specifies the name of a file directory, a fully qualified data set, or a prefix on the remote host.

Examples

Changing the Directory of an MVS FTP Server: If the remote host is using TCP/IP for MVS, *directory* specifies either a common prefix for a group of data sets or the qualifiers of a PDS. If the remote server is the MVS OE FTP server, *directory* can specify an HFS name.

When the CD subcommand is issued, the *directory* specified is appended to the current working directory. For example, if the current working directory is `TCPUSR14.TEST`, and you issue the CD subcommand:

```
CD FILES
```

then the new working directory becomes `TCPUSR14.TEST.FILES`.

To override the existing directory rather than append to the directory, issue the *directory* parameter in single quotation marks. For example, if the current working directory is `TCPUSR14.TEST`, and you issued the CD subcommand:

```
CD 'FTP.FILES'
```

then the new working directory would be `FTP.FILES`.

Note: If you leave a blank between the first single quotation mark and the *directory* parameter, FTP ignores the quotation marks. For example, if you want to change directory `TCPUSR14.TEST` to `FTP.FILES`, specify `CD 'FTP.FILES'`. However, if you specify `CD ' FTP.FILES'`, the directory becomes `TCPUSR14.TEST.FTP.FILES`.

If a PDS exists with the exact name of the current working directory, FTP considers the working directory to be that PDS. Otherwise, FTP considers the working directory to be a common prefix qualifier for sequential data sets.

If a PDS exists with the same name as the current working directory, but you want the current working directory to be treated as a common prefix for sequential data sets, specify the working directory with a period (.) at the end. For example, if a PDS named `TCPUSR14.TEST` exists, the subcommand:

```
CD 'TCPUSR14.TEST'
```

makes the PDS TCPUSR14.TEST the current working directory. A subsequent PUT of file name1 adds a member name1 to the TCPUSR14.TEST PDS. In contrast, the statement:

```
CD 'TCPUSR14.TEST.'
```

makes the current working directory, TCPUSR14.TEST., a prefix for sequential data sets. A subsequent PUT command used to copy data set name1 would create the sequential data set TCPUSR14.TEST.name1.

To back up one level of the current working directory, issue the CD subcommand with 2 periods (..) at the end. For example, if the working directory is jones.source, the statement:

```
CD ..
```

makes jones. the working directory.

You can also use the CDUP command to back up one level of the current working directory.

The following sample commands and responses are displayed as a result of the CD subcommand .

For an MVS data set:

```
cd hsmtest
 >>>CWD hsmtest
 250 "'USER17.HSMTEST.'" is working directory name prefix.
 Command:
```

For an HFS file:

```
cd '/u/user121/A/B/C'
  >>>CWD '/u/user121/A/B/C'
 250 HFS directory /u/user121/A/B/C is the current working directory

  Command:
```

Changing the Directory of a VM FTP Server: If the remote host is using TCP/IP for VM, the directory can be specified in either of the following ways:

user_id minidisk_address

or

user_id.minidisk_address

For example, to access the 191 minidisk of user ID jones, enter:

```
jones 191
```

or

```
jones.191
```

*Testing Throughput with *DEV.NULL:* If you have an MVS FTP server, for testing purposes, you can use the PUT command to copy a large number of files, or one large file, without having the file or files actually stored on an MVS server system.

Thus, you do not have to worry about allocating the disk space on the server system.

To use this function, first change the working directory to *DEV.NULL by using the following command:

```
CD *DEV.NULL
```

or

```
CWD *DEV.NULL
```

This will effect the working directory only for PUT commands. You then use the PUT command to copy the file to the server system. The input data set must be valid, and the output file can either be new or already exist. In either case, it is not actually stored.

The following response shows information such as the number of bytes transferred and the rate of transfer.

To end the use of the *dev.null directory for PUT commands, issue another cd command.

```
 Command:
cd *dev.null
 >>>CWD *dev.null
 250-Working directory for PUT is NULL Device;
 250 for GET is HFS directory /u/user31
 Command:
 put a.b a.bbbbb
 >>>SITE VARrecfm Lrecl=128 Recfm=VB BlockSize=6144
 200 Site command was accepted
 >>>PORT 14,0,0,0,4,14
 200 Port request OK.
 >>>STOR a.bbbbb
 125 Storing data set in the Null directory (*dev.null).
 250 Transfer completed successfully.
 82 bytes transferred in 0.245 seconds. Transfer rate 0.33 Kbytes/sec.
 Command:
 quit
 >>>QUIT
 221 Quit command received. Goodbye.
```

Usage

You can also use the CWD and CW subcommands to change the current working directory. These subcommands are synonyms of the CD subcommand.

Context

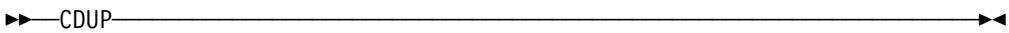
For more information on using the CDUP command, see “CDUP Subcommand—Change to the Parent of the Working Directory.”

CDUP Subcommand—Change to the Parent of the Working Directory

Purpose

Use the CDUP subcommand as a special case of the CD subcommand to change the working directory to the next higher directory level. You can use it to simplify the implementation of programs for transferring directory trees between operating systems that have different syntaxes for naming the parent directory.

Format



Parameters

None.

Examples

Change the working directory to the next higher directory level:

```
cd 'a.b.c.d'
  >>>CWD 'a.b.c.d'
 257 "'A.B.C.D.'" is working directory name prefix.

  Command: pwd

  >>>PWD
 257 "'A.B.C.D.'" is working directory

  Command: cdup

  >>>CDUP
 257 "'A.B.C.'" is working directory name prefix.

  Command: pwd

  >>>PWD
 257 "'A.B.C.'" is working directory

  Command: cdup

  >>>CDUP
 257 "'A.B.'" is working directory name prefix.

  Command: pwd

  >>>PWD
 257 "'A.B.'" is working directory
```

Change the working directory to the next higher directory level for an HFS file:

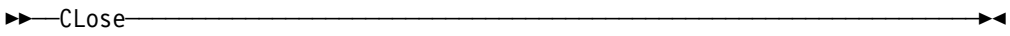
```
cd '/u/user121/A/B/C'
  >>>CWD '/u/user121/A/B/C'
 250 HFS directory /u/user121/A/B/C is the current working directory

  Command: pwd

  >>>PWD
 257 "/u/user121/A/B/C" is the HFS working directory

  Command: cdup

  >>>CDUP
 250 HFS directory /u/user121/A/B is the current working directory

  Command: pwd

  >>>PWD
 257 "/u/user121/A/B" is the HFS working directory

  Command: cdup

  >>>CDUP
  250 HFS directory /u/user121/A is the current working directory

  Command: pwd

  >>>PWD
 257 "/u/user121/A" is the HFS working directory.
 Command:
```

CLOSE Subcommand—Disconnect from a Remote Host

Purpose

Use the CLOSE subcommand to disconnect from the remote host and remain in FTP.

Format



Parameters

None.

Usage

The FTP session remains active on your local host, but the session to the remote host is terminated. You can use the OPEN subcommand to establish a new session with either the same or a different remote host. If you establish a new session with the same remote host, values set by the SITE subcommand during the previous session are cleared. The remote host's default values for the parameters of the SITE subcommand are used for the new session.

Context

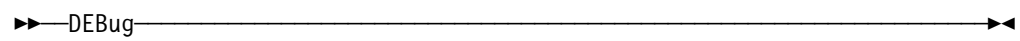
See “OPEN Subcommand—Connect to the FTP Server” on page 123 for information about the OPEN subcommand.

DEBUG Subcommand—Toggle Internal Debug Options

Purpose

Use the DEBUG subcommand to enable or disable internal debugging for the FTP client.

Format



Parameters

None.

Examples

This example shows how to obtain status and system information with and without the DEBUG command.

```
System:   READY
  User:   ftp 9.67.113.24 621
System:
           MVS TCP/IP FTP V3R2
           Connecting to 9.67.113.24, port 621
           220-FTPD1 IBM MVS V3R3 at MVSVIC03.TCP.RALEIGH,IBM.COM, 15:45:25
               on 1997-03-17.
           220 Connection will not timeout.
           USER (identify yourself to the host):
  User:   user121
System:
           >>>USER user121
          331 Send password please.
           Password:

           >>>PASS ********
          230 RTP is logged on.    Working directory is "/u/user121".

          Command:
  User:   system
System:   >>>SYST
          215 MVS is the operating system of this server.  FTP Server is the
              C-server running on OE.
          Command:
  User:   noop
System:   >>>NOOP
           200 Ok

           Command:
```

```
  User:   debug
System:   Internal debug options enabled

           Command:
  User:   system
System:   >>>SYST
          In SysSendFlush, calling TcpWaitSend with args: 0 00088788 6
          In SysSendFlush, TcpWaitSend returned: OK
          In SysRead, calling TcpWaitReceive with args: 0 0008AC6C 8192
          In SysRead: Note received: => TcpId 0 Data delivered (8682) 91 bytes Push
          In SysRead, TcpWaitReceive returned: 91
          215 MVS is the operating system of this server.  FTP Server is the
              C-Server running on OE.
          215 OS/MVS is the operating system of this server.
          GetReply returns 215

           Command:
  User:   noop
System:   >>>NOOP
          In SysSendFlush, calling TcpWaitSend with args: 0 00088788 6
          In SysSendFlush, TcpWaitSend returned: OK
          In SysRead, calling TcpWaitReceive with args: 0 0008AC6C 8192
          In SysRead: Note received:  => TcpId 0 Data delivered (8682) 8 bytes Push
          In SysRead, TcpWaitReceive returned: 8
          200 Ok
          GetReply returns 200

          Command:
  User:   quit
System:   >>>QUIT
          In SysSendFlush, calling TcpWaitSend with args: 0 00088788 6
          In SysSendFlush, TcpWaitSend returned: OK
          In SysRead, calling TcpWaitRecei with args: 0 0008AC6C 8192
          In SysRead: Note received: => TcpId 0 Data delivered (8682) 37 bytes Push
          In SysRead, TcpWaitReceive returned: 37
          221 Quit command received. Goodbye.
          GetReply returns 221
           SysClose called with FD = 1
          In SysClose: Note received: => TcpID 0 Connection state changed (8681)
          sending only (8675)
          In SysClose Newstate with FD = 1
          In SysClose: Note received: => TcpID 0 Connection state changed (8681)
          connection closing (8670)
          In SysClose Newstate connection closing (8670) for FD 1
          Exiting from SysClose: FD = 1, TcpID = 0
          SysHalt has been called
          READY
```

Usage

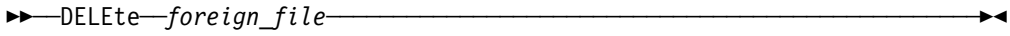
- The DEBUG subcommand is used for debugging. When DEBUG is on, FTP client displays internal tracing information.
- DEBUG acts as a toggle that turns the debugging option on or off. By default, DEBUG is off unless the TRACE parameter was specified on the FTP command.

DELETE Subcommand—Delete Files

Purpose

Use the DELETE subcommand to delete a file on the remote host.

Format



Parameters

foreign_file
Specifies the name of the file to be deleted on the remote host.

Context

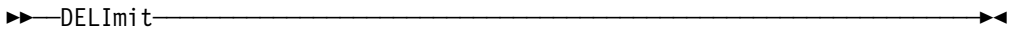
See Appendix A, “Specifying Data Sets and Files” on page 213, for information about file naming conventions.

DELIMIT Subcommand—Display the File Name Delimiter

Purpose

Use the DELIMIT subcommand to display the character that is used as the delimiter between the file name and the file type.

Format



Parameters

None.

Usage

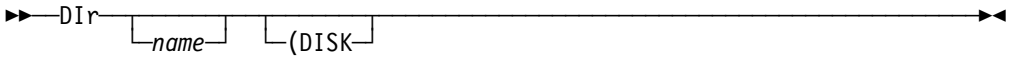
- The DELIMIT subcommand should be used for information purposes only.
- You cannot change which character is used as the delimiter.

DIR Subcommand—Obtain a List of Directory Entries

Purpose

Use the DIR subcommand to obtain a list of directory entries or a list of files in a file group on the remote host, or a list of the members of the partitioned data set, as well as auxiliary information about the files.

Format



Parameters

name
Specifies the name of the directory or file group. The default is the current directory or file group.

DISK
Stores the results of the DIR subcommand as data set FTP.DIROUTP in the local current working directory. If the local current working directory is an MVS PDS, then the member DIROUTP is stored.

Examples

- List the data sets with a common high-level qualifier as the current working directory:

```
 >>>PORT 9,67,58,227,4,62
 200 Port request OK.
 >>>LIST
 125 List started OK.
 Volume Unit    Referred Ext UsedRecfm Lrecl BlkSz Dsorg Dsname
 WRKLB2 3380   1996/07/02  1    1 F       80    80  PS  TEST.A
 WRKLB2 3380   1996/07/09  4    7 F       80    80  PO  TEST.PDS
 WRKLB2 3380   1996/07/09  2    3 VB      60   256  PO  TEST.PDSV
 WRKLB2 3380   1996/05/09  1    1 F       80    80  PO  TEST.DATA
 WRKLB2 3380   1996/07/01 16   17 V      200  3120  PS  TEST.XYZ
 WRKLB2 3380   1996/07/01 16   17 VB     200 32000  PS  TEST.XYZ1
 250 List completed successfully.
 Command:
```

List the files for an HFS directory:

```
cd '/u/user121/ftp.example'

 >>>CWD '/u/user121/ftp.example'
 250 HFS directory /u/user121/ftp.example is the current working directory
 Command:
dir
 >>>PORT 9,67,112,25,4,61
 200 Port request OK.
 >>>NLST
 125 List started OK
 total 64
 -rw-r-----    1 USER121  SYS1      6720 Feb  7 18:48 append02
 -rw-r-----    1 USER121  SYS1      3360 Feb  6 18:51 file1
 -rw-r-----    1 USER121  SYS1      3883 Feb  6 18:51 file2
 -rw-r-----    1 USER121  SYS1      3883 Feb  6 18:51 file3
 -rw-r-----    1 USER121  SYS1      7277 Feb  6 18:51 file4
 -rw-r-----    1 USER121  SYS1      3360 Feb  6 18:51 file5
  250 List completed successfully.
 Command:
```

- List the members of a partitioned data set containing load modules:

```
cd 'tcpv3.sezaxawl'
 >>>CWD 'tcpv3.sezaxawl'
 257 "'TCPV3.SEZAXAWL'" partitioned data set is working directory.
 Command: dir

 >>>PORT 9,67,112,25,4,27
 200 Port request OK.
 >>>LIST
 125 List started OK.
  Name      Size   TTR   Alias-of AC --------- Attributes --------- Amode Rmode
 @XADFTT1  0009F0 000A11 TEXTTR   00 FO                               31    ANY
 @XADFTT2  0009F0 000A11 TEXTTR   00 FO                               31    ANY
 @XADFTT3  0009F0 000A11 TEXTTR   00 FO                               31    ANY
 @XATXDRA  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXDSA  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXIFA  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXIFI  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXPSA  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXSCH  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXSTF  003998 00090C TEXTPOP  00 FO                               31    ANY
 @XATXZSL  005500 00080F TEXTACTI 00 FO                               31    ANY
 ACOMMAND  001420 00020A          00 FO                               31    ANY
 AFORM     001AC0 00021F          00 FO                               31    ANY
 ***
```

- List the members of a partitioned data set from a text library:

```
cd 'tcpv3.tcpip.profiles'
 >>>CWD 'tcpv3.tcpip.profiles'
 257 "'TCPV3.TCPIP.PROFILES'" partitioned data set is working directory.
 Command: dir

 >>>PORT 9,67,112,25,4,32
 200 Port request OK.
 >>>LIST
125 List started OK.
 Name     VV.MM    Created       Changed     Size  Init   Mod   Id
TST6MV1   01.05 1996/06/26 1996/07/10 06:38    16    16     0 USER34
TST6MV2   01.08 1996/05/23 1996/07/03 12:49    16    17     0 USER34
TST6MV3   01.19 1996/05/23 1996/07/10 06:34    16    17     0 USER34
TST6021   01.04 1996/03/04 1996/07/08 09:17    15    15     0 USER34
TST6121   01.10 1996/05/23 1996/07/10 06:26    16    17     0 USER34
250 List completed successfully.
 ***
```

Usage

- To make a file group the current working directory, use the CD command. The method you use to specify a directory or file group is host-dependent.
- You can use special characters for pattern matching when specifying the *name*. These characters depend on the host's FTP server.
- The DIR subcommand provides a complete list of directory entries and gives additional information about the files.

Context

- See Appendix A, "Specifying Data Sets and Files" on page 213, for more information about pattern matching and about specifying data sets and files.
- To get a list containing only the file names in a directory, use the LS subcommand (see "LS Subcommand—Obtain a List of File Names" on page 113).
- To make a file group the current working directory, see "CD Subcommand—Change the Directory on the Remote Host" on page 76.

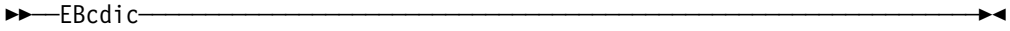
- To change the local directory, see “LCD Subcommand—Change the Local Working Directory” on page 97.

EBCDIC Subcommand—Change the Data Transfer Type to EBCDIC

Purpose

The EBCDIC subcommand allows you to change the data transfer type to EBCDIC.

Format



Parameters

None.

Usage

The EBCDIC transfer type is used to transfer data to or from an EBCDIC host.

Context

For more information about transfer methods, see Table 9 on page 25.

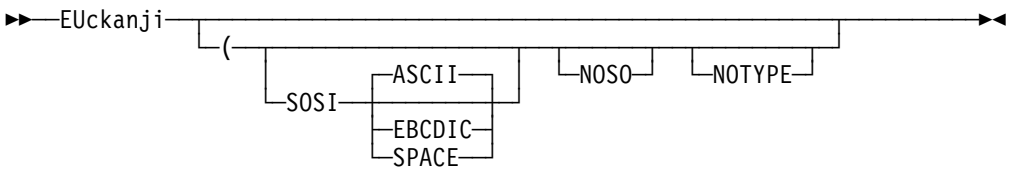
EUCKANJI Subcommand—Change the Data Transfer Type to EUCKANJI

Purpose

Use the EUCKANJI subcommand to change the data transfer type to Extended UNIX Code (EUC) kanji.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

SOSI
Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII
When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The EUCKANJI client subcommand is equivalent to the TYPE B 2 server command.

Context

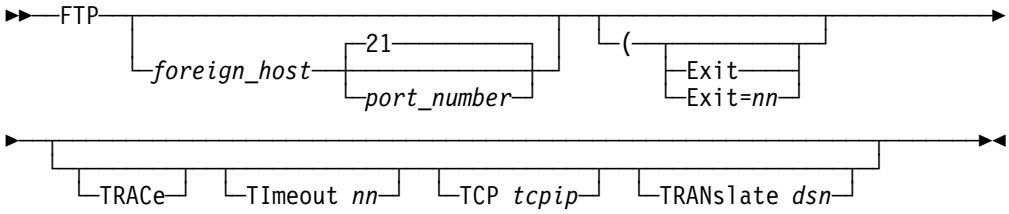
See “FTP with DBCS Support” on page 66 for more information.

FTP Command—Enter the FTP Environment

Purpose

Use the FTP command to enter the FTP environment.

Format



Note: The minimum abbreviation for each parameter is shown in uppercase letters.

Parameters

foreign_host
Specifies the name of the host to which you are connecting. Specify the host by its host name or its IP address. The host can be a remote host or your local host.

You are prompted for a host name if you do not specify a *foreign_host* with the FTP command. If you specify *foreign_host* incorrectly, or if the host is not accessible, you enter the FTP environment without connecting to a host. You should then use either the OPEN subcommand to attempt another connection with a host, or the QUIT subcommand to exit the FTP environment.

port_number
Specifies the port number of the FTP server on the remote host. The default is well-known port 21. The maximum port number that can be specified is 65 534.

Exit
Terminates FTP, for certain FTP errors, with a non-zero MVS return code.

Exit=*nn*
Terminates FTP with a nonzero return code of your choice when an FTP error occurs. Valid values are 0–4095.

TRACe
Starts the generation of tracing output. TRACe is used to assist in debugging.

TImeout *nn*
Specifies the number of seconds to be used for the following TImeout parameters:

MyopenTime (default=60)
Defines the amount of time to wait when waiting for a session to open before terminating the attempt and reporting an error.

DconnTime (default=120)
Defines the amount of time to wait after attempting to close a data transfer before terminating the connection and reporting an error.

CconnTime (default=30)
Defines the amount of time to wait after attempting to close a control connection before terminating the connection and reporting an error.

InactTime (default=120)
Defines the amount of time to wait with no activity for a session before terminating it and reporting an error.

DataCtTime (default=120)
Defines the amount of time to wait after attempting to send or receive data before terminating the connection and reporting an error.

Numeric values between 15 and 720 are accepted for the TImeout parameter. All 5 of the TImeout parameters are set to the value you choose at invocation.

If you want to individually customize the values, you must use the FTP.DATA data set. If an FTP.DATA data set exists but you also specify a time-out value at invocation, the TImeout parameter overrides the FTP.DATA data set.

If you specify an incorrect value for the TImeout parameter, FTP uses the default value.

TCP *tcpip*
Indicates the name of the TCP on the local host that the FTP client should connect to if more than one TCPIP is active.

Note: You must specify this value as a parameter, not as a value in the FTP.DATA data set. You may choose to specify this value with the TCPIPJOBNAME statement in the TCPIP.DATA data set.

TRANslate *data_set_name*
Specifies the name of a nonstandard translation table. If you specify this parameter, FTP uses the translation table in the *user_id.data_set_name.*TCPXLBIN data set, rather than the standard translation table provided with TCP/IP (*hlq*.STANDARD.TCPXLBIN). The *hlq*.STANDARD.TCPXLBIN data set is never used if you specify the TRANSLATE parameter.

If *user_id.data_set_name.*TCPXLBIN does not exist,

If *user_id.data_set_name.*TCPXLBIN and or if they were incorrectly created, FTP ends with an error message.

If LOADDBCSTABLES is specified in *hlq*.TCPIP.DATA, then *data_set_name* is used to determine which DBCS translation table to load. In order to load and use a customized DBCS translation table, an SBCS table must be defined as hlq.data_set_name.TCPXLBIN or userid.data_set_name.TCPXLBIN or the FTP request will fail. A copy of hlq.STANDARD.TXPXLBIN can be used if you do not require a modified SBCS table.

Attention

If the SBCS translation table that you need for data transfers does not support standard encodings for the portable character set, do not use the TRANSLATE option of the FTP command because such a table can adversly effect the EBCDIC to ASCII conversion of FTP commands. Instead, use the CCTRANS and SBTRANS statements in your local FTP.DATA to specify different SBCS tables for the control and data connections. For information on specifying these statements, see “Using Different SBCS Translation Tables for the Control and Data Connections” on page 39. If you also require a local DBCS translation table, you can name it *userid*.FTP.TCPdbBIN and it will be found in the client's search order. See the *TCP/IP for MVS: Customization and Administration Guide* for information about the DBCS translation table search order.

Usage

You will receive the following message when starting FTP in a TSO environment that includes support for the REXX programming language:

```
CSV003I Requested module IRXSTK not found
```

This is a normal informational message in this circumstance.

Context

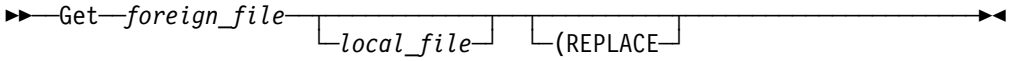
- See “OPEN Subcommand—Connect to the FTP Server” on page 123 and “QUIT Subcommand—Leave the FTP Environment” on page 126 for more information about the OPEN and QUIT subcommands.
- See “FTP EXIT Return Codes” on page 48 for more information about the EXIT return codes.
- See “Changing Local Site Defaults Using FTP.DATA” on page 33 for information about the FTP.DATA data set.
- See the *OS/390 TCP/IP OpenEdition Configuration Guide* for information about the TCPIP.DATA data set or loading and customizing DBCS translation tables.

GET Subcommand—Copy Files

Purpose

Use the GET subcommand to copy a file from the remote host to your local host.

Format



Parameters

foreign_file
Specifies the name of the file to be retrieved from the remote host.

local_file
Specifies the name of the local file created as a result of the GET subcommand. If the current local working directory is a PDS, *local_file* is the name of the member in the PDS. If the current local working directory is a data set prefix, the local file is a sequential data set with the *local_file* name appended to the current local working directory. You can override the usage of the current local working directory in the local file name by specifying *local_file* as a complete data set name enclosed in single quotation marks (’). If *local_file* is not specified, the *local_file* name is the same as the *foreign_file* name.

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 data set already exists and you specify REPLACE, the data in the file is overwritten, but not reallocated. This means the local data set retains its existing characteristics.

Usage

- FTP uses either the characteristics of the local file, if it exists, or uses the values specified with the LOCSITE subcommand. Characteristics of the transmitted (foreign file) data set are unknown.

  When you use the GET subcommand, FTP may truncate data records and you might lose data:

  – When you are creating a new data set at the client and the value of LRECL, as shown by the LOCSTAT command, is a value less than the LRECL of a received data set, FTP truncates the received data set.

  – If the data set name already exists at the receiving site and the logical record length (LRECL) of the data set at the receiving site is less than the LRECL of the transmitted data set, FTP truncates the transmitted data set.

    You can encounter this situation when you use GET with the REPLACE option.

- A GET subcommand performed on an empty data set erases the contents of the existing local data set.
- If the name specified for *local_file* is not acceptable to your local host, the file is not transferred.
- To get a file from the remote host, you must have a defined working directory on that host and you must have read privileges to the files in this working directory.

Context

- See Appendix A, “Specifying Data Sets and Files” on page 213, for more information about naming conventions.
- See “CD Subcommand—Change the Directory on the Remote Host” on page 76 and “ACCOUNT Subcommand—Supply Account Information” on page 71 for more information about working directories.

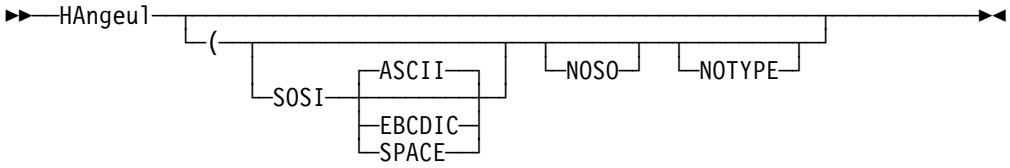
HANGEUL Subcommand—Change the Data Transfer Type to HANGEUL

Purpose

Use the HANGEUL subcommand to change the data transfer type to Hangeul.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

SOSI
Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII
When combined with the SOSI parameter, causes shift-out and shift-in characters `X'1E'` and `X'1F'` to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters `X'0E'` and `X'0F'` to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters `X'20'` and `X'20'` (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The HANGEUL client subcommand is equivalent to the TYPE B 5 server command.

Context

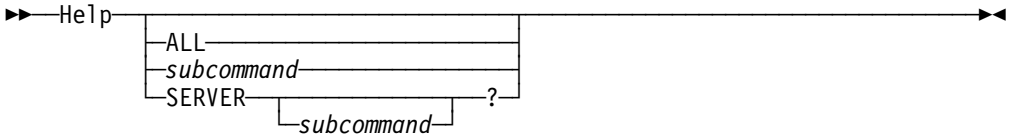
See “FTP with DBCS Support” on page 66 for more information.

HELP and ? Subcommands—Display Help Information

Purpose

Use the HELP command to get assistance with the FTP subcommands.

Format



Parameters

ALL
Displays a description of all subcommands.

subcommand
Displays a description of the specified subcommand. The subcommand name can be abbreviated to its minimum abbreviation.

SERVER
Displays the help offered by the foreign host for the internal FTP commands.

Examples

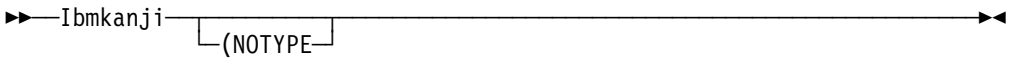
- If you enter the HELP subcommand without a parameter, you see the HELP FTP MENU, which lists the subcommands recognized by the FTP client and a description of the help information available.
- If you enter the ? subcommand by itself, you see introductory information about FTP.

IBMKANJI Subcommand—Change the Data Transfer Type to IBMKANJI

Purpose

Use the IBMKANJI subcommand to change the data transfer type to IBM kanji.

Format



Parameters

NOTYPE
Suppresses sending of the TYPE command for host servers that do not support this data transfer type.

Usage

This subcommand actually causes no conversion to be performed on the transferred file; it has exactly the same effect as the EBCDIC TYPE command alias.

Context

See “FTP with DBCS Support” on page 66 for more information.

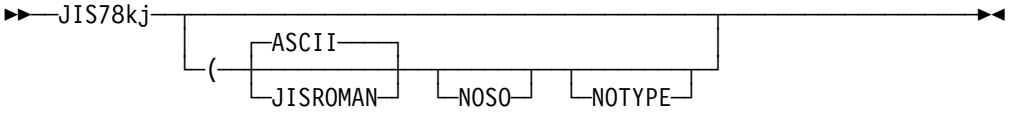
JIS78KJ Subcommand—Change the Data Transfer Type to JIS78KJ

Purpose

Use the JIS78KJ subcommand to change the data transfer type to JIS78KJ (1978 edition).

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

ASCII
Uses ASCII shift-in escape sequence `ESC ( B` in the transferred data.

If neither ASCII nor JISROMAN is specified, then the ASCII shift-in sequence will be used.

JISROMAN
Uses JISROMAN shift-in escape sequence `ESC ( J` in the transferred data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

- The JIS78KJ or JIS78KJ (ASCII client subcommands are equivalent to the TYPE B 4 A server command.
- The JIS78KJ (JISROMAN client subcommand is equivalent to the TYPE B 4 R server command.
- The JIS78KJ (JISROMAN NOSO client subcommand is equivalent to the TYPE B 4 R N server command.

Context

See “FTP with DBCS Support” on page 66 for more information.

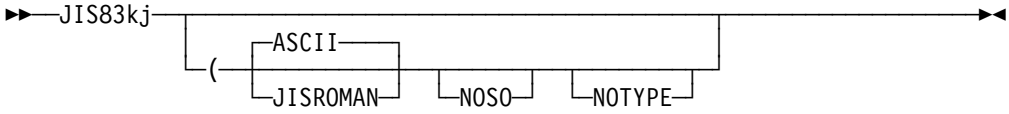
JIS83KJ Subcommand—Change the Data Transfer Type to JIS83KJ

Purpose

Use the JIS83KJ subcommand to change the data transfer type to JIS83KJ (1983 edition).

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

ASCII
Uses ASCII shift-in escape sequence `ESC ( B` in the transferred data.

If neither ASCII nor JISROMAN is specified, then the ASCII shift-in sequence will be used.

JISROMAN
Uses JISROMAN shift-in escape sequence `ESC ( J` in the transferred data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

- The JIS83KJ or JIS83KJ (ASCII client subcommands are equivalent to the TYPE B 3 A server command.
- The JIS78KJ (JISROMAN client subcommand is equivalent to the TYPE B 3 R server command.

Context

See “FTP with DBCS Support” on page 66 for more information.

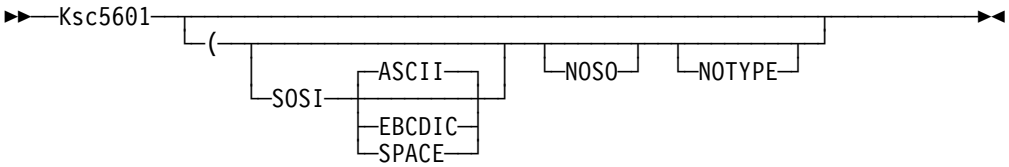
KSC5601 Subcommand—Change the Data Transfer Type to KSC-5601

Purpose

Use the KSC5601 subcommand to change the data transfer type to KSC-5601.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

SOSI
Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII
When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The KSC5601 client subcommand is equivalent to the TYPE B 6 server command.

Context

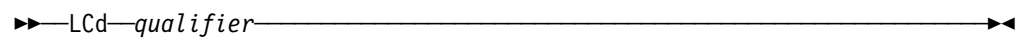
See “FTP with DBCS Support” on page 66 for more information.

LCD Subcommand—Change the Local Working Directory

Purpose

Use the LCD subcommand to change the current working directory on the local host.

Format



Parameters

qualifier
Specifies either a common prefix for a group of sequential data sets or the qualifiers of a PDS.

Examples

- Change the local current working directory:

```
lcd ftp.test1

 Local directory name set to partitioned data set USER14.FTP.TEST1.
 Command:
```

- When the LCD subcommand is issued, *qualifier* is appended to the current local working directory. For example, if the current local working directory is `TCPUSR14.TEST`, and you issue the LCD subcommand:

  ```
  LCD FILES
  ```

  the new working directory becomes `TCPUSR14.TEST.FILES`.

- To override the existing directory rather than append to the directory, issue the *qualifier* in single quotation marks. For example, if the current local working directory is `TCPUSR14.TEST`, and you issued the LCD subcommand:

  ```
  LCD 'FTP.FILES'
  ```

then the new working directory is `FTP.FILES`.

- If a PDS exists with the exact name of the current local working directory, FTP considers the working directory to be that PDS. Otherwise, FTP considers the working directory to be a common prefix qualifier for sequential data sets.

  If a PDS exists with the same name as the current local working directory, but you want the current local working directory to be treated as a common prefix for sequential data sets, specify the working directory with a period (.) at the end. For example, if a PDS named `TCPUSR14.TEST` exists, the subcommand:

  ```
  LCD 'TCPUSR14.TEST'
  ```

  makes the PDS `TCPUSR14.TEST` the current local working directory. A subsequent GET command used to copy data set `name1` would add the member `name1` to the `TCPUSR14.TEST` PDS. In contrast, the statement:

  ```
  LCD 'TCPUSR14.TEST.'
  ```

  would make the current local working directory, `TCPUSR14.TEST.`, a prefix for sequential data sets. A subsequent GET command used to copy data set `name1` would create the sequential data set `TCPUSR14.TEST.name1`.

- To back up one level of the current local working directory, issue the LCD subcommand with 2 periods (..) at the end. For example, if the working directory is `jones.source`, the statement:

  ```
  LCD ..
  ```

  makes `jones.` the working directory.

Usage

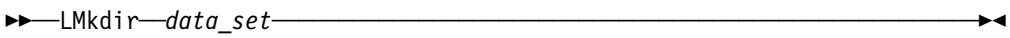
When you enter an FTP session, the working directory on the local host is set to either the TSO prefix or the TSO user ID if the TSO prefix does not exist.

LMKDIR Subcommand—Create a PDS on the Local Host

Purpose

Use the LMKDIR subcommand to create a PDS on the local host. This subcommand provides you with an easy way to create a directory in the local host for data transfer.

Format



Parameters

data_set
Specifies the name of the PDS to be created.

Examples

In this example, before LMKDIR is issued, the local host had the following data sets:

- MVSUSER.ISPF.ISPPROF
- MVSUSER.JCL.CNTL
- MVSUSER.SMFTEST
- MVSUSER.TCPIP.DATA

```
System:   READY
  User:   ftp 1.1.2.3
System:   MVS TCP/IP FTP V3R2
          220-EZAFTSRV IBM MVS V3R2 at EMU.ABC.OZ, 17:04:12 on 08/03/96
          220 Connection will close if idle for more than 5 minutes.
          USER (identify yourself to the host):
  User:   mvsuser
System:   >>>USER mvsuser
          331 Send password please.
           Password:
          >>>PASS ********
          230 MVSUSER is logged on.
          Command:
  User:   lpwd
System:   Local directory is MVSUSER.
          Command:
  User:   lcd ftp
System:   Local directory name set to MVSUSER.FTP.
          Command:
  User:   lmkdir example
System:   MVSUSER.FTP.EXAMPLE created.
          Command:
```

`MVSUSER.FTP.EXAMPLE` has now been created. You can get the same result directly with the `LMKDIR 'MVSUSER.FTP.EXAMPLE'` command.

After the LMKDIR was issued, the local host had the following data sets under MVSUSER:

- MVSUSER.FTP.EXAMPLE
- MVSUSER.ISPF.ISPPROF
- MVSUSER.JCL.CNTL
- MVSUSER.SMFTEST
- MVSUSER.TCPIP.DATA

Usage

- There is no FTP subcommand to display a list of local directory entries. You should use TSO's ISPF facility to check whether the directory is created by the LMKDIR subcommand.
- The *data_set* value is appended to the local current working directory to form the name of the created PDS. To override the local current working directory, specify:

  '*data_set*'

  The data set characteristics of the newly allocated PDS are determined by the settings of the local site variables.

Context

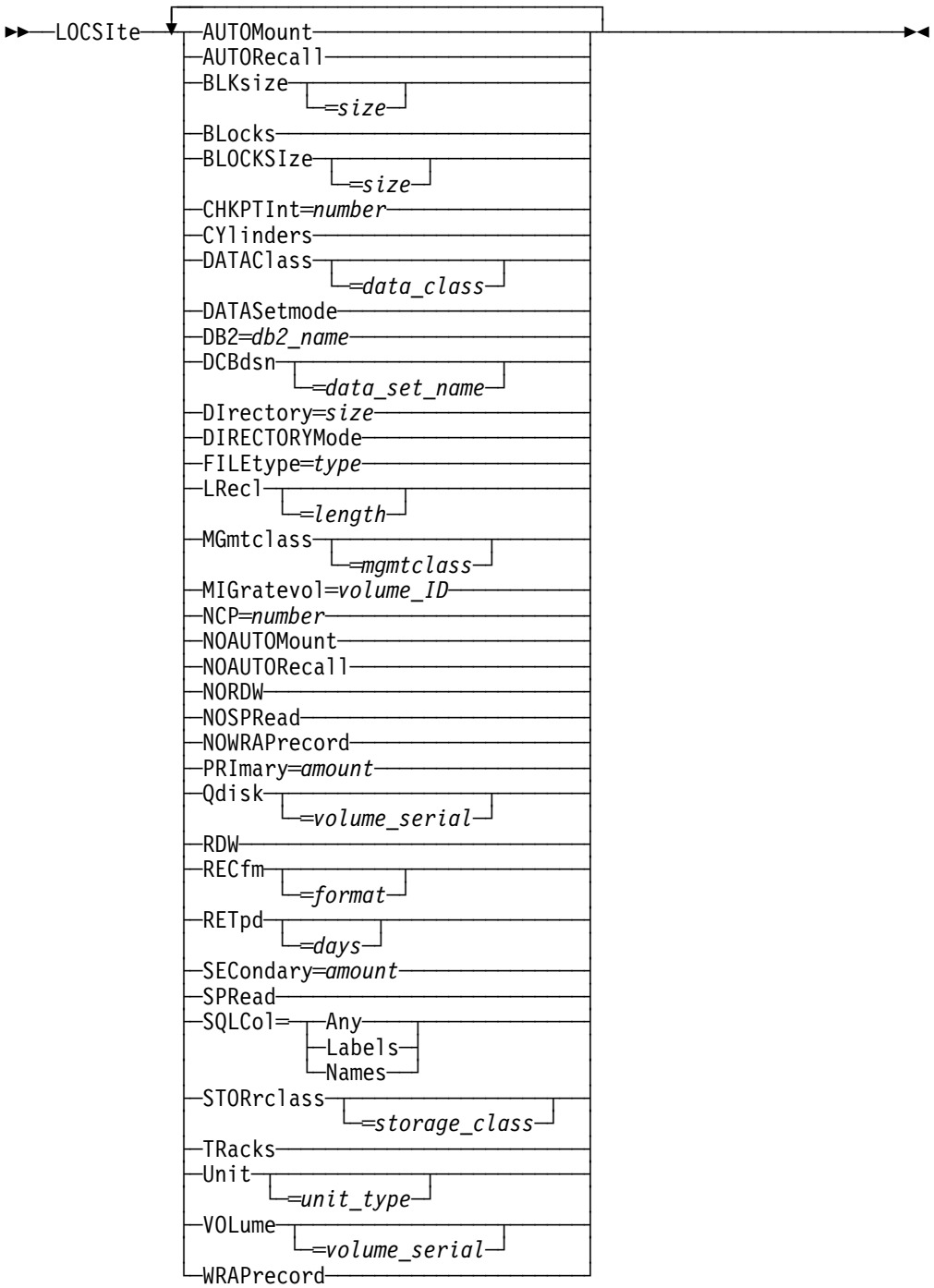
See “LOCSITE Subcommand—Specify Site Information to the Local Host” and “LOCSTAT Subcommand—Display Local Status Information” on page 110 for information on setting and displaying the data set characteristics.

LOCSITE Subcommand—Specify Site Information to the Local Host

Purpose

Use the LOCSITE subcommand to specify information that is used by the local host to provide services specific to that host system.

Format



Parameters

AUTOMount

Permits automatic mounting of volumes for data sets on volumes that are not mounted. If AUTOMount is specified and an unmounted volume is needed, a message is automatically issued to the MVS operator console requesting that the volume be mounted. The MVS operator must then mount the volume and reply to the message before FTP can proceed.

AUTORecall
Permits automatic recall of migrated data sets.

BLKsize
Specifies the block size of a newly allocated data set. BLKSIZE is functionally equivalent to BLOCKSIZE. The BLOCKSIZE parameter is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS.

When specified without a *size* or BLKSIZE=0, no block size will be used when allocating the data set. When specified without a *size*, the equal sign (=) is optional.

Specify BLKSIZE with no value if you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the BLKSIZE value, or if you are specifying DCBDSN=*data_set_name* and you want to use the blocksize from the DCBDSN data set. If BLKSIZE=*size* is specified as non-zero and with either the DATACLASS or DCBDSN parameter, the value specified by the LOCSITE BLKSIZE parameter overrides the DATACLASS or DCBDSN blocksize.

size
Specifies the block size of a newly allocated data set. The valid range is 0 through 32 760.

Note: If you specify BLKSIZE without a size or with size of 0, FTP will not specify the block size when allocating new data sets.

BLocks
Specifies that primary and secondary space allocations are in blocks.

If both PRIMARY and SECONDARY are specified as 0, and an SMS data class has been specified, then the space allocation is determined by the SMS data class and the BLocks parameter is ignored.

BLOCKSIze
Specifies the block size of a newly allocated data set. BLOCKSIze is functionally equivalent to BLKSIZE. BLOCKSIZE is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS. See the BLKSIZE parameter.

CHKPTInt
Specifies the checkpoint interval for the sending site in a file transfer request. If the checkpoint interval is zero, no checkpointing occurs and no marker blocks are transmitted. The default value is zero.

CHKPTInt must be set by LOCSITE when the client site is the sending site for the file transfer (PUT command). Do not increase the checkpoint interval above zero for a PUT unless the receiving site supports the RESTART command and is able to process checkpoint markers in the file transfer data stream.

CHKPTInt is valid only for block and compressed modes of data transfer.

number
The checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted so that transmission can be restarted based on the information in the last marker.

A large checkpoint interval means that a large amount of data is sent in between markers and therefore few markers are sent. A smaller checkpoint interval means that less data is sent in between markers and therefore more markers are sent.

The costs involved with using a nonzero checkpoint interval are:

- The markers themselves are transmitted, which means more bytes being sent across the network (approximately 44 bytes per marker).
- Additional packets and acknowledgements may be required. For example, when the MVS client PUTs a file, a reply packet is sent from the server to the client and then the client must acknowledge that packet.

To use this formula to estimate the appropriate checkpoint interval, you need to know the record length of the file you are transferring and how much data you think can always be transmitted reliably.

```
           amount of data in interval
CHKPTINT = --------------------------
           record length of the file
```

We recommend that you do not checkpoint more often than once every 200KB. Therefore, if the file we are transferring has 80-byte records:

```
CHKPTINT = 200KB / 80 bytes
         = 200 * 1024 bytes / 80 bytes
         = 2560
```

CYlinders

Specifies that primary and secondary space allocations are in cylinders.

If both PRIMARY and SECONDARY are specified as 0, and an SMS data class has been specified, the space allocation will be determined by the SMS data class and the CYlinders parameter will be ignored.

DATAClass

Specifies the SMS data class as defined by your organization for the target host. Specifying DATAClass with no parameter value cancels the dataclass specification. The equal sign (=) is optional in this case.

See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.

data_class

Specifies the SMS data class as defined by your organization for the target host. If values are specified for any of the following LOCSITE parameters, the values specified by the LOCSITE parameter will override the value specified in the SMS dataclass:

- BLKSIZE
- DIRECTORY
- LRECL
- PRIMARY
- RECFM
- RETPD
- SECONDARY

If the DCBDSN parameter is specified on the LOCSITE subcommand, the LRECL, RECFM, BLKSIZE, and RETPD (if specified) of the DCBDSN data set will override the values specified in the dataclass.

If the MGMTCLASS parameter is specified on the LOCSITE subcommand, and the requested management class specifies a retention period, the retpd value of the management class may override the retpd value of the dataclass.

DATASetmode
Specifies that all the data set qualifiers below the current directory are treated as entries in the directory (disables DIRECTORYMode).

DB2
Specifies the name of the DB2 subsystem.

db2_name
The name of the DB2 subsystem.

DCBdsn
Specifies the name of the data set to be used as a model for allocation of new data sets. Specifying DCBdsn with no parameter value cancels the DCBdsn specification. The equal sign (=) is optional in this case.

data_set_name
Appended to the local current working directory. To override the local current working directory, specify *data_set_name* within single quotes. If any of the following LOCSITE parameters are also specified, the value specified by the LOCSITE parameter will override the value obtained from the data set DCB characteristics:

- BLKSIZE
- LRECL
- RECFM
- RETPD

DIrectory
Specifies the number of directory blocks to be allocated for the directory of a PDS.

Specify DIRECTORY=0 to allocate without specifying the number of directory blocks. Specify DIRECTORY=0 when you are also specifying DATACLASS=dataclass and you want the SMS dataclass to provide the DIRECTORY *size*.

The *size* you specify with the DIRECTORY parameter will override the DATACLASS directory specification.

size
Specifies the number of directory blocks to be allocated for the directory of a PDS. The valid range is 1 through 16 777 215. A *size* of 0 indicates that the directory blocks should be taken from the SMS dataclass.

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 data set qualifier is the only one used by the MPUT subcommand.

FILEtype
Specifies the file type of the data set.

type

The file type of the data set can be:

Type	Description
SEQ	Sequential or partitioned data sets
SQL	SQL query function

LRecl

Specifies the logical record length of a newly allocated data set.

Specify LRECL with no value when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the LRECL value, or when you are specifying DCBDSN=*data_set_name* and you want to use the LRECL from the dcbdsn data set. If LRECL=*length* is specified with either DATACLASS or DCBDSN, the length specified by the LOCSITE LRECL parameter will override the DATACLASS or DCBDSN LRECL.

length

Specifies the logical record length of a newly allocated data set. The valid range is 0 through 32 760.

Specifying `LRecl=0` has the same effect as specifying `LRrecl` with no parameters.

MGmtclass

Specifies the SMS management class as defined by your organization for the target host. Specifying MGmtclass with no *mgmtclass* cancels the mgmtclass specification. The equal sign (=) is optional in this case.

mgmtclass

Specifies the SMS management class as defined by your organization for the target host. If the mgmtclass specified has a setting for RETPD, the value specified by the mgmtclass may override the setting of the LOCSITE RETPD parameter, the RETPD value of a model data set if the DCBDSN parameter is specified, and the RETPD value defined in an SMS data class if DATACLASS is specified. See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.

MIGratevol

Specifies the volume ID for migrated data sets if they do not use IBM storage management systems. If you do not specify MIGratevol, the default volume_serial is MIGRAT.

volume_ID

The volume ID for migrated data.

NCP

Specifies the number of I/O buffers to be used during data transfer.

number

The number of I/O buffers to be used during data transfer. The valid range is 2 through 20.

NOAUTOMount

Prevents automatic mounting of volumes for data sets on volumes that are not mounted.

NOAUTORecall
Prevents automatic recall of migrated data sets.

Note: A migrated data set can be deleted even though NOAUTORECALL is specified, because migrated data sets are not recalled for deletion.

NORDW
Specifies that Variable Record Descriptors (RDW) are discarded during FTP transmission of variable format data sets.

NOSPRead
Specifies that the output is in report format rather than spreadsheet format when the file type is SQL.

NOWRAPrecord
Indicates that data is truncated if no new line character is encountered before the logical record length of the receiving file is reached. NOWRAPrecord is the default.

PRImary
Specifies the amount of tracks, blocks, or cylinders for primary allocation. When specified with a value of 0, no primary value will be used when allocating the data set.

Specify PRIMARY with 0 when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the PRIMARY *amount*

To allow the SMS data class to determine the space allocation, both PRIMARY and SECONDARY must be specified as 0. The tracks/blocks/cylinders setting will be ignored in this case. If PRIMARY with *amount* not equal to 0 is specified with DATACLASS, the value specified by the LOCSITE PRIMARY parameter will override the DATACLASS space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for primary allocation. For allocating partitioned data sets, this is the amount that will be allocated for the primary extent.

For allocating sequential data sets this is the maximum amount that will be allocated for the primary extent. If a lesser amount is needed to hold the data being transferred, only the amount actually needed to hold the data will be allocated. The valid range is 1 through 16 777 215.

Qdisk
Displays statistics about available space on a specific volume. If Qdisk is entered without a specific *volume serial*, statistics about available space are displayed for each volume that is defined with “Use Attribute=storage”.

volume_serial
Displays statistics about available space on a specific volume.

RDW
Specifies that Variable Record Descriptors (RDW) are treated as if they were part of the record and not discarded during FTP transmission of variable format data sets in stream mode.

Note: RDW information is stored in a binary halfword. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.

RECfm

Specifies the record format of a data set. When specified without the *format*, no record format will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify RECFM with no value when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the RECFM *format*, or when you are specifying DCBDSN=*data_set_name* and you want to use the record format from the dcbdsn data set.

If RECFM=*format* is specified with either DATACLASS or DCBDSN, the value specified by the LOCSITE RECFM parameter will override the DATACLASS or DCBDSN record format.

format

Specifies the record format of a data set. Valid record formats are: F, FA, FB, FBA, FBM, FBS, FM, FS, U, V, VA, VB, VBA, VBM, VBS, VM, and VS. The characters used to specify these record formats have the following meanings:

Code	Description
F	Fixed record length
V	Variable record length
U	Undefined record length
B	Blocked records
S	Spanned records
A	Records contain ISO/ANSI control characters
M	Records contain machine code control characters

REtpd

Specifies the number of days that a newly allocated data set should be retained.

Specify RETPD with no value when you are also specifying DATACLASS=*data_class* or MGMTCLASS=*mgmtclass* and you want SMS to provide the RETPD value, or when you are specifying DCBDSN=*data_set_name* and you want to use the RETPD from the dcbdsn data set. If more than one of the LOCSITE parameters (RETPD, MGMTCLASS, DATACLASS, or DCBDSN) are specified, the order of precedence (highest to lowest) is:

1. MGMTCLASS
2. RETPD
3. DCBDSN
4. DATACLASS

If a retention period is associated with an SMS management or data class, or with a model DCBDSN data set, the value of the retention period may be overridden to another non-zero value, but it may not be overridden to have no retention period specified for the newly created data sets.

days

Specifies the number of days that a newly allocated data set should be retained. The valid range is 0 through 9 999. A value of 0 indicates that no retention period should be assigned to the data set.

SECondary
Specifies the amount of tracks, blocks, or cylinders for secondary allocation.

Specify SECONDARY=0 when you are also specifying DATACLASS=*dataclass* and you want the SMS dataclass to provide the SECONDARY value. To allow the SMS data class to determine the space allocation, both PRIMARY and SECONDARY must be specified as 0. The tracks/blocks/cylinders setting will be ignored in this case. If SECONDARY is specified as other than 0 with DATACLASS, the value specified by the SITE SECONDARY parameter will override the DATACLASS space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for secondary allocation. The valid range is 0 through 16 777 215. If you specify an *amount* of 0, then FTP will allocate without specifying secondary space.

SPRead
Specifies that the output is in spreadsheet format when the file type is SQL.

SQLCol
Specifies the column headings of the SQL output file.

Any
The label of the DB2 SQL table column heading is the first choice for column heading, but if there is no label, the name becomes the column heading.

Labels
Labels are the DB2 SQL table column headings. If any of the columns do not have labels, the corresponding column heading in the output file is left blank.

Names
Uses the names of the DB2 SQL table column headings. The labels are ignored.

STOrclass
Specifies the SMS storage class as defined by your organization for the target host. When specified without a *storage_class*, no parameter value cancels the storage class specification. The equal sign (=) is optional in this case.

See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.

storage_class
Specifies the SMS storage class as defined by your organization for the target host. The SMS storage class may override settings for the VOLUME or UNIT site parameters.

TRacks
Specifies that primary and secondary space allocations are in tracks.

If both PRIMARY and SECONDARY are specified as 0, and an SMS data class has been specified, then the space allocation will be determined by the SMS data class and the TRacks parameter will be ignored.

Unit
Specifies the unit type for allocation of new data sets.

The setting for Unit may be overridden by the SMS storage class, if one is specified. We recommend that Unit not be coded if an SMS storage class is in use.

unit_type

The unit type (for example, 3380 or TAPE) for the allocation of new data sets. You can use the Unit parameter to specify either direct access or tape device types. If *unit_type* is not specified, the unit type used for allocation is set back to the system default.

For tape output, only standard label (SL) and file sequence number 1 is supported. For input, multiple file volumes are supported if the data set is cataloged.

VOLume

Specifies the volume serial number for allocation of new data sets. If *volume_serial* is not specified, the volume serial number used for allocation is set back to the system default.

The setting for VOLume may be overridden by the SMS storage class, if one is specified. We recommend that VOLume not be coded if an SMS storage class is in use.

volume_serial

The volume serial number for allocation of new data sets.

WRAPrecord

Indicates that data is wrapped to the next record if no new line character is encountered before the logical record length of the receiving file is reached.

Usage

- Because more than one parameter can be specified with the LOCSITE subcommand, *parameter* can be repeated many times, with each *parameter* separated by a blank space.
- Issue the HELP LOCSITE subcommand to display a list of the types of services available on the local host.
- The site-dependent information set with the LOCSITE subcommand remains active until you issue a new LOCSITE subcommand. The new LOCSITE subcommand adds to or changes the parameters established by previous LOCSITE subcommands.
- If you specify one or more incorrect parameters with the LOCSITE subcommand, an error message specifying the incorrect parameter is displayed. All correct parameters are set, regardless of any incorrect parameters, and do not need to be reissued.

Context

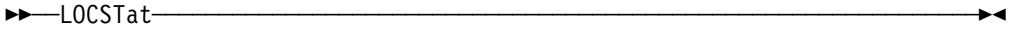
- See “HELP and ? Subcommands—Display Help Information” on page 92 for more information on the HELP subcommand.
- To check the effect of the LOCSITE command on the attributes at the local host, see the FTP LOCSTAT command on page 110.

LOCSTAT Subcommand—Display Local Status Information

Purpose

Use the LOCSTAT subcommand to display local status information.

Format



Parameters

None.

Examples

- Display local status information:

```
locstat
 Trace:FALSE, Send Port:TRUE
 Send Site with Put command:TRUE
 Connected to:Loopback, Port:FTP control (21), logged in
 Local Port:1026
 Data type:a, Transfer mode:s
 Automatic recall of migrated data sets.
 Automatic mount of direct access volumes.
 Data set mode. (Do not treat each qualifier as a directory.)
 Primary allocation 1 track. Secondary allocation 1 track.
 Partitioned data sets will be created with 27 blocks.
 FileType is SEQ (Sequential - the default).
 Number of disk I/O buffers is 3.
 RDW's from VB/VBS files are discarded.
 Retention period is 0.
 DB2 subsystem name is D23
 Volid of Migrated Data Sets is CPDLB2
 Record format VB. Lrecl: 256, Blocksize: 6233
 Data sets will be allocated on WRKLB2.
 Checkpoint interval is 0
 Command:
```

- This example shows the LOCSTAT information display before and after the DEBUG option is turned on. TRACE is changed from FALSE to TRUE when DEBUG was enabled.

```
System:    READY
  User:
             ftp 192.9.2.2
System:
             MVS TCP/IP FTP V3R2
           Connecting to 192.9.2.2, port 21
           220-EZAFTSRV IBM MVS V3R2 at TREEFROG.ABC.OZ, 13:22:57 on 08/02/96
           220 Connection will close if idle for more than 5 minutes.
           USER (identify yourself to the host):
  User:
            rtp
  User:
            >>>USER rtp
           331 Send password please.
           Password:
```

```
            >>>PASS ********
          230 RTP is logged on.
          Command:
  User:
           locstat
  User:
           Trace:FALSE, Send Port:>TRUE
           Send Site with Put command:TRUE
           Connected to:192.9.2.2, Port:FTP control (21), logged in
           Local Port:1045
           Data type:a, Transfer mode:s
           Automatic recall of migrated data sets.
           Automatic mount of direct access volumes.
           Data set mode. (Do not treat each qualifier as a directory.)
           Primary allocation 10 tracks. Secondary allocation 5 tracks.
           Partitioned data sets will be created with 27 blocks.
           FileType is SEQ (Sequential - the default).
           Records in Parallel I/O buffer is 8.
           Record format FB. Lrecl: 80, Blocksize: 3200

           Command:
  User:    debug
System:    Internal debug options enabled
           Command:
  User:    locstat
System:    Trace:TRUE, Send Port:TRUE
           Send Site with Put command:TRUE
           Connected to:192.9.2.2, Port:FTP control (21), logged in
           Local Port:1045
           Data type:a, Transfer mode:s
           Automatic recall of migrated data sets.
           Automatic mount of direct access volumes.
           Data set mode. (Do not treat each qualifier as a directory.)
           Primary allocation 10 tracks.
           Secondary allocation 5 tracks.
           Partitioned data sets will be created with 27 blocks.
           FileType is SEQ (Sequential - the default).
           Records in Parallel I/O buffer is 8.
           Record format FB. Lrecl: 80, Blocksize: 3200
           Command:
  User:    quit
System:
           >>>QUIT
           In SysSendFlush, calling TcpWaitSend with args: 0 000733F8 6
           In SysSendFlush, TcpWaitSend returned: OK
           In SysRead, calling TcpWaitReceive with args: 0 00075474 8192
            In SysRead: Note received:  => TcpId 0 Data delivered 37 bytes Push
           In SysRead, TcpWaitReceive returned: 37
           221 Quit command received. Goodbye.
           GetReply returns 221
           Entering WaitAndClose
           In WaitAndClose: Note received:  => TcpId 0 Connection state changed
           Sending only
           READY
```

Usage

The following status information is displayed:

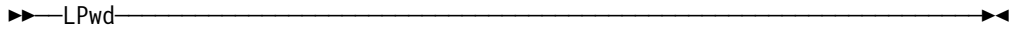
- TRACE setting (true or false)
- SENDSITE setting (true or false)
- SENDPORT setting (true or false)
- Name, port number, and login status of the foreign host
- Port number of the local host
- FTP data type (ASCII, EBCDIC, or Image) and transfer mode (block, stream, or compressed)
- AUTORECALL setting (autorecall or no autorecall)
- AUTOMOUNT setting (automount or no automount)
- DATASETMODE or DIRECTORYMODE
- Primary and secondary space allocation
- Number of PDS directory blocks
- File type (JES, SEQ, or SQL)
- Number of disk I/O buffers used during data transfer
- Storage Management Subsystem (SMS) management class (displayed only if one has been set)
- Volume used when creating new data sets (displayed only if a volume has been specified)
- Unit used when creating new data sets (displayed only if a unit has been specified)
- Reader descriptor words (RDWs) are discarded or kept
- Retention period (in days) for new data sets
- Name of the DB2 subsystem
- Record format, logical record length, and block size, or the data set name when the DCBSN parameter is used
- SMS STORCLASS
- Whether data has been wrapped
- VOLID of migrated data sets

LPWD Subcommand—Display the Current Working-Level Qualifier

Purpose

Use the LPWD subcommand to display the name of the current working-level qualifier on the local host.

Format



Parameters

None.

Examples

Display the name of the current working directory:

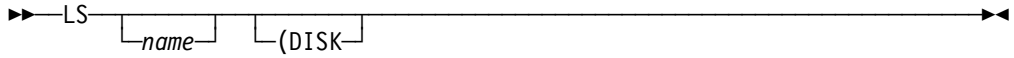
```
lpwd
 Local directory is partitioned data set USER14.FTP.TEST1.
 Command:
```

LS Subcommand—Obtain a List of File Names

Purpose

Use the LS subcommand to list only the names of a set of remote files, file group, or directory.

Format



Parameters

name

Specifies the set of remote files whose names are to be listed. The default is the entire current directory or file group.

DISK

Stores the results of the LS subcommand in the *user_id*.FTP.LSOUTPUT data set. The results are not displayed on the screen.

Examples

The following is a sample response that is displayed as a result of the LS subcommand.

```
>>>PORT 9,67,58,227,4,63
200 Port request OK.
>>>NLST
125 List started OK.
A.X
CHR.TXT
OBEY.TCPIP
PROFILE.EXEC
SPF.ISPPROF
USERTRAN.TCPXLBIN
250 List completed successfully.
Command:
```

The following is a sample entry and response that is displayed as a result of the LS subcommand listing HFS files.

```
cd '/u/user121/ftp.example'

 >>>CWD '/u/user121/ftp.example'
 250 HFS directory /u/user121/ftp.example is the current working directory
 Command:
ls
 >>>PORT 9,67,112,25,4,62
 200 Port request OK.
 >>>NLST
 125 List started OK
 append02
 file1
 file2
 file3
 file4
 file5
 250 List completed successfully.
 Command:
```

Usage

- To make a file group the current working directory, use the CD subcommand. The method you use to specify a directory or file group is host-dependent.
- You can use special characters for pattern matching when specifying the *name*. these characters depend on the host's FTP server.
- If the current local directory is a PDS, then only a member named `LSOUTPUT` is created. If the current local directory is not a PDS, then the local directory, not the user ID, is used as the high-level qualifier for the data set name.

Context

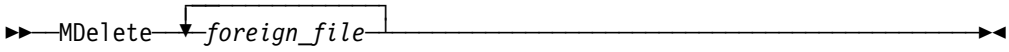
- See Appendix A, Specifying Data Sets and Files, for more information about pattern matching and about specifying data sets and files.
- To make a file group the current working directory, see "CD Subcommand—Change the Directory on the Remote Host" on page 76.
- To get a list of complete directory entries with auxiliary information about the files, see "DIR Subcommand—Obtain a List of Directory Entries" on page 83.
- To change the local direcory, see "LCD Subcommand—Change the Local Working Directory" on page 97.

MDELETE Subcommand—Delete Multiple Files

Purpose

Use the MDELETE subcommand to delete multiple files.

Format



Parameters

foreign_file
Specifies the name of the file to be deleted on the remote host.

Examples

The following is a sample entry and the response that is displayed as a result of the MDELETE subcommand for multiple HFS files.

```
cd '/u/user121/ftp.example'

 >>>CWD '/u/user121/ftp.example'
 250 HFS directory /u/user121/ftp.example is the current working directory
 Command:
 mdelete file1 file2 file3
 >>>PORT 9,67,112,25,4,75
  200 Port request OK.
 >>>NLST file1
 125 List started OK
 250 List completed successfully.
 >>>PORT 9,67,112,25,4,77
 200 Port request OK.
 >>>NLST file2
 125 List started OK
 250 List completed successfully.
 >>>PORT 9,67,112,25,4,76
 200 Port request OK.
 >>>NLST file3
 125 List started OK
 250 List completed successfully.
 >>>DELE file1
 250 /u/user121/ftp.example/file1 deleted.
 >>>DELE file2
 250 /u/user121/ftp.example/file2 deleted.
 >>>DELE file3
 250 /u/user121/ftp.example/file3 deleted.
 Command:
```

Usage

- Because more than one file can be deleted with the MDELETE subcommand, the *foreign_file* parameter of the MDELETE subcommand can be repeated many times, with each *foreign_file* separated by a blank space.
- If you specify one or more incorrect foreign files with the MDELETE subcommand, an error message specifying the incorrect foreign file is displayed. All correct foreign files are deleted, regardless of any incorrect foreign file, and do not need to be reissued.
- HFS file names require special handling for certain special characters. Except for single quote ('), double quote ("), or blank ( ), all special characters that the operating system requires to be preceded by an escape character in commands issued to the shell must be preceded by the backlash (\) escape character. The wildcard characters asterisk (*), question mark (?) and left/right ([]) will be treated as wildcard characters. If these characters appear as actual

characters in a file name, they must be preceded by the escape character to prevent them from being interpreted as wildcard characters.

Context

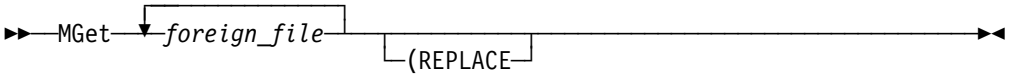
See Appendix A, Specifying Data Sets and Files, for more information about naming conventions.

MGET Subcommand—Copy Multiple Files

Purpose

Use the MGET subcommand to copy multiple files to your local host and create a corresponding number of local files.

Format



Parameters

foreign_file
Specifies the name of the file to be retrieved from the remote host.

Because more than one file can be copied with the MGET subcommand, the *foreign_file* parameter of the MGET subcommand can be repeated many times, with each *foreign_file* separated by a blank space. You can use special characters for pattern matching when specifying the *foreign_file* with the MGET subcommand. These characters are dependent on the foreign host's FTP server.

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 data set already exists and you specify REPLACE, the data in the file is overwritten, but not reallocated. This means the local data set retains its existing characteristics.

Examples

The following is a sample entry and response that is displayed as a result of the MGET subcommand for multiple HFS files.

```
cd '/u/user121/ftp.example'

 >>>CWD '/u/user121/ftp.example'
 250 HFS directory /u/user121/ftp.example is the current working directory
 Command:
 mget file1 file2 file3
 >>>PORT 9,67,112,25,4,90
  200 Port request OK.
 >>>NLST file1
 125 List started OK
 250 List completed successfully.
 >>>PORT 9,67,112,25,4,91
 200 Port request OK.
 >>>NLST file2
 125 List started OK
 250 List completed successfully.
 >>>PORT 9,67,112,25,4,92
 200 Port request OK.
 >>>NLST file3
 125 List started OK
 250 List completed successfully.
 >>>PORT 9,67,112,25,4,93
 200 Port request OK.
 >>>RETR file1
 125 Sending data set /u/user121/ftp.example/file1
 250 Transfer completed successfully.
 3464 Bytes transferred in 1.031 seconds.  Transfer rate 3.36 kbytes/sec.
 >>>PORT 9,67,112,25,4,94
 200 Port request OK.
 >>>RETR file2
 125 Sending data set /u/user121/ftp.example/file2
 250 Transfer completed successfully.
 3993 Bytes transferred in 0.923 seconds.  Transfer rate 4.33 kbytes/sec.
 >>>PORT 9,67,112,25,4,95
 200 Port request OK.
 >>>RETR file3
 125 Sending data set /u/user121/ftp.example/file3
 250 Transfer completed successfully.
 3993 Bytes transferred in 0.791 seconds.  Transfer rate 5.05 kbytes/sec.
 Command:
```

Usage

- **Attention**

  When you use the MGET subcommand, FTP may truncate data records and you might lose data.

  – When you are creating a new data set at the client and the value of LRECL, as shown by the LOCSTAT command, is a value less than the LRECL of a received data set, FTP truncates the received data set.

  – If the data set name already exists at the client and the logical record length (LRECL) of the data set at the client is less than the LRECL of the transmitted data set, FTP truncates the transmitted data set.

    You can encounter this situation when you use MGET with the REPLACE option.

- If the name specified for *foreign_file* is not acceptable to your local host, the file is not transferred. To get a file from the remote host, you must have a defined working directory on that host, and you must have read privileges to the files in this working directory.
- If you specify one or more incorrect foreign files with the MGET subcommand, an error message specifying theincorrect foreign file is displayed. All correct

foreign files are retrieved, regardless of any incorrect foreign files, and do not need to be reissued.

- HFS file names require special handling for certain special characters. Except for single quote ('), double quote ("), or blank ( ), all special characters that the operating system requires to be preceded by an escape character in commands issued to the shell must be preceded by the backlash (\) escape character. The wildcard characters asterisk (*), question mark (?) and left/right ([]) will be treated as wildcard characters. If these characters appear as actual characters in a file name, they must be preceded by the escape character to prevent them from being interpreted as wildcard characters.

Context

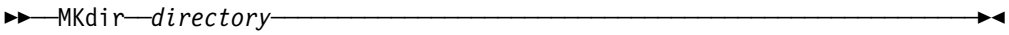
- See "CD Subcommand—Change the Directory on the Remote Host" on page 76 and "ACCOUNT Subcommand—Supply Account Information" on page 71, for more information about working directories.
- See Appendix A, Specifying Data Sets and Files, for more information about naming conventions.

MKDIR Subcommand—Create a Directory on the Remote Host

Purpose

Use the MKDIR subcommand to create a directory on the remote host,

Format



Parameters

directory
Specifies the name of the directory to be created.

Examples

Create a directory on the remote host (1.1.2.3 in this example). Both `EXAMPLE` and `FTP.EXAMPLE` are created in the remote host, showing the difference between specifying and omitting quotation marks in the directory name.

```
System:    READY
  User:
           ftp 9.67.113.24.621
System:
           MVS TCP/IP FTP V3R2
            FTP.DATA FILE NOT FOUND.  USING HARDCODED DEFAULT VALUES.
            Connecting to 1.1.2.3, port 21
            220-EZAFTSRV IBM MVS V3R2 at EMU.ABC.OZ, 15:34:32 on 08/03/93
            220 Connection will not timeout.
            USER (identify yourself to the host):
```

```
  User:    user121

System:    >>>USER user121
            331 Send password please.
            Password:

              >>>PASS ********
            230 USER121 is logged on.  Working directory is '/u/user121".
            Command:

  User:   dir
System:
            >>>PORT 9,67,112,25,4,96
           200 Port request OK.
           >>>LIST
           125 List started OK.
            total 2768
          -rwxr-xr-t    2  USER121  SYS1     389120  Feb  5  16:03  ftpdka
          -rwxr-xr-t    2  USER121  SYS1     962560  Feb  5  16:04  ftpsrvka
          -rw-r-----    1  USER121  SYS1      11648  Jan 20  14:30  g.s
          drwxr-x---    3  USER121  SYS1          0  Oct 21  17:50  msg
          -rw-r-----    1  USER121  SYS1       1458  Jan 10  19:25  s.k
          drwxr-x---    2  USER121  SYS1          0  Feb  6  15:59  tcpip
          drwxr-x---    2  USER121  SYS1          0  Feb  6  17:29  test
            250 List completed successfully.
            Command:
  User:
           mkdir example
System:
           >>>MKD example
           257 "/u/user121/example" created.
           Command:
  User:   dir
System:
            >>>PORT 9,67,112,25,4,97
           200 Port request OK.
           >>>LIST
           125 List started OK.
            total 2768
          drwxr-xr--    2  USER121  SYS1          0  Feb  7  19:57  example
          -rwxr-xr-t    2  USER121  SYS1     389120  Feb  5  16:03  ftpdka
          -rwxr-xr-t    2  USER121  SYS1     962560  Feb  5  16:04  ftpsrvka
          -rw-r-----    1  USER121  SYS1      11648  Jan 20  14:30  g.s
          drwxr-x---    3  USER121  SYS1          0  Oct 21  17:50  msg
          -rw-r-----    1  USER121  SYS1       1458  Jan 10  19:25  s.k
          drwxr-x---    2  USER121  SYS1          0  Feb  6  15:59  tcpip
          drwxr-x---    2  USER121  SYS1          0  Feb  6  17:29  test
           250 List completed successfully.
           Command:

  User:    mkdir '/u/user121/ftp.example'

System:     >>>MKD '/u/user121/ftp.example'
           257 "/u/user121/ftp.example" created.
           Command:

  User:   dir
```

```
System:    >>>PORT 9,67,112,25,4,98
           200 Port request OK.
           >>>LIST
           125 List started OK.
          total 2800
          drwxr-x---    2  USER121  SYS1          0  Feb  7  19:57  example
          drwxr-x---    2  USER121  SYS1          0  Feb  7  19:57  ftp.example
          -rwxr-xr-t    2  USER121  SYS1     389120  Feb  5  16:03  ftpdka
          -rwxr-xr-t    2  USER121  SYS1     962560  Feb  5  16:04  ftpsrvka
          -rw-r-----    1  USER121  SYS1      11648  Jan 20  14:30  g.s
          drwxr-x---    3  USER121  SYS1          0  Oct 21  17:50  msg
          -rw-r-----    1  USER121  SYS1       1458  Jan 10  19:25  s.k
          drwxr-x---    2  USER121  SYS1          0  Feb  6  15:59  tcpip
          drwxr-x---    2  USER121  SYS1          0  Feb  6  17:29  test
           250 List completed successfully.
          Command:
```

Usage

- The MKDIR subcommand sends a request to the remote host's FTP server to create a directory with name *directory* in the current working directory. When the request is sent to a remote MVS server, a PDS is created with either a fully qualified name (if *directory* is specified in single quotation marks), or FTP appends *directory* to the current working directory.
- For an MVS server, the data set characteristics used to allocate the new PDS are determined by the settings of the SITE parameters of the server.

Context

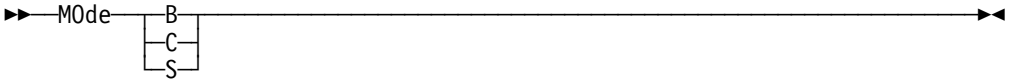
- See "SITE Subcommand—Send Site Specific Information to a Host" on page 132 for information on setting the SITE parameters.
- See "STATUS Subcommand—Retrieve Status Information from a Remote Host" on page 147 for information on displaying the SITE parameters.

MODE Subcommand—Set the Data Transfer Mode

Purpose

Use the MODE subcommand to define how bits of data are to be transmitted.

Format



Parameters

B Sets the block mode. In block mode, data is transmitted as a series of data blocks, preceded by one or more header bytes. Block mode preserves the logical record boundaries of the data set or file. When MODE is set to B, the data transfer type must be EBCDIC.

C Sets the compressed mode. In compressed mode, data is transmitted as a series of data blocks, preceded by one or more header bytes. Compressed mode preserves the logical record boundaries of the data set or file. In compressed mode, data is transmitted without repetitive characters and blanks. When MODE is set to C, the data transfer type must be EBCDIC.

Note: Since additional processing time is required for both the sender and receiver to compress or decompress the data, you need to evaluate the tradeoffs before you compress a file.

S Sets the stream mode. In stream mode, data is transmitted as a stream of bytes. Any data transfer type can be used with stream mode. Stream mode is more efficient, because data block information is not transferred.

Usage

- The receiving host must support the compressed data mode.
- Data compression increases CPU processing costs even if the amount of data transferred is not large.

Context

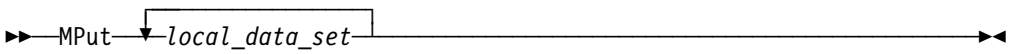
For more information about transfer methods, see Table 9 on page 25.

MPUT Subcommand—Copy Multiple Data Sets to the Remote Host

Purpose

Use the MPUT subcommand to copy multiple data sets from your local host to the remote host.

Format



Parameters

local_data_set
Specifies the name of the sequential data set or PDS member on your local host being sent to the remote host.

Because more than one data set can be copied with the MPUT subcommand, the *local_data_set* parameter of the this subcommand can be repeated many times, with each *local_data_set* separated by a blank space. You can use the asterisk (*) character for pattern matching when specifying the *local_data_set* with the MPUT subcommand.

Usage

- **Attention**

  FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the MPUT subcommand, FTP may truncate data records and you might lose data.

  - When you are creating a new file at the server and the value of LRECL, as shown by the STATus subcommand, is a value less than the LRECL of the transmitted data set and SENDSITE has been set to OFF, FTP truncates the transmitted data set.
  - If the data set name already exists at the receiving site and the logical record length (LRECL) of the data set at the receiving site is less than the LRECL of the transmitted data set, FTP truncates the transmitted data set.

- By default, if you use the MPUT subcommand, the remote host creates files with the same names as those specified in *local_data_set* and overwrites any existing files with those names.

  To put files on the remote host with unique file names, you must have set unique storage on before issuing the MPUT command. Use the SUNIQUE subcommand to change the storage method.

- To send a data set to the remote host, you must have a defined working directory on the remote host and write privileges to the files in this working directory.
- If you specify one or more incorrect parameters with the MPUT subcommand, an error message specifying the incorrect parameter is displayed. All correct files are transferred, regardless of any incorrect parameters, and do not need to be reissued.

Context

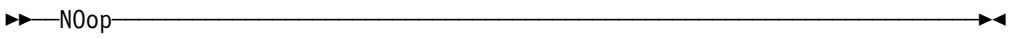
- See "SUNIQUE Subcommand—Toggle the Storage Method" on page 165 for information about changing the storage method on the remote host.
- See Appendix A, Specifying Data Sets and Files, for more information about naming conventions.

NOOP Subcommand—Test the Connection

Purpose

Use the NOOP subcommand to determine if the foreign host is still responding.

Format



Parameters

None.

Examples

- If the foreign host is responding, you will receive one of the following responses:

  ```
  200 OK
  ```

  or

  ```
  200 NOOP command successful
  ```

- If the foreign host does not respond or is not connected, you receive an appropriate error message, such as:

  ```
  421 no connection to foreign server
  ```

Usage

You can use the NOOP command to keep a connection alive that would otherwise be disconnected if it were idle for longer than the system time-out period.

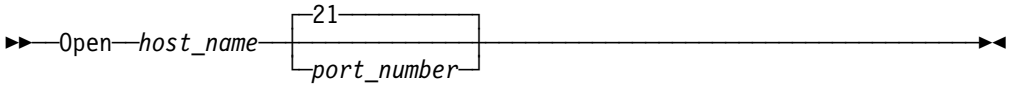
OPEN Subcommand—Connect to the FTP Server

Purpose

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

- If, after closing a connection, you want to open another connection without leaving the FTP environment.
- If you were unable to open a connection when you specified a *foreign_host* with the FTP command.

Format



Parameters

host_name
Specifies the host name or IP address of the foreign host.

port_number
Identifies a port on the foreign host. The default is well-known port 21.

Usage

If you are already connected to a host, you must disconnect from the host before you can connect to a different host with the OPEN subcommand.

Context

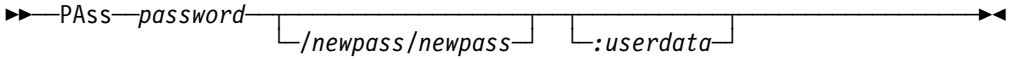
See "CLOSE Subcommand—Disconnect from a Remote Host" on page 80 for more information about closing a connection.

PASS Subcommand—Supply a Password

Purpose

Use the PASS subcommand to supply a password to a host.

Format



Parameters

password
: Specifies your password on the remote host.

/newpass/newpass
: Resets a password.

:userdata
: An optional character string to be passed to the FTCHKPWD user exit routine. The userdata must be separated from the password information by a colon (:) and may be any combination of up to 200 non-blank characters and numbers except the colon. Care should be taken when using the backslash character (\) in combination with other characters which might be interpreted as an escape sequence by the C compiler.

Usage

- The PASS subcommand must be preceded by the USER subcommand. For some sites, the password completes your identification for access control on the remote host.
- On MVS hosts, a PASS command is automatically generated by the USER command. If you do not enter the password as part of the user command, you will be prompted for it.

Context

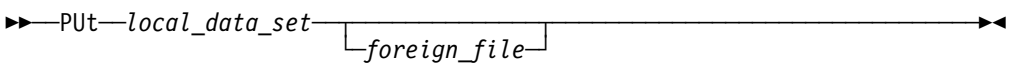
"USER Subcommand—Identify Yourself to a Host or Change Your TSO User ID Password" on page 172)

PUT Subcommand—Copy Data Sets to the Remote Host

Purpose

Use the PUT subcommand to copy a data set from your local host to the remote host.

Format



Parameters

local_data_set
Specifies the name of the sequential data set or PDS member on your local host being sent to the remote host.

foreign_file
Specifies the name that the delivered data set is given on the remote host. If the *foreign_file* name is not specified, the *foreign_file* name is the same as the *local_data_set* name.

Usage

- **Attention**

  FTP maintains the attributes of a data set that is transmitted between a client and a server. However, when you use the PUT subcommand, FTP may truncate data records and you might lose data.

  – When you are creating a new file at the server and the value of LRECL, as shown by the STATus subcommand, is a value less than the LRECL of the transmitted data set and SENDSITE subcommand has been set to OFF, FTP truncates the transmitted data set.

  – If the data set name already exists at the receiving site and the logical record length (LRECL) of the data set at the receiving site is less than the LRECL of the transmitted data set, FTP truncates the transmitted data set.

- If the remote host already has a file with the name specified by *foreign_file*, the remote host overwrites the existing file. If the remote host does not have a file with the same name specified by *foreign_file*, the remote host creates a new file.
- To put files on the remote host with unique file names, you must have set unique storage on before issuing the PUT command. Use the SUNIQUE subcommand to change the storage method.
- To send a data set to the remote host, you must have a defined working directory on the remote host and write privileges to the files in this working directory.

Context

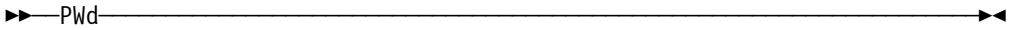
- See “SUNIQUE Subcommand—Toggle the Storage Method” on page 165 for information about changing the storage method on the remote host.
- See Appendix A, Specifying Data Sets and Files, for more information about naming conventions.

PWD Subcommand—Display the Current Working Directory

Purpose

Use the PWD subcommand to display the name of the current working directory on the remote host.

Format



Parameters

None.

Examples

Display the name of the current working directory:

```
pwd
 >>>PWD
 257 "'USER17.HSMTEST.'" is working directory
 Command:
```

Display the name of the current HFS working directory:

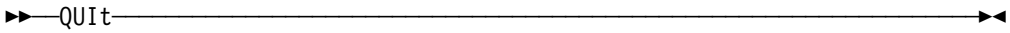
```
pwd
 >>>PWD
 257 "/u/user121/example" is the HFS working directory.
 Command:
```

QUIT Subcommand—Leave the FTP Environment

Purpose

Use the QUIT subcommand to disconnect from the foreign host and end the FTP session.

Format



Parameters

None.

Usage

The QUIT subcommand ends the FTP session with the remote host and exits FTP on the local host. To establish a new session, use the FTP command.

Context

See “FTP Command—Enter the FTP Environment” on page 87 for information about the FTP command.

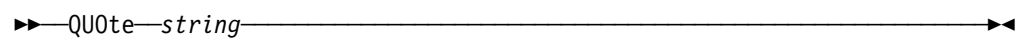
QUOTE Subcommand—Send an Uninterpreted String of Data

Purpose

Use the QUOTE subcommand to send an uninterpreted string of data to the server port on the foreign host.

The QUOTE subcommand bypasses the FTP interface of your local host. You can use the QUOTE subcommand to send commands that the remote server understands, but that the local host does not understand.

Format



Parameters

string
Specifies the data to be sent verbatim to the remote host's FTP server.

Examples

- For example, `QUOTE TYPE B 1` causes the FTP server to change its transfer type to Shift JIS Kanji, without changing the transfer type in the FTP client. The client in this example should be set to the ASCII transfer type before the QUOTE subcommand is issued.
- The following examples show the screen display when setting the DBCS transfer type to JIS78KJ, shift-in JISROMAN, and then setting it to HANGEUL using EBCDIC SO/SI characters. The examples show an MVS TCP/IP FTP client connected to an MVS TCP/IP FTP server. All three methods of setting the DBCS transfer type are shown.

```
  User:   jis78kj (jisroman
System:   >>>TYPE b 4 r
           200-Representation type is KANJI JIS 1978 shift-in JISROMAN
           200 Standard DBCS control used
           Command:
  User:   type b 4 r
System:   >>>TYPE b 4 r
           200-Representation type is KANJI JIS 1978 shift-in JISROMAN
           200 Standard DBCS control used
           Command:

  User:   jis78kj (jisroman notype

System:   Command:

  User:   quote type b 4 r

System:   >>>type b 4 r
           200-Representation type is KANJI JIS 1978 shift-in JISROMAN
           200 Standard DBCS control used
           Command:
```

```
  User:
           hangeul (sosi ebcdic
System:   >>>TYPE b 5 s e
          200-Representation type is Hangeul
          200-SO/SI characters X'0E'/X'0F' used
          200 Data transfer is mixed SBCS/DBCS
          Command:
  User:   type b 5 s e
System:   >>>TYPE b 5 s e
           200-Representation type is Hangeul
           200-SO/SI characters X'0E'/X'0F' used
           200 Data transfer is mixed SBCS/DBCS
           Command:
  User:   hangeul (sosi ebcdic notype
System:   Command:
  User:   quote type b 5 s e
System:   >>>type b 5 s e
          200-Representation type is Hangeul
          200-SO/SI characters X'0E'/X'0F' used
          200 Data transfer is mixed SBCS/DBCS
          Command:
```

Usage

- **Attention**

  No parsing or validity checking is performed on the character string you enter by FTP on your local host. If the character string you send to the FTP server is part of a required sequence of commands, you are required to provide this sequence correctly, or the results might be unpredictable.

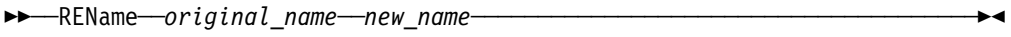
- The QUOTE subcommand can be used to generate any of the DBCS TYPE commands supported by the server. This subcommand is used when the FTP server supports the DBCS TYPE command, but the FTP client does not.

RENAME Subcommand—Rename Files

Purpose

Use the RENAME subcommand to rename a file on the remote host.

Format



Parameters

original_name
Specifies the current name of the file.

new_name
Specifies the new name of the file.

Usage

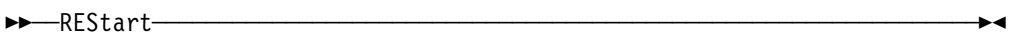
- For MVS data sets, if the file specified by the new_name already exists, the existing file is not replaced by the new file.
- For HFS files, if the file specified by new_name already exists, the existing file will be replaced.

RESTART Subcommand—Restart a Checkpointed Data Transfer

Purpose

Use the RESTART subcommand to restart a checkpointed data transfer.

Format



Parameters

None.

Usage

- The RESTART subcommand restarts the last checkpointed file transfer request at the point of the last valid checkpoint stored in the checkpoint data set, *userid*.FTP.CHKPOINT, for the TSO user *userid*. The file transfer environment, such as file transfer mode and type, which was in effect at the time of the last checkpointed file transfer request must be re-established before issuing the RESTART subcommand.
- The RESTART subcommand should be used when a checkpointed file transfer request fails because of some temporary condition such as the loss of the connection between the client and the server.

Context

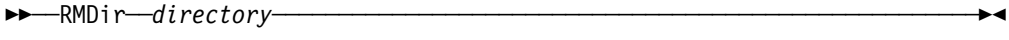
For more information about checkpointing a file transfer request, see the CHKPTint parameter of the SITE subcommand on page 135 or the CHKPTint parameter of the LOCSITE command on page 102.

RMDIR Subcommand—Remove a Directory on the Remote Host

Purpose

Use the RMDIR subcommand to remove a directory on the remote host.

Format



Parameters

directory
Specifies the name of the directory to be removed.

Usage

- The RMDIR subcommand sends a request to the remote host's FTP server to remove a directory with name *directory* from the current remote directory.
- RMDIR subcommand can be used to delete a PDS.

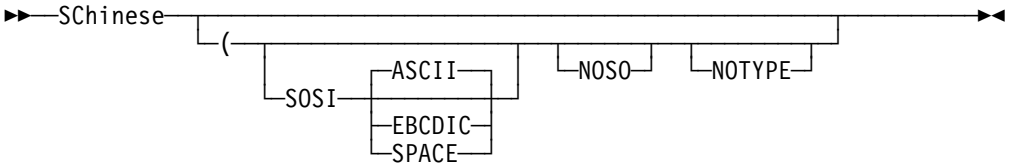
SCHINESE Subcommand—Change the Data Transfer Type to SCHINESE

Purpose

Use the SCHINESE subcommand to change the data transfer type to Simplified Chinese.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see "FTP Command—Enter the FTP Environment" on page 87 and "Changing Local Site Defaults Using FTP.DATA" on page 33.

Format



Parameters

SOSI
Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC, or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII
When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The SCHINESE client subcommand is equivalent to the TYPE B 9 server command.

Context

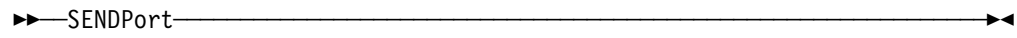
See “FTP with DBCS Support” on page 66 for more information.

SENDPORT Subcommand—Toggle the Sending of Port Information

Purpose

Use the SENDPORT subcommand to toggle the automatic sending of the PORT command.

Format



Parameters

None.

Usage

- By default, the SENDPORT subcommand is turned on when you start an FTP session. Each time you use the SENDPORT subcommand, it is turned alternately on and off.
- FTP uses a PORT command, by default, when establishing a connection for each data transfer. FTP does not send PORT commands for data transfer when you disable PORT commands by toggling the function off.
- SENDPORT is useful for communication with those FTP implementations that ignore PORT commands, but show (incorrectly) that the PORT command has been accepted.
- To determine if the sending of port information is enabled or disabled on your local host, use the LOCSTAT subcommand.

Context

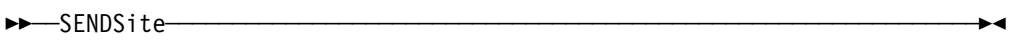
See “LOCSTAT Subcommand—Display Local Status Information” on page 110 for more information about LOCSTAT subcommand.

SENDSITE Subcommand—Toggle the Sending of Site Information

Purpose

Use the SENDSITE subcommand to toggle the automatic sending of the SITE subcommands when sending a data set to a foreign host.

Format



Parameters

None.

Usage

- By default, the SENDSITE subcommand is turned on when you start an FTP session. Each time you use the SENDSITE subcommand, it is turned alternately on and off.

  When turned on, FTP sends a SITE subcommand containing record format information for the file or data set, when you issue the PUT or MPUT subcommand.
- SENDSITE is useful when you want to PUT a file to the remote host and have the file created with the same characteristics as defined at the local host.
- If you are using either an SMS dataclass or a model DCB at your MVS server to provide the lrecl or recfm, you must toggle the SENDSITE setting off at the client. Otherwise, the SITE information that is sent automatically by the client will override the values provided by the SMS dataclass or model DCB.
- To determine if the sending of site information is enabled or disabled on your local host, use the LOCSTAT subcommand.

Context

See “LOCSTAT Subcommand—Display Local Status Information” on page 110) for information about the LOCSTAT subcommand.

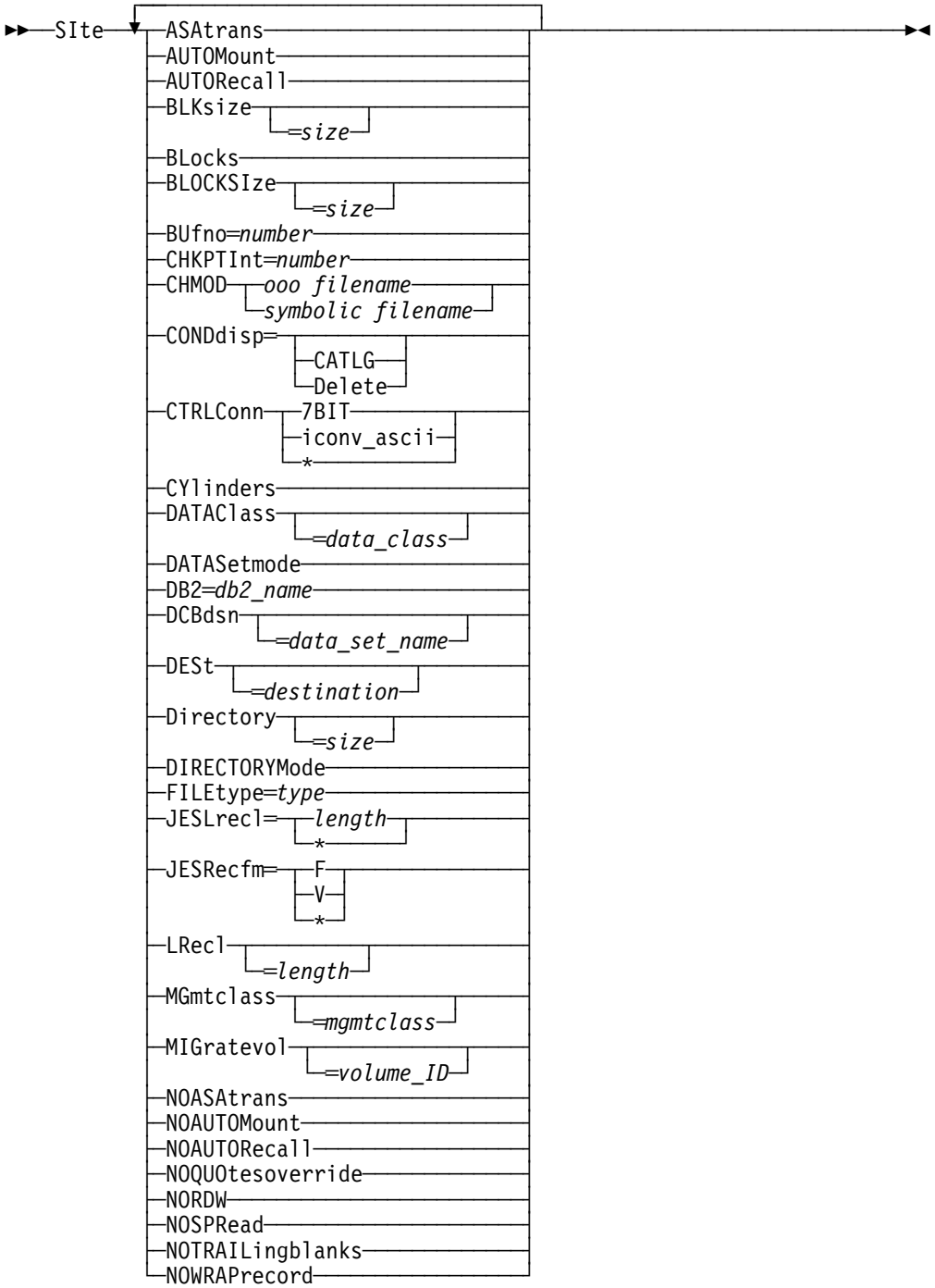
SITE Subcommand—Send Site Specific Information to a Host

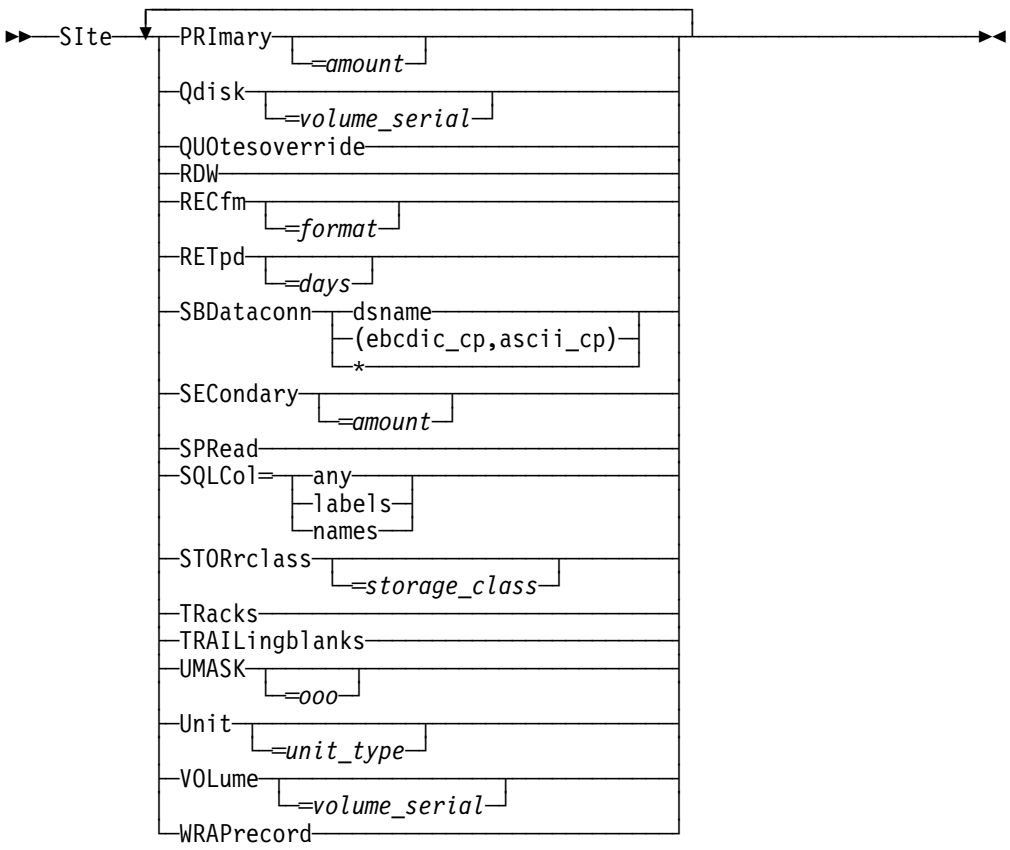
Purpose

Use the SITE subcommand to send information that is used by the remote host to provide services specific to that host system.

The parameters used by the SITE command are dependent on the server. The explanation supplied in this section applies to the MVS OE FTP server.

Format


SIte
ASAtrans
AUTOMount
AUTORecall
BLKsize
=size
BLocks
BLOCKSIze
=size
BUfno=number
CHKPTInt=number
CHMOD
ooo filename
symbolic filename
CONDdisp=
CATLG
Delete
CTRLConn
7BIT
iconv_ascii
*
CYlinders
DATAClass
=data_class
DATASetmode
DB2=db2_name
DCBdsn
=data_set_name
DESt
=destination
Directory
=size
DIRECTORYMode
FILEtype=type
JESLrecl=
length
*
JESRecfm=
F
V
*
LRecl
=length
MGmtclass
=mgmtclass
MIGratevol
=volume_ID
NOASAtrans
NOAUTOMount
NOAUTORecall
NOQUOtesoverride
NORDW
NOSPRead
NOTRAILingblanks
NOWRAPrecord



Parameters

ASAtrans

Permits the C-FTP server to interpret the characters in the first column of ASA files being transferred as print control characters, if recognized as such. ASAtrans is valid only for C/370 2.2 and LE/370 1.3 runtime libraries.

AUTOMount

Permits automatic mounting of volumes for data sets on volumes that are not mounted. If AUTOMount is specified and an unmounted volume is needed, a message is automatically issued to the MVS operator console requesting that the volume be mounted. The MVS operator must then mount the volume and reply to the message before FTP can proceed.

AUTORecall

Permits automatic recall of migrated data sets.

BLKsize

Specifies the block size of a newly allocated data set. BLKSIZE is functionally equivalent to BLOCKSIZE. The BLOCKSIZE parameter is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS.

When specified without a *size* or with BLKSIZE=0, no block size will be used when allocating the new data set. When specified without a size, the equal sign (=) is optional.

Specify BLKSIZE with no value or BLKSIZE=0 if you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the

BLKSIZE value, or if you are specifying DCBDSN=*data_set_name* and you want to use the blocksize from the DCBDSN data set. If BLKSIZE=*size* is specified as non-zero and with either the DATACLASS or DCBDSN parameter, the value specified by the SITE BLKSIZE parameter overrides the DATACLASS or DCBDSN blocksize.

size
Specifies the block size of a newly allocated data set. The valid range is 0 through 32 760.

BLocks
Specifies that primary and secondary space allocations are in blocks.

If both PRIMARY and SECONDARY are unspecified, and an SMS data class has been specified, the space allocation is determined by the SMS data class and the BLOCKS parameter is ignored.

BLOCKSIze
Specifies the block size of a newly allocated data set. BLOCKSIZE is functionally equivalent to BLKSIZE. BLOCKSIZE is obsolete but it is accepted to provide compatibility with previous releases of TCP/IP for MVS. See the BLKSIZE parameter for more information.

BUfno
Specifies the number of access method buffers that are used when data is read from or written to a data set. The valid range is 1 through 255. The default value is 5.

CHKPTInt
Specifies the checkpoint interval for the sending site in a file transfer request. If the checkpoint interval is zero, no checkpointing occurs and no marker blocks are transmitted. The default value is zero.

Use SITE to set CHKPTInt when the FTP server is the sending site (GET command). It will not be useful to change the checkpoint interval to more than zero in a GET unless the receiving site is able to process checkpoint markers in the file transfer data stream.

CHKPTInt is valid only for block and compressed modes of data transfer.

number
The checkpoint interval for the sending site in a file transfer request. This value is used to determine when checkpoint marker blocks are to be transmitted so that transmission can be restarted based on the information in the last marker.

A large checkpoint interval means that a large amount of data is sent in between markers and therefore few markers are sent. A smaller checkpoint interval means that less data is sent in between markers and therefore more markers are sent.

The costs involved with using a nonzero checkpoint interval are:

- The markers themselves are transmitted, which means more bytes being sent across the network 44 bytes per marker).
- Additional packets and acknowledgements may be required. For example, when the MVS client PUTs a file, a reply packet is sent from the server to the client and then the client must acknowledge that packet.

To use this formula to estimate the appropriate checkpoint interval, you need to know the record length of the file you are transferring and how much data you think can always be transmitted reliably.

```
           amount of data in interval
CHKPTINT = --------------------------
           record length of the file
```

We recommend that you do not checkpoint more often than once every 200KB. Therefore, if the file we are transferring has 80-byte records:

```
CHKPTINT = 200KB / 80 bytes
         = 200 * 1024 bytes / 80 bytes
         = 2560
```

CHMOD

Changes the permission bits for a file. The format of the CHMOD keyword will be:

- **CHMOD** *ooo filename* where *ooo* is the octal mask and *filename* is the name of the file whose mask is to be changed. If filename does not begin with a slash (/), it will be appended to the current working directory. If filename begins with a slash (/), it will be interpreted as a complete directory name.
- **CHMOD** *symbolic filename* where *symbolic* is a value in the format accepted by the OE CHMOD command (for example, a+r).

  Note: Only r, w, and x permission bits are accepted in *symbolic*.

  The *filename* specified must be an HFS filename. The setting of QUOtesoverride is ignored and all quotes are treated as part of the filename.

The CHMOD parameter must be the only or last keyword on a SITE subcommand.

CONDdisp

Specifies the disposition of the data set if a store operation for a new data set ends before all of the data is written.

Catlg

Specifies that a data set is kept and catalogued when an FTP file transfer ends prematurely.

Delete

Specifies that a data set is deleted when an FTP file transfer ends prematurely.

Delete is ignored if the file transfer failed as a result of the FTP server being terminated or if the server has received checkpoint information during data transfer.

CTRLCONN

Specifies the ASCII code page to be used for the control connection.

```
SITE CTRLConn=7BIT
SITE CTRLConn=iconv_ascii
SITE CTRLConn=*
```

where

7BIT
Indicates 7-bit ASCII is to be used

iconv_ascii
Indicates the name of an ASCII code page recognized by iconv. For a list of code pages supported by iconv, see code set converters in an appendix in the *OS/390 C/C++ Programming Guide.*

***** Indicates ASCII translate tables set up at initialization are to be used.

CYlinders
Specifies that primary and secondary space allocations are in cylinders.

PRIMARY and SECONDARY are unspecified, and an SMS data class has been specified, the space allocation will be determined by the SMS data class and the CYlinders parameter will be ignored.

DATAClass
Specifies the SMS data class as defined by your organization for the target host. Specifying DATAClass with no parameter value cancels the dataclass specification. The equal sign (=) is optional in this case.

See "Specifying Values for New Data Sets" on page 39 for more information about specifying attributes when allocating new data sets.

data_class
Specifies the SMS data class as defined by your organization for the target host. If values are specified for any of the following SITE parameters, the values specified by the SITE parameter will override the value specified in the SMS dataclass:

- BLKSIZE
- DIRECTORY
- LRECL
- PRIMARY
- RECFM
- RETPD
- SECONDARY

If the SITE DCBDSN parameter is specified, the LRECL, RECFM, BLKSIZE, and RETPD (if specified) of the DCBDSN data set will override the values specified in the dataclass.

If the SITE MGMTCLASS parameter is specified, and the requested management class specifies a retention period, the retpd value of the management class may override the retpd value of the dataclass.

DATASetmode
Specifies that all the data set qualifiers below the current directory are treated as entries in the directory (disables DIRECTORYMode).

DB2
Specifies the name of the DB2 subsystem.

db2_name
The name of the DB2 subsystem.

DCBdsn

Specifies the name of the data set to be used as a model for allocation of new data sets. Specifying DCBdsn with no parameter value cancels the DCBdsn specification.

data_set_name

Specifies the name of the data set. The file name must be an MVS data set name. HFS file names are not allowed on the DCBDSN parameter. The setting of QUOtesoverride is ignored. If the file name is enclosed in single quotes, it will override the current working directory; otherwise it will be appended to the current working directory.

Notes:

1. Specify SITE RECFM, LRECL, and BLKSIZE parameters with no values to allow characteristics from the model DCB to be used.
2. To override the model characteristics of RECFM, LRECL, BLKSIZE, or RETPD, specify a value on the SITE command.
3. Ensure that SENDSITE subcommand is toggled off. Otherwise, the SITE information that is sent automatically by the client will override the values provided by the model DCB.
4. If MGMTCLASS is specified, the RETPD value of the MGMTCLASS may override the RETPD value.

Specifying a GDG data set with a relative index produces an error message. The following examples are unsupported specifications:

```
SITE DCBDSN=MYGDG(0)
SITE DCBDSN=MYGDG(–nnn)or
SITE DCBDSN=MYGDG(+nnn)
```

DESt

Indicates the NJE destination to which the files are routed when you enter a PUT command. If specified without a *destination*, the destination will reset and files will be stored at the host system rather than sent to a remote network. the disposition of the data set if a store operation for a new data set ends before all of the data is written.

The SITE DEST subcommand allows you to send data sets to other users on machines that are connected on a Network Job Entry (NJE) network rather than storing them at the server.

destination

Indicates the NJE destination to which the files are routed when you enter a PUT command. The value specified for destination may be:

- userID@nodeID
- nodeID.userID
- nodeID
- DestID

The file will be sent over the NJE network to the specified destination.

This parameter is ignored if FILETYPE=JES is set.

DIrectory
Specifies the number of directory blocks to be allocated for the directory of a PDS. without the *size*, no directory value will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify DIRECTORY without a *size* when you are also specifying DATACLASS=dataclass and you want the SMS dataclass to provide the DIRECTORY *size*. If DIRECTORY=*size* is specified with DATACLASS, the value specified by the SITE DIRECTORY parameter will override the DATACLASS directory specification.

size
Specifies the number of directory blocks to be allocated for the directory of a PDS. The valid range is 1 through 16 777 215.

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 data set qualifier is the only one used by the MGET, LS, and DIR subcommands.

FILEtype
Specifies the file type of the data set.

type
The file type of the data set can be:

Type	Description
SEQ	Sequential or partitioned data sets
SQL	SQL query function
JES	Remote job submission

JESLrecl
Specifies the logical record length for the JES internal reader at the foreign host.

length
The logical record length for the JES internal reader at the foreign host. The valid range is 1 through 254.

*****
Indicates that the logical record length should be taken from the site LRECL parameter setting.

JESRecfm
Specifies the record format for the JES internal reader at the foreign host.

F
Fixed record format.

V
Variable record format.

*****
Indicates that the record format should be taken from the SITE RECFM parameter setting.

LRecl

Specify the logical record length of a newly allocated data set. When specified without a *length* for the C server, no lrecl will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify LRECL with no value when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the LRECL value, or when you are specifying DCBDSN=*data_set_name* and you want to use the LRECL from the dcbdsn data set. If LRECL=*length* is specified with either DATACLASS or DCBDSN, the length specified by the SITE LRECL parameter will override the DATACLASS or DCBDSN LRECL.

length

Specifies the logical record length of a newly allocated data set. The valid range is 0 through 32 760.

MGmtclass

Specifies the SMS management class as defined by your organization for the target host. Specifying MGmtclass with no *mgmtclass* cancels the mgmtclass specification. The equal sign (=) is optional in this case.

mgmtclass

Specifies the SMS management class as defined by your organization for the target host. If the mgmtclass specified has a setting for RETPD, the value specified by the mgmtclass may override the setting of the RETPD site parameter, the RETPD value of a model data set if the DCBDSN parameter is specified, and the RETPD value defined in an SMS data class if DATACLASS is specified. See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.

MIGratevol

Specifies the volume ID for migrated data sets if they do not use IBM storage management systems. If you do not specify MIGratevol, the default volume_serial is MIGRAT.

volume_ID

The volume ID for migrated data.

NOASAtrans

Treats ASA file transfers as regular file transfers; that is, the ASA characters are not converted to C control character sequences.

NOAUTOMount

Prevents automatic mounting of volumes for data sets on volumes that are not mounted.

NOAUTORecall

Prevents automatic recall of migrated data sets.

Note: A migrated data set can be deleted even though NOAUTORECALL is specified, because migrated data sets are not recalled for deletion.

NOQUOTESOVERRIDE

Treats a single quote appearing at the beginning of the file name, as well as all other single quotes contained in the file name, as part of the actual file name. The entire file name, including the leading single quote, will be appended to the current working directory.

NORDW
Specifies that Variable Record Descriptors (RDW) are not treated as if they were part of the record and are discarded during FTP transmission of variable format data sets.

NOSPRead
Specifies that the output is in report format rather than spreadsheet format when the file type is SQL.

NOTRAILingblanks
Specifies that the FTP server does not preserve the trailing blanks that are in a fixed format data set when the data is retrieved.

NOWRAPrecord
Indicates that data is truncated if no new line character is encountered before the logical record length of the receiving file is reached. NOWRAPrecord is the default.

PRImary
Specifies the amount of tracks, blocks, or cylinders for primary allocation. When specified without an *amount*, no primary value will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify PRIMARY with no value when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the PRIMARY *amount*

To allow the SMS data class to determine the space allocation, both PRIMARY and SECONDARY must be specified with no value. The tracks/blocks/cylinders setting will be ignored in this case. If PRIMARY=*amount* is specified with DATACLASS, the value specified by the SITE PRIMARY parameter will override the DATACLASS space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for primary allocation. For allocating partitioned data sets, this is the amount that will be allocated for the primary extent.

For allocating sequential data sets this is the maximum amount that will be allocated for the primary extent. If a lesser amount is needed to hold the data being transferred, only the amount actually needed to hold the data will be allocated. The valid range is 1 through 16 777 215.

Qdisk
Displays statistics about available space on a specific volume. If Qdisk is entered without a specific *volume serial*, statistics about available space are displayed for each volume that is defined with “Use Attribute=storage”.

volume_serial
Displays statistics about available space on a specific volume.

QUOTESOVERRIDE
Indicates the usage of single quotes appearing at the beginning and the end of the file name are interpreted to mean that the file name contained in the single quotes should override the current working directory. This is the way single quotes are currently used in all previous MVS FTP servers, and is the default. Any single quotes inside the beginning and ending quote are treated as part of the file name.

The setting of this parameter affects all FTP subcommands that have a pathname as a parameter except parameters on the SITE subcommand. The minimum abbreviation for this parameter is QUO.

RDW

Specifies that Variable Record Descriptors (RDW) are treated as if they were part of the record and not discarded during FTP transmission of variable format data sets in stream mode.

Note: RDW information is stored in a binary halfword. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.

RECfm

Specifies the record format of a data set. When specified without the *format*, no record format will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify RECFM with no value when you are also specifying DATACLASS=*data_class* and you want the SMS dataclass to provide the RECFM *format*, or when you are specifying DCBDSN=*data_set_name* and you want to use the record format from the dcbdsn data set.

If RECFM=*format* is specified with either DATACLASS or DCBDSN, the value specified by the SITE RECFM parameter will override the DATACLASS or DCBDSN record format.

format

Specifies the record format of a data set. are: F, FA, FB, FBA, FBM, FBS, FBSA, FBSM, FM, VB, VBA, VBM, VBSA, VBSM, VBS, VM, VS, VSA, and VSM. The characters used to specify these record formats have the following meanings:

Code	Description
F	Fixed record length
V	Variable record length
U	Undefined record length
B	Blocked records
S	Spanned records (if variable) / standard records (if fixed)
A	Records contain ISO/ANSI control characters
M	Records contain machine code control characters

REtpd

Specifies the number of days that a newly allocated data set should be retained. When specified without the number of *days*, a retention period will not be specified when allocating new data sets. The equal sign (=) is optional in this case.

Specify RETPD with no value when you are also specifying DATACLASS=*data_class* or MGMTCLASS=*mgmtclass* and you want SMS to provide the RETPD value, or when you are specifying DCBDSN=*data_set_name* and you want to use the RETPD from the dcbdsn data set. If more than one of the SITE parameters (RETPD, MGMTCLASS, DATACLASS, or DCBDSN) are specified, the order of precedence (highest to lowest) is:

1. MGMTCLASS
2. RETPD

3. DCBDSN
4. DATACLASS

If a retention period is associated with an SMS management or data class, or with a model DCBDSN data set, the value of the retention period may be overridden to another retention period, but it may not be overridden to have no retention period specified for the newly created data sets.

days

Specifies the number of days that a newly allocated data set should be retained. The valid range is 0 through 9 999. A value of 0 indicates a retention period of 0 days so that the data set will expire the same day it was created.

Note: An attempt to either append or replace an existing dataset with a retention period requires operator interaction to take place for permission to alter the dataset. This is normal MVS behavior.

SBDATACONN

Specifies the EBCDIC ASCII code pages to be used for the data connection.

```
SITE SBDataconn=dsname
SITE SBDataconn=(ebcdic_cp,ascii_cp)
SITE SBDataconn=*
```

dsname

The fully-qualified name of an MVS data set or HFS file that contains the EBCDIC to ASCII and ASCII to EBCDIC translate tables generated by the CONVXLAT utility.

Notes:

1. The name must NOT be enclosed in quotes. If quotes appear, they will be treated as part of the name. (QUOtesoverride is ignored.)
2. The HFS name is case-sensitive. The MVS name may be entered in any case.
3. The name cannot begin with a left parenthesis (.
4. The SBDataconn parameter must be the only or last parameter on the SITE subcommand.

ebcdic_cp

The name of an EBCDIC code page recognized by iconv. For a list of code pages supported by iconv, see code set converters in an appendix in the *C/C++ MVS Programming Guide.*

ascii_cp

The name of an ASCII code page recognized by iconv. For a list of code pages supported by iconv, see code set converters in an appendix in the *OS/390 C/C++ Programming Guide.*

***** The translate tables set up at initialization for the data connection are to be used.

SECondary

Specifies the amount of tracks, blocks, or cylinders for secondary allocation. When specified without the *amount* for the C server, no secondary value will be used when allocating the data set. The equal sign (=) is optional in this case.

Specify SECONDARY with no value when you are also specifying DATACLASS=*dataclass* and you want the SMS dataclass to provide the SECONDARY value. To allow the SMS data class to determine the space allocation, both PRIMARY and SECONDARY must be specified with no value. The tracks/blocks/cylinders setting will be ignored in this case. If SECONDARY=*amount* is specified with DATACLASS, the value specified by the SITE SECONDARY parameter will override the DATACLASS space allocation.

amount
Specifies the amount of tracks, blocks, or cylinders for secondary allocation. The valid range is 0 through 16 777 215.

SPRead
Specifies that the output is in spreadsheet format when the file type is SQL.

SQLCol
Specifies the column headings of the SQL output file.

Any
The label of the DB2 SQL table column heading is the first choice for column heading, but if there is no label, the name becomes the column heading.

Labels
Labels are the DB2 SQL table column headings. If any of the columns do not have labels, FTP supplies a column heading in the form of

```
COLnnn
```

.

Names
Uses the names of the DB2 SQL table column headings. The labels are ignored.

STOrclass
Specifies the SMS storage class as defined by your organization for the target host. When specified without a *storage_class*, no parameter value cancels the storage class specification. The equal sign (=) is optional in this case.

See “Specifying Values for New Data Sets” on page 39 for more information about specifying attributes when allocating new data sets.

storage_class
Specifies the SMS storage class as defined by your organization for the target host. The SMS storage class may override settings for the VOLUME or UNIT site parameters.

TRacks
Specifies that primary and secondary space allocations are in tracks. If both PRIMARY and SECONDARY are unspecified, and an SMS data class has been specified, the space allocation will be determined by the SMS data class and the TRacks parameter will be ignored.

TRAILingblanks
Specifies that the FTP C server preserves the trailing blanks that are in a fixed format data set when the data is retrieved.

UMASK
Defines the file mode creation mask. The format of the UMASK parameter is UMASK ooo.

Unit
Specifies the unit type for allocation of new data sets.

The setting for Unit may be overridden by the SMS storage class, if one is specified. We recommend that Unit not be coded if an SMS storage class is in use.

unit_type
The unit type (for example, 3380 or TAPE) for the allocation of new data sets. You can use the Unit parameter to specify either direct access or tape device types. If *unit_type* is not specified, the unit type used for allocation is set back to the system default.

For tape output, only standard label (SL) and file sequence number 1 is supported. For input, multiple file volumes are supported if the data set is cataloged.

VOLume
Specifies the volume serial number for allocation of new data sets. If *volume_serial* is not specified, the volume serial number used for allocation is set back to the system default.

The setting for VOLume may be overridden by the SMS storage class, if one is specified. We recommend that VOLume not be coded if an SMS storage class is in use.

volume_serial
The volume serial number for allocation of new data sets.

WRAPrecord
Indicates that data is wrapped to the next record if no new line character is encountered before the logical record length of the receiving file is reached.

Usage

- Because more than one parameter can be specified with the SITE subcommand, *parameter* can be repeated many times, with each *parameter* separated by a blank space.
- If the remote host is using TCP/IP, use the SITE subcommand to set data set routing and send data set allocation attributes to the host. The site-dependent information sent with the SITE subcommand remains active until you issue a new SITE subcommand. The new SITE subcommand adds to, or changes, the attributes established by previous SITE subcommands.
- If you specify one or more incorrect parameters with the SITE subcommand, an error message specifying the incorrect parameter is displayed. All correct parameters are set, regardless of any incorrect parameters, and do not need to be reissued.
- If the data sets are sent from an MVS host, use the SENDSITE subcommand to automatically invoke the SITE subcommand.

  If the local host does not support the SITE subcommand, site information can be sent to the MVS FTP server from the client using the QUOTE subcommand.

- Use the HELP SERVER SITE command to get information on the SITE parameters supported by a server.

Context

- To check the effect of the SITE subcommand on the attributes at the foreign host, see “STATUS Subcommand—Retrieve Status Information from a Remote Host” on page 147.
- See “SENDSITE Subcommand—Toggle the Sending of Site Information” on page 132 for more information about the SENDSITE subcommand.
- See “QUOTE Subcommand—Send an Uninterpreted String of Data” on page 127 for more information about the QUOTE subcommand.
- See *JCL Reference Manual* for more information about some of the SITE and LOCSITE parameters.

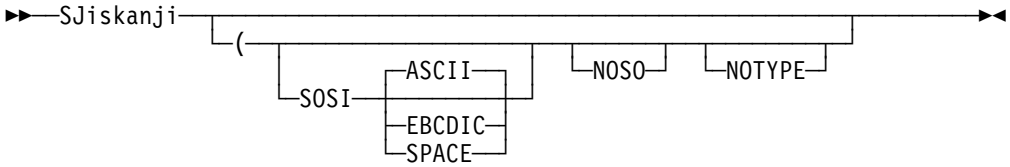
SJISKANJI Subcommand—Change the Data Transfer Type to SJISKANJI

Purpose

Use the SJISKANJI subcommand to change the data transfer type to SJISKANJI.

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see “FTP Command—Enter the FTP Environment” on page 87 and “Changing Local Site Defaults Using FTP.DATA” on page 33.

Format



Parameters

SOSI

Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII

When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Examples

To cause the FTP client to change its transfer type to Shift JIS Kanji, without sending a TYPE command to the FTP server:

```
SJISKANJI (NOTYPE
```

The server in this example should be set to the ASCII transfer type before the (NOTYPE subcommand is issued.

Usage

- The SJISKANJI client subcommand is equivalent to the TYPE B 1 server command.
- The SJISKANJI (SOSI or SJISKANJI (SOSI ASCII client subcommands are equivalent to the TYPE B 1 S A server command.
- The SJISKANJI (SOSI EBCDIC client subcommand is equivalent to the TYPE B 1 S E server command.
- The SJISKANJI (SOSI SPACE client subcommand is equivalent to the TYPE B 1 S S server command.
- The SJISKANJI (NOSO client subcommand is equivalent to the TYPE B 1 N server command.

Context

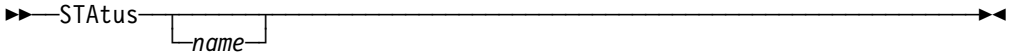
See “FTP with DBCS Support” on page 66 for more information.

STATUS Subcommand—Retrieve Status Information from a Remote Host

Purpose

Use the STATUS subcommand to retrieve configuration information in effect at the current time, from the FTP server. This information includes the current settings of the configuration variables, which can be initialized in the FTP.DATA data set or changed using various FTP subcommands. For information on the parameters of the FTP.DATA data set, refer to the *OS/390 TCP/IP OpenEdition Configuration Guide*

Format



Parameters

name

Specifies the file or directory for which status information is requested. The *name* parameter is not supported by the MVS FTP server.

Examples

- Retrieve the status information from an OE FTP server:

```
status
 >>>STAT
 211-Server FTP talking to host 9.67.112.25, port 1084
 211-User: USER21  Working directory: /u/user121/example
 211-The control connection has transferred 11015 bytes
 211-There is no current data connection.
 211-The next data connection will be actively opened
 211-to host 9.67.112.25, port 1084,
 211-using Mode Stream, Structure File, type ASCII, byte-size 8
 211-Automatic recall of migrated data sets.
 211-Automatic mount of direct access volumes.
 211-ASA control characters in ASA files opened for text processing
 211-will be transferred as ASA control characters.
 211-Trailing blanks are removed from a fixed format
 211-data set when it is retrieved.
 211-Data set mode.  (Do not treat each qualifier as a directory.)
 211-Primary allocation 5 tracks.  Secondary allocation 2 tracks.
 211-Partitioned data sets will be created with 15 directory blocks.
 211-FileType SEQ (Sequential - default).
 211-Number of access method buffers is 5
 211-RDWs from variable format data sets are discarded.
 211-Site DB2 subsystem name is DB2
 211-No DB2 subsystem in use
 211-Data not wrapped into next record.
 211-JESLRECL is 80
 211-JESRECFM is Fixed
 211-SMS is active.
 211-Mgmtclass for new data sets is TCPMGMT
 211-New data sets will be catalogued if a store operation terminates abnormally
 211-Single quotes will override the current working directory.
 211-UMASK value is 027
 211-Process id is 67108870
 211-Checkpoint interval is 0
 211 Record format VB, Lrecl: 256, Blocksize: 6233
 Command:
```

Usage

- The retrieved status information can be a directory, a file, or general status information, such as a summary of activity. If *name* is omitted, general status information is retrieved.
- Table 18 on page 150 lists the messages that are returned from the OE FTP server for the STATUS command. Within the table, the STATUS messages are listed in the order they appear, followed by:
  - Their usage and variable values
  - If the message is always or conditionally displayed
  - Where the values are set during server initialization
  - Where the values can be changed during a client session

For further information on setting values for server initialization in the FTP.DATA data set, see the *OS/390 TCP/IP OpenEdition Configuration Guide*.

Table 18 (Page 1 of 15). STATUS Subcommand Output

211-Server FTP talking to host *ip_address* port *port*	
The client's remote host *ip_address* and *port*.	Always displayed
211-User: *userid* Working directory: *directory*	
The *userid* that the client is currently logged in as and the *directory* the client is currently using. If the client has not entered a valid login (using the USER and PASS subcommand), then *userid* will be * NONE *. If the client is logged in as the "anonymous" user, then *userid* will be Anonymous. If the client has entered a valid userid and password, then *userid* will be the userid entered by the client on the USER subcommand. If the client currently has no working directory, then *directory* will be displayed as * NONE * If the current working directory is an HFS mount point, the working directory will be displayed as that mount point. If the current working directory is an MVS high level qualifier, the working directory will be displayed as that high level qualifier, followed by a period. If the current working directory is an MVS partitioned data set, the current working directory will be displayed, prefaced by the words“partitioned data set.”	Always displayed *userid* can be changed by the USER and PASS subcommands. *directory* can be changed by the CWD subcommand.
211-There is an RNFR pending for '*filename*'	
A rename is in progress for *filename*, and the RNTO subcommand has not been received yet.	Conditionally displayed when a rename is in progress.
211-The control connection has transferred *number* bytes	
The total number of bytes that have been sent or received over the control connection between the client and the server.	Always displayed
211-The current data connection has transferred: *number* bytes 211-using Mode *mode*, Structure: *structure*, Type: *type*, byte-size 8	

Table 18 (Page 2 of 15). STATUS Subcommand Output

The *number* of bytes sent or received over the data connection. The *mode* being used for the data transfer. The data set *structure* currently being used by the server for transfer. The data transfer *type* currently being used by the server for transfer.	Conditionally displayed when there is a currently active data connection. *mode* can be changed by the `MODE` subcommand. *structure* can be changed by the `STRU` subcommand. *type* can be changed by the `TYPE` subcommand.
211-There is no current data connection.	
There is currently no data connection between the client and the server.	Conditionally displayed when there is no data connection.
211-The next data connection will be *open_mode* opened	
Indicates how the next data connection is opened. If *open_mode* is "actively", then the server issues the connect request to the client, which should be in "listen" state on the client data port. If *open_mode* is "passively", then the server listens on the data port for the client to establish the data connection.	Always displayed
211-to host *IP_address*, port *port*,	
The remote host *IP_address* and *port* the next data connection will connect to.	Conditionally displayed when the next data connection will be opened actively.
211-using Mode *mode*, Structure *structure*, type *type*, byte-size 8	
The *mode* used for the next data transfer. The *structure* used by the server for file transfer. The data transfer *type* used by the server for file transfer.	Always displayed. *mode* can be changed by the `MODE` subcommand. *structure* can be changed by the `STRU` subcommand. *type* can be changed by the `TYPE` subcommand.
211-Automatic recall of migrated data sets.	

Table 18 (Page 3 of 15). STATUS Subcommand Output

Migrated data sets are recalled when they are accessed for data transfer or file rename. The AUTORECALL setting does not apply when deleting an entire migrated data set. If a DELE subcommand is entered to delete a migrated data set, the data set is not recalled prior to deletion. However, if the DELE subcommand is entered to delete a member of a migrated partitioned data set, the data set is recalled.	Conditionally displayed when AUTORECALL has been set to TRUE and has not been overridden with a SITE NOAUTORECALL subcommand. Set by specifying either the AUTORECALL statement in the FTP.DATA data set and the AUTORECALL FTP parameter. Changed by issuing theSITE subcommand with the AUTORECALL parameter.
211-No Automatic recall of migrated data sets.	
Migrated data sets are not automatically recalled when accessed for data transfer or file rename. Subcommands issued for a migrated data set are rejected. The NOAUTORECALL setting does not apply when deleting an entire migrated data set. If a DELE subcommand is entered to delete a migrated data set, the data is not recalled prior to deletion; the data set is deleted even though the data set was migrated and NOAUTORECALL was specified. However, if the DELE subcommand is used to delete a member of a migrated partitioned data set, the data set is not recalled and the DELE subcommand is rejected.	Conditionally displayed when AUTORECALL has been set to FALSE and has not been overridden with a SITE AUTORECALL subcommand. Set by specifying the AUTORECALL statement in the FTP.DATA data set and the NOAUTORECALL FTP parameter. Changed by issuing the SITE subcommand with the AUTORECALL parameter.
211-Automatic mount of direct access volumes.	
If a data set resides on a volume that is not mounted, the operator is automatically prompted to mount the necessary volume on the MVS system. The FTP server waits for the operator to mount the volume and reply to the prompt at the operator console before continuing.	Conditionally displayed when AUTORECALL has been set to TRUE and has not been overridden with a SITE NOAUTOMOUNT subcommand. Set during server initialization by specifying either the AUTOMOUNT statement in the FTP.DATA data set or the AUTOMOUNT FTP parameter. Changed by issuing the SITE subcommand with the AUTOMOUNT parameter.
211-No Automatic mount of direct access volumes.	

Table 18 (Page 4 of 15). STATUS Subcommand Output

If a data set resides on a volume that is not mounted, the FTP subcommand is rejected.	Conditionally displayed when AUTOMOUNT has been set to FALSE and has not been overridden with a `SITE AUTOMOUNT` subcommand. Set by specifying either the `AUTOMOUNT` statement in the FTP.DATA data set or the `NOAUTOMOUNT` FTP parameter. Changed by issuing the `SITE` subcommand with the `NOAUTOMOUNT` parameter.
211-ASA control characters in ASA files opened for text processing will 211-be converted to C control character sequences during file transfer.	
If the record format (RECFM) of the file indicates that the file contains ISO/ANSI control characters (RECFM with "A"), then the ISO/ANSI control characters are converted to C control character sequences during file transfer. Files stored at the FTP server host are stored with the converted characters rather than the ISO/ANSI control characters.	Conditionally displayed when ASATRANS has been set to TRUE and has not been overridden by the `SITE NOASATRANS` subcommand. Set by specifying the `ASATRANS` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `ASATRANS` parameter.
211-ASA control characters in ASA files opened for text processing 211-will be transferred as ASA control characters.	
If the record format (RECFM) of the file indicates that the file control ISO/ANSI control characters (RECFM with "A"), then the ISO/ANSI control characters remain unchanged during file transfer. Files stored at the FTP server host are stored with the ISO/ANSI control characters.	Conditionally displayed when ASATRANS has been set to FALSE and has not been overridden by the `SITE ASATRANS` subcommand. Set during server initialization by specifying the `ASATRANS` statement in the FTP.DATA data set. Changed during a client session by issuing the `SITE` subcommand with the `NOASATRANS` parameter.
211-Trailing blanks are removed from a fixed format 211-data set when it is retrieved.	

Table 18 (Page 5 of 15). STATUS Subcommand Output

If a RETR subcommand is issued to retrieve a data set with a fixed record format, any trailing blanks at the end of each record in the data set are removed before the record is sent to the client.	Conditionally displayed when TRAILINGBLANKS has been set to FALSE and has not been overridden with a SITE TRAILINGBLANKS subcommand. Set by specifying the TRAILINGBLANKS statement in the FTP.DATA data set. Changed during a client session by issuing the SITE subcommand with the NOTRAILINGBLANKS parameter.
211-Trailing blanks are not removed from a fixed format 211-data set when it is retrieved.	
If a RETR subcommand is issued to retrieve a data set with a fixed record format, any trailing blanks at the end of each record of the data set are included with the record that is sent to the client.	Conditionally displayed when TRAILINGBLANKS has been set to TRUE and has not been overridden by the SITE NOTRAILINGBLANKS subcommand. Set during server initialization by specifying the TRAILINGBLANKS statement in the FTP.DATA data set. Changed during a client session by issuing the SITE subcommand with the TRAILINGBLANKS parameter.
211-Data set mode. (Do not treat each qualifier as a directory.)	
All the data set qualifiers below the current directory are treated as entries in the directory.	Conditionally displayed when DATASETMODE has been set to TRUE and has not been overridden by the SITE subcommand with the DIRECTORYMODE parameter. Set by specifying the DATASETMODE statement in the FTP.DATA data set. Changed by issuing the SITE subcommand with the DATASETMODE parameter.
211-Directory mode. (Treat each qualifier as a directory.)	

Table 18 (Page 6 of 15). STATUS Subcommand Output

Only the data set qualifier immediately below the current directory is treated as an entry in the directory. Directory mode applies only to MVS PS and PDS data sets.	Conditionally displayed when DATASETMODE has been set to FALSE, or DIRECTORYMODE has been set to TRUE and has not been overridden by the `SITE DATASETMODE` subcommand. Set by specifying the `DATASETMODE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `DIRECTORYMODE` parameter.
211-Primary allocation *p_amt spacetype*. Secondary allocation *s_amt spacetype* - or - 211-Primary allocation *p_amt spacetype*. - or - 211-Secondary allocation *s_amt spacetype*	

These messages display the primary and secondary allocation information used when allocating new data sets. *spacetype* indicates whether the primary and secondary space is allocated as BLOCKS, TRACKS, or CYLINDERS
 	
p_amt is the number of BLOCKS, TRACKS, or CYLINDERS allocated for the primary extent.
 	
s_amt is the number of BLOCKS, TRACKS, or CYLINDERS allocated for each secondary extent.
 	
If the primary or secondary or both space information is displayed, these settings override any space allocation information contained in an SMS DATACLASS. Both primary and secondary allocation information must be unspecified for SMSDATA CLASS space allocation information to be used.
	Only one line will be displayed conditionally when values have been set.
	
	Set by specifying the FTP.DATA data set statement:
	`PRIMARY p_amt`
	or
	`SECONDARY s_amt`
	or
	`SPACETYPE`
	
	Set by issuing the subcommand:
	
	`SITE PRIMARY=p_amt`
	or
	`SITE SECONDARY=s_amt`
	or
	`SITE`
	
	with either the `BLOCKS`, `TRACKS`, or `CYLINDERS` parameter.

211-Partitioned data sets will be created with *number* directory blocks.	
The number of directory blocks allocated for a new partitioned data set. If this is displayed in the `STAT` subcommand output, then a value has been specified for DIRECTORY and this value will override any value specified for directory blocks in an SMS DATACLASS. To use the value specified for directory blocks in an SMS DATACLASS, specify either `DIRECTORY` with no value in the FTP.DATA data set, or specify the `SITE DIRECTORY` subcommand with no value.	Conditionally displayed when a value has been specified for directory blocks and was not overridden by the `SITE DIRECTORY` subcommand. Set by specifying the `DIRECTORY` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `DIRECTORY` parameter.

Table 18 (Page 7 of 15). STATUS Subcommand Output

211-FileType SEQ (Sequential - default)	
The server is currently operating in the SEQ (sequential) filetype mode. FTP subcommands will process MVS sequential or partitioned data sets, or partitioned data set members.	Conditionally displayed when FILETYPE has been set to SEQ and has not been overridden by the `SITE FILETYPE=JES` or `SITE FILETYPE=SQL` subcommand. Set by specifying the `FILETYPE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `FILETYPE` parameter.
211-FileType JES (MVS job spool) JES name is *jesname*	
The server is currently operating in the JES (MVS job spool) filetype mode with *jesname* as the name of the JES subsystem. When the server filetype is JES, FTP subcommands can submit, list, or delete JES jobs.	Conditionally displayed when FILETYPE has been set to JES and has not been overridden by the `SITE FILETYPE=SEQ` or `SITE FILETYPE=SQL` subcommand. Set by specifying the `FILETYPE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `FILETYPE` parameter.
211-FileType SQL (returns results of DB2 query)	
The server is currently operating in the SQL (DB2 query) filetype mode. When the server filetype is SQL, FTP subcommands can submit DB2 queries and retrieve the resulting DB2 output.	Conditionally displayed when FILETYPE has been set to SQL and has not been overridden by the `SITE FILETYPE=SEQ` or `SITE FILETYPE=JES` subcommand. Set by specifying the `FILETYPE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `FILETYPE` parameter.
211-Number of access method buffers is `number`	

Table 18 (Page 8 of 15). STATUS Subcommand Output

The number of access method buffers that are used when data is read from or written to a data set. The valid range is 1 through 255.	Always displayed Set by specifying the `BUFNO` statement in the FTP.DATA data set. Default is 5. `number` can be changed by issuing the `SITE` subcommand with the `BUFNO` parameter.
211-RDWs from variable format data sets are retained as part of the data.	
Variable Record Descriptors (RDW) are treated as if they are part of the record and not discarded during FTP transmission of variable format data sets in stream mode. RDW information is stored in binary halfword. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.	Conditionally displayed when RDW has been set to TRUE and has not been overridden with the `SITE NORDW` subcommand. Set during server initialization by specifying the `RDW` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `RDW` parameter.
211-RDWs from variable format data sets are discarded.	
NORDW specifies that the Variable Record Descriptors (RDW) are discarded during FTP transmission of variable format data sets.	Conditionally displayed when RDW has been set to FALSE and has not been overridden with the `SITE RDW` subcommand. Set during server initialization by specifying the `RDW` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `NORDW` parameter.
211-Retention period is *number_of_days*	

Table 18 (Page 9 of 15). STATUS Subcommand Output

A retention period of *number_of_days* is assigned to a newly created data set. A retention period of "0 days" is not the same as "no retention period". If this line is not displayed, then no retention period is assigned to newly created data sets. If this line is displayed, and *number_of_days* has been set to 0, a retention period of "0 days" is assigned to newly created data sets. Consequently, newly created data sets expire the same day they are created. If this line is displayed, then a retention period has been specified; this value overrides the retention period specified in an SMS data class. If an SMS management class has been specified, the retention period defined in the SMS management class may override the *number_of_days* specified by the FTP server.	Conditionally displayed when a value has been set for RETPD and has not been overridden by the `SITE RETPD` subcommand. Set by specifying the `RETPD` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `RETPD` parameter.
211-Site DB2 subsystem name is *db2name*	
SQL queries are sent to the DB2 subsystem indicated by *db2name*.	Conditionally displayed Set by specifying the `DB2` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `DB2` parameter.
211-Wrapping data into next record.	
When the FTP server stores data at the server's host, data wraps to a new record if it reaches the logical record length before encountering a new line character.	Conditionally displayed when WRAPRECORD has been set to TRUE and has not been overridden by the `SITE NOWRAPRECORD` subcommand. Set by specifying the `WRAPRECORD` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `WRAPRECORD` parameter.
211-Data not wrapped into next record.	
When the FTP server stores data at the server's host, data is truncated if it reaches the logical record length before encountering a new line character.	Conditionally displayed when WRAPRECORD has been set to FALSE and has not been overridden by the `SITE WRAPRECORD` subcommand. Set by specifying the `WRAPRECORD` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `NOWRAPRECORD` parameter.

Table 18 (Page 10 of 15). STATUS Subcommand Output

211-SQL results sent in *report_format* format.	
The format of SQL query output.	Conditionally displayed when FILETYPE is set to SQL. Set by specifying the SPREAD statement in the FTP.DATA data set. Changed by issuing the SITE subcommand with either the SPREAD or NOSPREAD parameter.
211-SQLCOL (column headings) use *heading*	
The column headings for the SQL output	Conditionally displayed when FILETYPE is set to SQL. Set by specifying the SQLCOL statement in the FTP.DATA data set. Changed by issuing the SITE subcommand with the SQLCOL parameter.
211-JESLRECL is *lrecl*	
The logical record length for the JES internal reader at the server's host system. If *lrecl* is *, then the JES logical record length is the same as the logical record length used to allocate new data sets.	Always displayed Set by specifying the JESLRECL statement in the FTP.DATA data set. Changed by issuing the SITE subcommand with the JESLRECL parameter.
211-JESRECFM is *recfm*	
The record format for the JES internal reader at the server's host system. If *recfm* is *, then is truncated if it the JES record format is the same as the record format used to allocate new data sets.	Always displayed Set by specifying the JESRECFM statement in the FTP.DATA data set. Changed by issuing the SITE subcommand with the JESRECFM parameter.
211-Migrate Volid is *volid*	

Table 18 (Page 11 of 15). STATUS Subcommand Output

The FTP server recognizes a data set with a volser of *volid* as a migrated data set.	Always displayed Set by specifying the `MIGRATEVOL` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `MIGRATEVOL` parameter.
211-NJE Destination is User *userid* Node *node* - or - 211-NJE Destination is *destination*	
Files transferred to the FTP Server are forwarded to the specified *userid* at the specified *node*, or to the specified *destination*, rather than stored at the host.	Conditionally displayed when a remote destination has been specified for server files. Set by one of the following: The `SITE DEST=dest` subcommand has been issued. The FTP.DATA data set specified `DEST dest` and this setting has not been overridden with a `SITE DEST` subcommand.
211-SMS is active.	
The FTP server has determined that SMS is active on the server's host MVS system.	Conditionally displayed when SMS is active.
211-SMS is not active.	
The FTP server has determined that SMS is not active on the server's host MVS system.	Conditionally displayed when SMS is not active.
211-Mgmtclass for new data sets is *mgmtclass*	
The specified SMS Management class, *mgmtclass*, is used when allocating a new data set. If a retention period was defined in the specified SMS management class, this value may override the retention period specified by the FTP server.	Conditionally displayed when an SMS management class has been specified for the server and this setting has not been overridden by the `SITE MGMTCLASS` subcommand. Set by specifying the `MGMTCLASS` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `MGMTCLASS` parameter.

Table 18 (Page 12 of 15). STATUS Subcommand Output

211-Storclass for new data sets is *storclass*	
SMS Storage class used when allocating a new data set. The specified SMS storage class may override the FTP server's settings for the volume and unit for new data set allocation.	Conditionally displayed when an SMS storage class has been specified for the server and this setting has not been overridden by the `SITE STORCLASS` subcommand. Set by specifying the `STORCLASS` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `STORCLASS` parameter.
211-Dataclass for new data sets is *dataclass*	
SMS Dataclass used when allocating a new data set. The STATUS message's value for the following parameters overrides the corresponding parameter in the specified SMS data class: BLKSIZE DIRECTORY LRECL PRIMARY RECFM RETPD SECONDARY	Conditionally displayed when an SMS data class has been specified for the server and this setting has not been overridden by the `SITE DATACLASS` subcommand. Set by specifying the `DATACLASS` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `DATACLASS` parameter.
211-Data sets will be allocated on *volser.*	
The volume on which new data sets are allocated. If no volume was specified for data set allocation, new data sets are allocated on the system default volume. If an SMS storage class has been specified, the SMS storage class may override the setting of *volser*.	Conditionally displayed when a volume has been specified for data set allocation and this setting was not overridden by the `SITE VOLUME` subcommand. Set by specifying the `VOLUME` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `VOLUME` parameter.

Table 18 (Page 13 of 15). STATUS Subcommand Output

211-Data sets will be allocated using unit *unit_name.*	
The unit on which new data sets will be allocated. If an SMS storage class has been specified, the SMS storage class may override the setting of *unit_name*	Conditionally displayed when a unit has been specified for data set allocation and the setting has not been overridden with a `SITE UNIT` subcommand. Set by specifying the `UNIT` statement of the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `UNIT` parameter.
211-New data sets will be *disposition* if a store operation terminates abnormally	
This line indicates the *disposition* of new data sets if the `STOR`, `STOU`, or `APPE` subcommand terminates while the data is being stored in a newly created data set at the server. If *disposition* is "catalogued", the new data set is kept and catalogued. Data in the data set may be incomplete. If *disposition* is "deleted", the server deletes the new data set.	Always displayed Set by specifying the `CONDDISP` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `CONDDISP` parameter.
211-Single quotes will override the current working directory.	
Single quotes appearing at the beginning and end of a file name are interpreted to mean that the file name contained inside the single quotes should override the current working directory instead of being appended to the current working directory.	Conditionally displayed when `QUOTESOVERRIDE` has been set to TRUE and has not been overridden with the `SITE NOQUOTESOVERRIDE` subcommand. Set by during server initialization by specifying the `QUOTESOVERRIDE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `QUOTESOVERRIDE` parameter.
211-Single quotes will be treated as part of the file name.	
A single quote at the beginning, as well as other single quotes contained in the file name, will be treated as part of the actual file name. The entire file name including the leading single quote will be appended to the current working directory.	Conditionally displayed when `NOQUOTESOVERRIDE` has been set to TRUE and has not been overridden with the `SITE QUOTESOVERRIDE` subcommand. Set by during server initialization by specifying the `QUOTESOVERRIDE` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `NOQUOTESOVERRIDE` parameter.

Table 18 (Page 14 of 15). STATUS Subcommand Output

211-UMASK value is*umask_value*	
umask_value is the octal mask that specifies which permission bits must be set OFF for a newly created file.	Always displayed Set by during server initialization by specifying the `UMASK` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `UMASK` parameter.
211-Process id is *pid number*	
The process identifier for the current client session logged into the FTP server.	Always displayed
211-Checkpoint interval is *interval*	
The *interval* at which the server sends checkpoint markers to the client when the client is retrieving a file. Checkpoint markers are sent only if the file transfer is done in EBCDIC in BLOCK or COMPRESSED mode. An *interval* of 0 indicates that no checkpoint markers are sent.	Always displayed Set by specifying the `CHKPTINT` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `CHKPTINT` parameter.
211-*'dsn'* will be used as the DCBDSN model data set	
Data set used as a model DCB when allocating new data sets. Unless otherwise specified, the LRECL, RECFM, and BLKSIZE of *dsn* are used for the newly allocated data set. If the STAT subcommand output shows that a value has been specified for the LRECL, RECFM, or BLKSIZE parameters, these values override the information obtained from the model DCB data set. To use the BLKSIZE, LRECL, and RECFM of the model DCB data set, *dsn*, the LRECL, BLKSIZE, and RECFM parameters must be unspecified by one of the following methods: Specify the BLKSIZE, RECFM, and LRECL keywords in the FTP.DATA data set with no value Issue the `SITE LRECL RECFM BLKSIZE` subcommand with no values for the parameters.	Conditionally displayed when a model DCB has been specified for the server and this setting has not been overridden by issuing the `SITE DCBDSN` subcommand. Set by specifying the `DCBDSN` statement in the FTP.DATA data set. Changed by issuing the `SITE` subcommand with the `DCBDSN` parameter.

Table 18 (Page 15 of 15). STATUS Subcommand Output

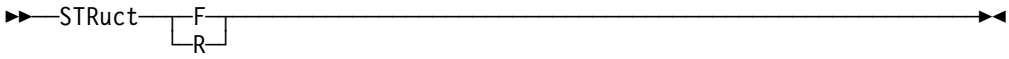
211 Record format *recfm*, Lrecl: *lrecl*, Blocksize: *blksize*
- or -
211 Record format *recfm*, Lrecl: *lrecl*
- or -
- 211 Record format *recfm*, Blocksize: *blksize*
- or -
211 Lrecl: *lrecl*, Blocksize: *blksize*
- or -
211 Record format *recfm*
- or -
211 Lrecl: *lrecl*
- or -
211 Blocksize: *blksize*
	The logical record length, block size, and record format assigned to newly allocated data sets.
	
	If one of these lines displays, then a value was specified for one or more of LRECL, BLKSIZE, and RECFM. Any values specified for LRECL, BLKSIZE, or RECFM override the corresponding value of a model DCB or SMS data class, if specified.
	
	If values for *lrecl*, *blksize*, or *recfm* are not included in the message, then no values have been specified and the value is obtained from a model DCB data set, or from an SMS data class, if one has been specified.
		Conditionally displayed when values have been specified.
		
		Set by specifying the FTP.DATA data set statement:
		`LRECL`
		
		`BLKSIZE`
		
		`RECFM`
		
		Changed by issuing:
		
		the `SITE` subcommand with the`LRECL` parameter
		
		the `SITE` subcommand with the `BLKSIZE` parameter
		
		the `SITE` subcommand with the `RECFM` parameter

STRUCT Subcommand—Set the File Structure

Purpose

Use the STRUCT subcommand to set the file structure.

Format



Parameters

F Sets the file structure to file. The file structure affects both the transfer mode and the interpretation and storage of the data set or file. With a file structure of F, the data being transferred is considered to be a continuous sequence of data bytes.

R Sets the file structure to record.

Usage

Record structure (STRUCT R) is not supported by the MVS client. It is supported by the MVS server for PS and PDS data sets, but cannot be used with HFS files.

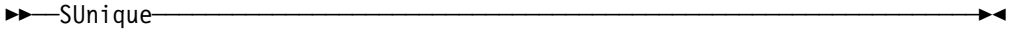
SUNIQUE Subcommand—Toggle the Storage Method

Purpose

Use the SUNIQUE subcommand to toggle the method of storing files on the foreign host.

Format

Use the SUNIQUE subcommand in the following format:



Parameters

None.

Usage

By default, SUNIQUE is toggled off, and FTP uses a store command (STOR) with the PUT and MPUT subcommands. If the foreign host already has a data set or file with the name specified by *foreign_file*, the foreign host overwrites the existing data set or file.

If SUNIQUE is toggled on, FTP uses a store-unique command (STOU) with the PUT and MPUT subcommands, and prevents you from overwriting or erasing the existing data set or file on the foreign host. The created foreign data set or file is stored with a unique name. FTP sends the unique name of the created foreign data

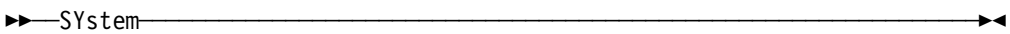
set or file to the local host, where the data set or file name is displayed on your terminal.

SYSTEM Subcommand—Display the Operating System Name

Purpose

Use the SYSTEM subcommand to display the name of the remote host's operating system. The remote host must have also implemented the SYSTEM subcommand.

Format



Parameters

None.

Usage

- If the MVS FTP server is the V3R2 FTP server, 215 MVS is the operating system of this server.
- If the MVS FTP server is the V3R1 Pascal FTP server, 215 OS/MVS is the operating system of this server.
- If the MVS FTP server is the V3R1 C FTP server, 215 MVS is the operating system of this server. FTP Server is the C-server.
- If the MVS FTP server is the V3R2 C FTP server, 215 MVS is the operating system of this server.
- If the MVS FTP server is the MVS OE-FTP server, 215 MVS is the operating system of this server. FTP Server is the C-server running on OE.

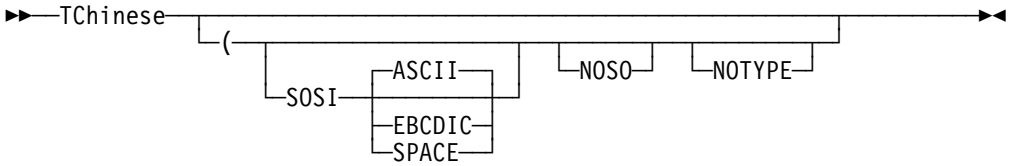
TCHINESE Subcommand—Change the Data Transfer Type to TCHINESE

Purpose

Use the TCHINESE subcommand to change the data transfer type to Traditional Chinese (5550).

CAUTION:
The MVS FTP client uses the same SBCS translate table for single-byte or double-byte data transfers. If you require an alternate SBCS table for a double-byte transfer, start a new FTP session using the TRANS options on the FTP command or specify SBTRANS in your local FTP.DATA file. For more information, see "FTP Command—Enter the FTP Environment" on page 87 and "Changing Local Site Defaults Using FTP.DATA" on page 33.

Format



Parameters

SOSI
Transferred data will contain the shift-out and shift-in characters specified by the following parameter - ASCII, EBCDIC or SPACE. If no parameter is specified, then ASCII will be used as the default.

specified at all, shift-out/shift-in characters are not used in the transferred data.

ASCII
When combined with the SOSI parameter, causes shift-out and shift-in characters X'1E' and X'1F' to be used to delimit DBCS strings in ASCII data.

EBCDIC
When combined with the SOSI parameter, causes shift-out and shift-in characters X'0E' and X'0F' to be used to delimit DBCS strings in ASCII data.

SPACE
When combined with the SOSI parameter, causes shift-out and shift-in characters X'20' and X'20' (ASCII spaces) to be used to delimit DBCS strings in ASCII data.

NOSO
Specifies that the data transferred is pure DBCS (this is, data with no SBCS characters) and that the data is to be transferred to/from EBCDIC DBCS data that contains no shift-out/shift-in delimiters.

NOTYPE
Suppresses the sending of the corresponding TYPE command to the server. Use this parameter when translation is to be done by the FTP client only.

Usage

The TCHINESE client subcommand is equivalent to the TYPE B 7 server command.

Context

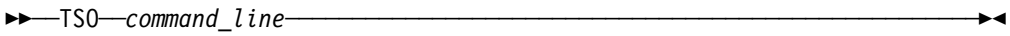
See “FTP with DBCS Support” on page 66 for more information.

TSO Subcommand—Use TSO Commands

Purpose

Use the TSO subcommand to pass a Time Sharing Option (TSO) command to a local host TSO environment.

Format



Parameters

command_line Specifies a TSO command. Do not use synonyms.

Usage

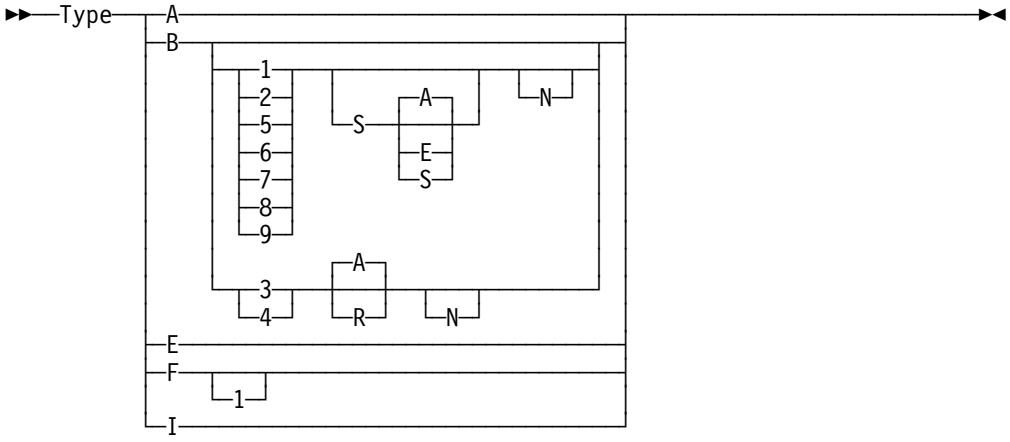
You cannot issue another TCP/IP command. The VMCF interface is used when you use the FTP command. If the VMCF interface is used again while in the FTP session (for example, doing a TSO NETSTAT or a TSO PING), the next nonlocal FTP subcommand will fail. That nonlocal subcommand causes the connection to be closed and the FTP session ends abnormally.

TYPE Subcommand—Set the Data Transfer Type

Purpose

Use the TYPE subcommand to set the data transfer type for the client and server at the same time with one command. FTP supports the ASCII, EBCDIC, image (binary), and two data transfer types.

Format



Parameters

A Sets the transfer type as ASCII. Specifying the ASCII transfer type has the same effect as using the ASCII subcommand. The ASCII transfer type is intended for the transfer of data to or from an ASCII host. ASCII is the default transfer type.

B Sets the transfer type as DBCS. Specifying the B transfer type with the appropriate options has the same effect as using the BIG5, EUCKANJI, HANGEUL, JIS78KJ, JIS83KJ, KSC5601, SJISKANJI, SCHINESE, or TCHINESE subcommands. If B is specified alone, the second type parameter defaults to 1 and current transfer type is changed to Shift JIS Kanji

CAUTION:
When you transfer double-byte data, the currently active SBCS translation table will be used for an SBCS characters in the data set. If necessary, use the SITE SBDATACONN FTP command to select an alternate SBCS translation table that is appropriate for your data before transferring your double-byte data.

B 1 Change current transfer type to Shift JIS Kanji

B 2 Change current transfer type to Extended Unix Code Kanji

B 3 Change current transfer type to JIS 1983 Kanji

B 4 Change current transfer type to JIS 1978 Kanji

B 5 Change current transfer type to Hangeul

B 6 Change current transfer type to Korean Standard Code KSC-5601, 1989 version

B 7 Change current transfer type to Traditional Chinese (5550)

B 8 Change current transfer type to Big-5.

B 9 Change current transfer type to Simplified Chinese.

S Transferred data contains shift-out and shift-in delimeters.

If S is specified alone, the second parameter defaults to A. Shift-out and shift-in characters `X'1E'` and `X'1F'` are used.

The S parameter can be used to control the use of shift-out (SO) and shift-in (SI) characters during DBCS data transfer for Big5, SChinese, Shift-JIS Kanji, EUC Kanji, Hangeul, KSC-5601, and TChinese.

If SOSI is not specified at all, shift-out/shift-in characters are not used in the transferred data.

S A Use shift-out and shift-in characters X'1E' and X'1F' in the transferred data.

S E Use shift-out and shift-in characters X'0E' and X'0F' in the transferred data.

S S Use ASCII spaces (X'20') as shift-out and shift-in characters in the transferred data.

A Use ASCII shift-in escape sequence `ESC ( B`. This is the default. (Used for DBCS data types JIS 1983 Kanji and JIS 1978 Kanji only.)

R Use JISROMAN shift-in escape sequence `ESC ( J`. (Used for DBCS data types JIS 1983 Kanji and JIS 1978 Kanji only.)

N Indicates the transfer is to be pure DBCS data (that is, data with no SBCS characters) and that the data is to be transferred to or from EBCDIC DBCS data which contains no shift-out/shift-in delimiters.

transferred from the EBCDIC host, the entire data set is assumed to be EBCDIC DBCS with no SO/SI characters in the data. The data is then con-

verted to the required ASCII type and if any the transferred data then the corresponding SO/SI characters are used to delimit the ASCII DBCS strings.

When transferring data to the EBCDIC host, no SO/SI characters are inserted and if any SO/SI option is specified corresponding SO/SI characters are removed from the ASCII data and not replaced at the host. The length of data may change during transfer to and from the EBCDIC host when pure DBCS is specified with any SO/SI option. When pure DBCS is specified by itself the length of data will not change. shift-out/shift-in characters X'0E' and X'0F' are used at the host.

E Sets the transfer type as EBCDIC. Specifying the EBCDIC transfer type has the same effect as using the EBCDIC subcommand. The EBCDIC transfer type is intended for efficient transfer between hosts that use EBCDIC for their internal character representation.

F Sets the transfer type as EBCDIC IBM Kanji. Specifying the IBM Kanji transfer type has the same effect as using the IBMKANJI subcommand.

F 1 Change current transfer type to IBM (EBCDIC) Kanji

I Sets the transfer type as image (binary). Specifying the image transfer type has the same effect as using the BINARY subcommand. With the image transfer type, data is sent as contiguous bits, packed into 8-bit bytes. The image transfer type is used for the efficient storage and retrieval of data sets or files, and for the transfer of binary data.

Examples

- Transfer text data from an EBCDIC host to an ASCII host:

```
System:       Command:
  User:      ascii
System:
              >>>TYPE a
              200 Representation type is ASCII.
              Command:
```

- Transfer binary data from an EBCDIC host to an ASCII host:

```
System:       Command:
  User:      type i
System:
              >>>TYPE i
              200 Representation type is IMAGE.
              Command:
```

- Transfer text data from an ASCII host to an EBCDIC host:

```
System:       Command:
  User:      type a
System:
              >>>TYPE a
              200 Representation type is ASCII.
              Command:
```

- Transfer binary data from an ASCII host to an EBCDIC host:

```
System:      Command:
  User:     type i
System:
             >>>TYPE i
             200 Representation type is IMAGE.
             Command:
```

- Transfer text data from an EBCDIC host to an EBCDIC host:

```
System:      Command:
  User:     ebcdic
System:
             >>>TYPE e
             200 Representation type is EBCDIC.
             Command:
```

- Transfer binary data from an EBCDIC host to an EBCDIC host:

```
System:      Command:
  User:     type i
System:
             >>>TYPE i
             200 Representation type is IMAGE.
             Command:
```

- Set the transfer type to JIS 1983 Kanji using the JISROMAN shift-in escape sequence `ESC ( J`:

  ```
  TYPE B 3 R
  ```

- Set the transfer type to Shift-JIS Kanji using the EBCDIC SO/SI characters X'0E'/X'0F'in the transferred date:

  ```
  TYPE B 1 S E
  ```

Usage

If no SO/SI option is specified by the TYPE command for BIG5, SCHINESE, EUCKANJI, HANGEUL, KSC5601, SJISKANJI, or TCHINESE, then standard DBCS control is used for the data transfer. This means that no SO/SI characters are placed in the ASCII data when transferring from the (EBCDIC) host to ASCII and the value of each ASCII character is used to determine if it is a single-byte character or part of a double-byte character when transferring to the host. For JIS 1983 Kanji and JIS 1978 Kanji, three-character escape sequences are always used to delimit DBCS strings in mixed SBCS/DBCS ASCII data. These escape sequences cannot be altered by using the S, S A, S E, or S S parameters.

If no SO/SI option is specified, then the length of data may change as it is transferred to or from the EBCDIC host since EBCDIC DBCS types on the host contain SO/SI characters in mixed SBCS/DBCS data to determine which characters are part of a DBCS string. Any of the above SO/SI options (S, S A, S E or S S) maybe used for mixed SBCS/DBCS data so that the length of data will not change when transferred to or from the EBCDIC host. Use of three-character escape sequences for JIS 1983 Kanji and JIS 1978 Kanji means that the length of data for these types

will always change when transferring mixed SBCS/DBCS data to or from the EBCDIC host.

The option to use ASCII spaces as SO/SI characters in the transferred data should be used only for transfer from the EBCDIC host. Data may be transferred to the host when using this option but care must be taken as each ASCII space is interpreted as a shift-out or shift-in character and is replaced with the corresponding SO/SI character on the host.

Context

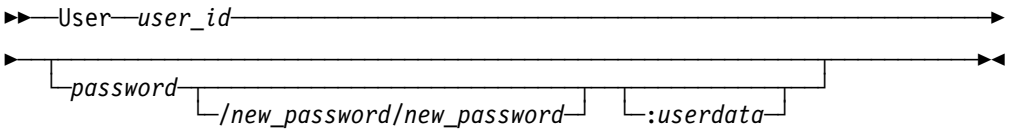
For more information about transfer methods, see Table 9 on page 25.

USER Subcommand—Identify Yourself to a Host or Change Your TSO User ID Password

Purpose

Use the USER subcommand to identify yourself to the host after opening a connection or, if the foreign host is an MVS FTP server, you can change your TSO user ID password.

Format



Parameters

user_id
Specifies your logon name on the host.

password
Specifies your current password on the host. If you do not supply a password when invoking the USER subcommand, you are prompted to enter a password if the host requires a logon password.

new_password
Specifies your new password on the host. You must enter the password twice.

:userdata
An optional character string to be passed to the FTCHKPWD user exit routine. The userdata must be separated from the password information by a colon (:) and may be any combination of up to 200 non-blank characters and numbers except the colon. Care should be taken when using the backslash character (\) in combination with other characters which might be interpreted as an escape sequence by the C compiler.

Usage

- If you enter your password incorrectly, you are not prompted to enter the password again. You must reissue the USER subcommand to enter the correct password.
- If you do not specify *password/new_password/new_password* on the USER subcommand, you can specify it when you are prompted for the password after entering the USER subcommand. You can specify the USER subcommand to change your TSO *user_id* password at any time during the FTP session.
- To avoid having your password print when issuing your user ID and password as part of a CLIST or batch job, enter your user ID and password on separate lines.
- not place any spaces between the passwords and the slashes (/), and the userdata.
- You can use the NETRC data set to automatically log on to a remote host.

Context

For information about using NETRC, see “NETRC Data Set” on page 14.

Chapter 3. Using Remote Execution Clients in an OE Environment

OE Remote Execution Protocol (OE REXEC) is a remote execution client that you can use to execute a command on a remote host and receive the results on the local host.

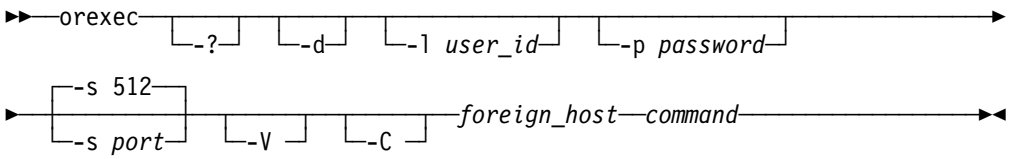
To use OE REXEC, you must have the OE REXEC daemon (orexecd) running on the remote host. The OE REXEC client passes the user name, password, and command to the OE REXEC daemon. The daemon provides automatic logon and user authentication, depending on the parameters that you set. If the user ID or password is incorrect, the server enters a loop.

The OE REXEC (orexec) Command—Execute a Command on the Remote Host

Purpose

Use the OE REXEC (orexec) command to execute a command on the remote host and receive the results on the local host.

Format



Note: Enter the orexec parameters -d, -l, -p, and -s in lowercase letters because they are case-sensitive. The *user_id* and *password* parameters may be case-sensitive, depending on the operating system of the remote host.

Parameters

-?
Displays the help message.

-d
Activates debug tracing.

-l *user_id*
Specifies the user ID on the foreign host.

-p *password*
Specifies the password for the user ID on the foreign host.

-s *port*
Specifies the TCP port number of the OE REXECD server on the foreign host. The default is port number 512.

-V
Writes the name and the APAR level to syslog.

-C
Changes all messages to uppercase.

foreign_host
Specifies the name or IP address of the foreign host to which you are sending the orexec command. Specify the foreign host by its host name or IP address.

command
Specifies the command that is sent to the foreign host. The command is composed of one or more words. Coding is assigned after checking the prefixed parameters (-l, -p, -s) and assigning the remaining string as the command. The command you specify must not require a response from you to complete. OE REXEC cannot interact with you after you enter data in the command format.

Examples

Use the orexec command to execute a command on a remote host:

```
orexec -l user28 -p user28 -s 512 mvs1 lista

MVS TCP/IP OREXEC V3R2
SYS1.HELP
GIM.SGIMCLS0
DSN230.DSNCLIST
USER.CLIST
BUILD.CLIST
SYS1.HRFCLST
USER28.ORSHD5.JOB00160.D0000103.?
```

Usage

If you omit the *user_id*, the *password*, or both when entering the orexec command, the system prompts you to supply the parameters.

Part 2. System Administrator Applications

Chapter 4. Monitoring the TCP/IP Network in an OE Environment

This chapter describes how to use TCP/IP commands to obtain information from the network.

- The OE NETSTAT (onetstat) command provides information about the status of the local host.
- The OE PING (oping) command determines the accessibility of a foreign node.
- The Display command provides information about the status of the TCP/IP images.
- The OE RPCINFO (orpcinfo) command displays the servers registered to portmapper.
- The OE Traceroute (otracert) command lets you debug network problems.

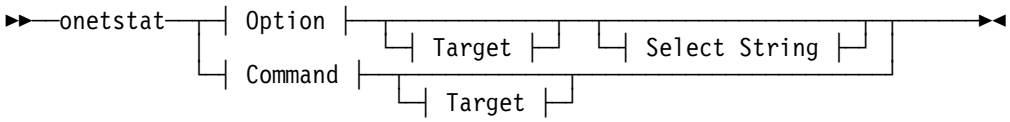
Using the OE NETSTAT (onetstat) Command — Display Local Host Information

Purpose

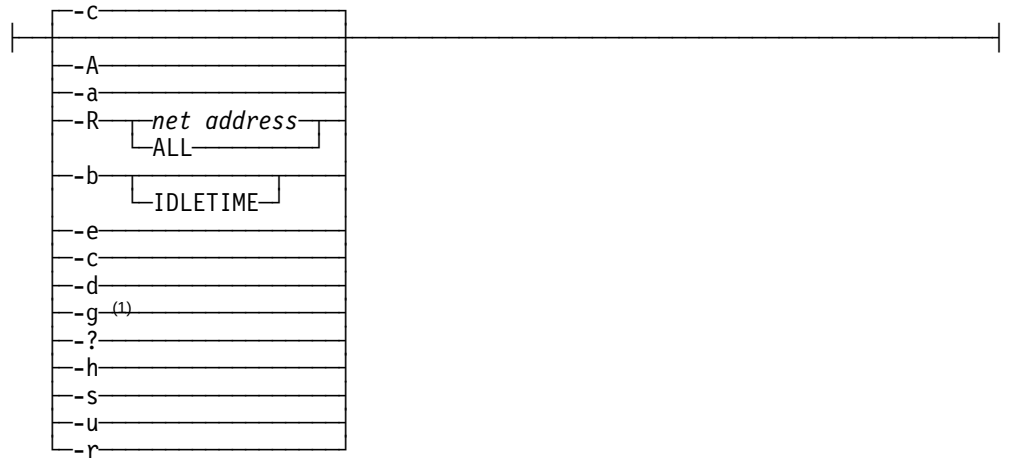
The OE NETSTAT (onetstat) command displays the network status of the local host, including information about TCP/IP connections, network clients, gateways, and devices. OE NETSTAT also drops connections for users who have the MVS.VARY.TCPIP.DROP statement defined in their RACF profile.

Use the onetstat command to display the network status of the local host.

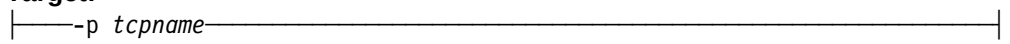
Format



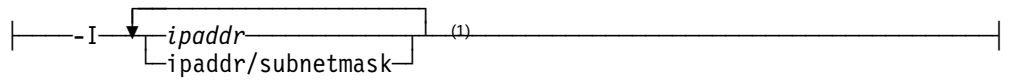
Option:



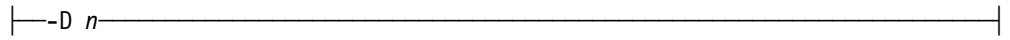
Target:



Select String:



Command:



Note:
1 -I is only valid with -g.

Parameters

-A

Provides detailed information about TCP/IP connections. This option is useful for debugging the TCP/IP address space.

For more information about maintaining the TCP/IP address space, see *OS/390 TCP/IP OpenEdition Configuration Guide*.

-a

Specifies that all information for all connections, including recently closed ones, is displayed.

-R

Queries the ARP cache information.

net address
Queries the ARP cache for a given address.

ALL
Queries all ARP cache entries.

-b

Displays the byte-count information about each connection. This option is most useful when used in conjunction with the redirect function ">".

The following information is given for each connection:

- Client name
- Bytes sent on the connection
- Bytes received on the connection
- Local port
- Foreign socket
- State

IDLETIME
Displays the preceding byte-count information plus the idle time for each connection.

The idle time is displayed in the following format:

```
hours:minutes:seconds
```

-e

Provides the following information about each client:

- Client's authorization
- Elapsed time since the client was last used.

-c

Provides the following information about each active TCP/IP connection. An active connection is a connection that is not in the *closed* or *time-wait* state.

- Client name
- Client ID
- Local IP address
- Foreign IP address
- Connection state

-c is the default parameter.

A connection progresses through a series of states during its lifetime. The following are the possible states for a TCP connection.

State	Description
LISTEN	Waiting for a connection request from any remote TCP and port
SYN-SENT	Waiting for a matching connection request after having sent a connection request
SYN-RECEIVED	Waiting for a confirming connection request acknowledgment after having both received and sent a connection request
ESTABLISHED	Represents an open connection; this is the normal state for the data transfer phase of the connection
FIN-WAIT-1	Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request
FIN-WAIT-2	Waiting for a connection termination request from the remote TCP
CLOSE-WAIT	Waiting for a connection termination request from the local user

CLOSING	Waiting for a connection termination request acknowledgment from the remote TCP
LAST-ACK	Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP

For more information about the TCP connection states, see RFC793.

-d

Displays the following information about devices and defined links in the TCP/IP address space:

- Device name
- Device type
- Device number
- Link name
- Link type
- Status of link

Status	Description
Starting	A START of the device has been issued by the operator, and TCP/IP has been sent an Activation request to the data link control (DLC) layer.
Sent SETUP	DLC has acknowledged TCP/IP's Activation request, and TCP/IP has requested DLC to perform the initial I/O sequence with the device.
Connecting	DLC has accepted the Initial I/O Sequence request.
Connecting2	The control connection for a CLAW device has been established, and the second connection (on which IP traffic us carried) is being established.
Negotiating	The initial I/O sequence with the device is complete, and TCP/IP is performing additional link-layer initialization.
Ready	The initialization sequence with the device is complete. The device is now ready.
Sent CLEAR	A STOP of the device has been issued by the operator, and TCP/IP has sent a Deactivation request to DLC.
Deactivated	DLC has performed the first stage of an orderly device deactivation.
Not active	The device is not active. (The device has never been started, or has been stopped after having been started.)

- Net number

  This field is significant only for links on LCS and CTC devices.
- Queue size

  This field is significant only for links on LCS devices.
- Number of bytes received
- Number of bytes transmitted
- BSD parameters
- Packet trace settings

Notes:

1. There is no link related information, packet trace settings and BSD parameters displayed for a device which has no link defined.
2. The packet trace setting will only be displayed when it is defined and set to on.
3. The LOOPBACK devices and links are displayed.

-D *n*

Drops the TCP/IP connection specified by *n*. You can determine the connection number from the -c column in the onetstat -c display. If you drop the server's *passive open* connection, the server immediately reissues the open request.

You can use this parameter only if your RACF profile contains the MVS.VARY.TCPIP.DROP definition.

-g

Provides the following information about each gateway:

- 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

-?

Provides help information for the onetstat parameters.

-h

Displays the HOME list. The IP address, link name, and primary interface information are displayed for each entry in the list.

For more information about the home list, see *OS/390 TCP/IP OpenEdition Configuration Guide*.

select_string

Specifies a character string that is used to limit the option responses .

-I *ipaddr*
-I *ipaddr/subnetmask*

Provides the response of -g on the specified IP address *ipaddr* or *ipaddr/subnetmask*. If *subnetmask* is not specified, the default of 255.255.255.255 is used.

Note: When filtering -g responses on a specified *ipaddr*, the DEFAULT and DEFAULTNET routes will always be displayed as they apply to all ip addresses.

You can enter up to six select_strings and each select_string can be up to 16 characters long. If specified, select_string must be the last parameter on the onetstat command line.

-s

Displays information about each client using the socket interface. When you specify the command `onetstat -s`, information about the client using the socket interface is displayed along with information about the sockets and associated connections owned by the client.

The following is a list of the information displayed after invoking the -s parameter:

Name	Displays the client's address space name.
Subtask	Displays the subtask identifier. The subtask identifier is combined with the address space name to produce a unique identifier for the client. • For socket programs written in the C language, the EBCDIC hexadecimal representation of an address within the program is used as the subtask identifier.
Type	Displays one of the following socket types: • `Stream` for stream (TCP) sockets • `Dgram` for datagram (UDP) sockets
Bound to	Indicates the address and port to which the socket is bound or not bound (unbound). The output is in the format `internet address..bound port` where `internet address` is the address to which the socket is bound and `bound port` is the port number to which the socket is bound. Unbound TCP and UDP sockets are not displayed by onetstat -c.
Connected to	Displays the address and port to which the socket is connected or not connected.
State	Displays the TCP connection state for TCP sockets or `UDP` for UDP sockets.
Conn	Displays the client identifier which is a unique number assigned by TCP/UDP stack to uniquely identify a particular socket entity.

-p *tcpname*

Displays detailed information about the specified TCP/IP address space. You can use -p *tcpname* with any other onetstat parameter to find out information about the specified TCP/IP address space; however, this parameter works only for TCP/IP address spaces having the same version as onetstat.

The *tcpname* is an 8-byte uppercase character file system type name. The file system name matches the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTEM statement that defined this Physical File System in the BPXPRM parameter library (PARMLIB) member.

-u

Provides the date and time that TCP/IP was started.

-r

Displays routing information in a standard fashion:

Destination	The address of a destination host or network
Gateway	The gateway used in forwarding packets
Flags	The state of the route: U The route is up. H The route is to a host rather than to a network. G The route is a gateway.

D	The route was created dynamically by a redirect.
Reference count	The current number of active users for the route.
Interface	The link name for the route.

Usage

- The time displayed in the header for each option is local time. The time field displayed in options -A, -b, -e, and -u is Greenwich mean time (GMT).
- The *tcpname* specified in -p option has to be an uppercase character string.
- Unspecified addresses are displayed as an asterisk (*).
- Port numbers are displayed numerically. A socket is displayed as an IP address followed by 2 periods (..) and a port number.

Examples

This section contains examples of the response that is displayed as a result of issuing the onetstat command with each parameter. To help you find the examples, here is a list of the onetstat parameters and the page number where you will find the example:

onetstat -A: Displays information about TCP/IP connections. The client TCPCLIE1 in the following example shows information about a TCP connection, and the client UDPCLIE1 shows information about a UDP connection.

```
onetstat -A

 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

 Client Name: TCPCLIE1               Client Id: 10002
 Local Socket: 9.67.113.10..2021     Foreign Socket: 9.67.43.62..621
   Last Touched:       3:35:05         State:              Listen
   RcvNxt:             0               SndNxt:             513488700
   ClientRcvNxt:       0               ClientSndNxt:       513488701
   InitRcvSeqNum:      0               InitSndSeqNum:      513488700
   CongestionWindow:   65535
   IncomingWindowNum:  0               OutgoingWindowNum:  0
   SndWl1:             0               SndWl2:             0
   SndWnd:             0               MaxSndWnd:          0
   SndUna:             513488700       rtt_seq:            0
   MaximumSegmentSize: 536             SlowStartThreshold: 65535
   BackoffCount:                       Precedence:         Routine
   Round-trip information:
     SmoothTripTime:   0.000           SmoothTripVariance: 1.500
   ReXmt:              0               ReXmtCount:         0
   SockOpt:            0               TcpTimer:           0
   TcpSig:             0               TcpSel:             0
   TcpDet:             0               TcpPol:             0
   No pending TCP-receive
 ----
 Client name: UDPCLIE1              Client id: 10003
 Local Socket: *..2025              Foreign Socket: *..*
   Last Touched:       3:35:05
   Precedence:         Routine
   BytesIn:                          BytesOut:
   DgramIn:                          DgramOut:
   MaxSendLim:                       MaxRecvLim:
   SockOpt:
 ----
```

onetstat -a: Displays either closed or time-wait connections, whereas the -c and -b options default to display only active TCP/IP connections:

```
onetstat -a

 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

 User Id  Conn   Local Socket        Foreign Socket     State
 ---- --  ----   ----- ------        ------- ------     -----
 TSUSER1  10020  *..2150              *..*              Listen
 TSUSER2  10010  *..2151              *..*              Listen
 PORTMP3  10035  *..2220              *..*              Listen
 PORTMP3  21002  *..2221              *..*                UDP
```

onetstat -R: Queries the ARP cache for a given address or all ARP cache entries:

```
onetstat -R 9.67.112.25

 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

Querying ARP cache for address 9.67.112.25
Link: TR1              IBMTR: 10005A0019F5
Route info: 0000
```

onetstat -b: The onetstat -b command displays byte-count information about each connection:

```
 onetstat -b

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

03/15/1996         MVS TCP/IP Real Time Network Monitor
User Id  B Out      B In       L Port  Foreign Socket          State
-------- ---------- ---------- ------- --------------          -------
TSUSER1  1234567890 1234567800 2150    *..*                    Estab
TSUSER2  2345670000 1234560000 2151    *..*                    Listen
PORTMP3           0          0 2221    *..*                     UDP

Connections displayed: 3
```

You can redirect the preceding screen to a file by using the redirect function (>) in the following format:

```
 onetstat -b > byteinfo
```

The file byteinfo is created in your home directory with the following contents:

```
MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

03/15/1996         MVS TCP/IP Real Time Network Monitor
User Id  B Out      B In       L Port  Foreign Socket          State
-------- ---------- ---------- ------- --------------          -------
TSUSER1  1234567890 1234567800 2150    *..*                    Estab
TSUSER2  2345670000 1234560000 2151    *..*                    Listen
PORTMP3           0          0 2221    *..*                     UDP

Connections displayed: 3
```

You can also use the onetstat -b IDLETIME command to display the idle time for each connection:

```
 onetstat -b IDLETIME

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

03/15/1996         MVS TCP/IP Real Time Network Monitor
User Id  B Out  B In   L Port  Foreign Socket          State   Idle
-------- ------ ------ ------- --------------          ------- --------
TSUSER1  ****** ****** 2150    *..*                    Estab   5:36:25
TSUSER2  234567 123456 2151    *..*                    Listen  5:36:25
PORTMP3       0      0 2221    *..*                     UDP    5:36:24

Connections displayed: 3
```

The asterisks (******) in the B Out or B In field indicate that the byte count value is more than 6 digits. In this case, if you want to know the exact byte count value, you should issue the onetstat -b again.

onetstat -e: Displays information about clients:

```
onetstat -e

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

  Current Clients:

  Client: USER1
  Authorization: None
  Last Touched:  4:01:17

  Client: USER2
  Authorization: None
  Last Touched:  4:01:17
```

onetstat -c: Displays information about active TCP/IP connections:

```
onetstat -c

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

  User Id  Conn   Local Socket       Foreign Socket     State
  ---- --  ----   ----- ------       ------- ------     -----
  TSUSER1  10020  *..2150            *..*               Listen
  TSUSER2  10010  *..2151            *..*               Listen
  PORTMP3  21002  *..2221            *..*                UDP
```

onetstat -d: Displays information about devices and defined links in the TCP/IP address space:

```
onetstat -d

  MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

  DevName: CTCCC12345       DevType: CTC         DevNum: 0CC2
    LnkName: LOCAL123         LnkType: CTC       Status: Not Active
      NetNum: 1   QueSize: 0  ByteIn: 00000000   ByteOut: 000000000
    BSD Routing Parameters:
      MTU Size: 00000576         Metric: 00000000
      DestAddr: 9.67.113.22      SubnetMask: 255.255.0.0
    Packet Trace Setting:
      Protocal: *         TrRecCnt: number    PckLength: ABBREV size
      SrcPort:   *        DestPort: *
      IpAddress: *                  Subnet: ipaddressmask

  DevName: LCS1      DevType: LCS       DevNum: 0CC4
    LnkName: TR1       LnkType: TR        Status: Not Active
      NetNum: 0   QueSize: 0   ByteIn: 00000000  ByteOut: 000000000
    BSD Routing Parameters:
      MTU Size: 00000000         Metric: 00000000
      DestAddr: 0.0.0.0          SubnetMask:  0.0.0.0

    LnkName: ETH1      LnkType: ETH      Status:  Active
      NetNum: 0   QueSize: 0   ByteIn: 00000000   ByteOut: 000000000
    BSD Routing Parameters:
      MTU Size: 00000000          Metric: 00000000
      DestAddr: 0.0.0.0           SubnetMask:  0.0.0.0
    Packet Trace Setting:
      Protocal: TCP      TrRecCnt:  number     PckLength: FULL
      SrcPort: *         DestPort:  *
      IpAddress: ipaddress          Subnet:    *

 DevName:  LCS3      DevType: LCS       DevNum: 0CC8
```

onetstat -D: Drops the TCP connection specified by *n*:

```
onetstat -c

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

 User Id  Conn   Local Socket       Foreign Socket     State
 ---- --  ----   ----- ------       ------- ------     -----
 TSUSER1  10020  *..2150            *..*               Listen
 TSUSER2  10010  *..2151            *..*               Listen
 PORTMP3  10032  *..2220            *..*               Listen
 PORTMP3  21002  *..2221            *..*                UDP

onetstat -D 10032

Connection successfully dropped

onetstat -c

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

 User Id  Conn   Local Socket       Foreign Socket     State
 ---- --  ----   ----- ------       ------- ------     -----
 TSUSER1  10020  *..2150            *..*               Listen
 TSUSER2  10010  *..2151            *..*               Listen
 PORTMP3  21002  *..2221            *..*                UDP
```

onetstat -g: Displays information about gateways:

```
onetstat -g
 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56
 Known gateways:

 NetAddress      FirstHop        Link    Pkt Sz  Subnet Mask     Subnet Value
 ----------      --------        ----    ------  -----------     ------------
 Default         9.67.113.1      TR2     576     <none>
 9.0.0.0         <direct>        TR2     2000    0.255.255.128   0.67.113.0
 9.67.116.16     <direct>        CTCD06  4000    HOST
```

onetstat -?: Displays help information for the onetstat parameters:

```
onetstat -?

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56
Usage: onetstat < Option | Command > < Target > < Select >
Option:
-A  -  Everything about a connection
-a  -  TCP/IP connections, include TIME-WAIT and CLOSED connections
-b  -  Current connection display
-c  -  Active TCP/IP connection (Default option)
-d  -  Devices and links
-e  -  Current clients
-g  -  Current known gateways
-h  -  Home address list
-R  -  Query ARP table or entry information
-r  -  Display routing information in a standard fashion
-s  -  Socket interface users and their sockets
-u  -  Date and time tcpip was last started
-?  -  onetstat information list
Target option:
-p  -  Displays detailed information about the specified TCPIP
       address space
Select-String:
-I     For -g, select information on the specified IP address
Command:
-D  -  Drop a TCP/IP connection
```

onetstat -h: Displays the HOME list:

```
onetstat -h
MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56
 Home address list:
 Address           Link             Flg
 -------           ----             ---
 9.67.113.29       TR2              P
 9.67.116.15       CTCD06
```

onetstat -s: Provides information about each client using the socket interface:

```
onetstat -s

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56
Socket interface status:

Type    Bound to             Connected to          State   Conn
----    --------             ------------          -----   ----
Name: PORTMP3     Subtask: 00015D10
Dgram   *..2700              Not connected         UDP     01014
Stream  *..2701              *..*                  Listen  01001
Name: PORTS3      Subtask: 000185C8
Dgram   *..1027              Not connected
Stream  *..1025              *..*                  Listen  01005
Stream  9.67.112.25..1025    9.67.112.25..1026     Estab   01008
Stream  9.67.112.25..1025    9.67.112.26..1024     Estab   01011
Stream  9.67.112.25..1025    9.67.112.26..1036     Estab   01003
Name: PORTC3      Subtask: 00022810
Stream  9.67.112.25..1026    9.67.112.25..1025     Estab   01010
Name: PORTC3      Subtask: 006602D0
Dgram   *..1033              Not connected         UDP     01015
Stream  *..1027              *..*                  Listen  01012
Stream  9.67.112.25..1027    9.67.112.26..1037     Estab   01000
Name: PORTC3      Subtask: 007652D0
Dgram   *..1037              Not connected         UDP     01018
Stream  *..1028              *..*                  Listen  01013
```

onetstat -p: This command displays detailed information about the specified TCPIP job. The output will format in different ways, depending on the specific command that was used, but it will return information about the TCPIP address space that was specified on the command.

The following example shows information about active TCP/IP connections for the TCPIP address space TCPIP33X.

```
onetstat -p TCPIP33X

 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPIP33X          12:34:56

User Id  Conn  Local Socket            Foreign Socket            State
-------  ----  ------------            --------------            -----
TSUSER5  10010 *..33333                *..*                      Listen
CFTPSERV 10040 *..5097                 *..*                      Listen
MISCSRV  10031 *..7                    *..*                      Listen
MISCSRV  10051 *..9                    *..*                      Listen
```

The following example shows an onetstat -p command issued for the TCPIP address space TCPIP33 other than the default TCPIP address space TCPV3. The name of the TCPIP address space must be included in the command.

```
 onetstat -R  9.67.112.25 -p TCPIP33

 MVS TCP/IP onetstat V3R3      TCPIP Name: TCPIP33          12:34:56
 Querying ARP cache for address 9.67.112.25
 Link: TR1               IBMTR: 10005A0019F5
    Route info: 0000
```

onetstat -u: Displays the date and time that TCP/IP was started:

```
 onetstat -u

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56
 Tcpip started at 09:08:15 on 08/06/1996
```

onetstat -r: Displays routing information:

```
onetstat -r

MVS TCP/IP onetstat V3R3      TCPIP Name: TCPV3          12:34:56

Destination     Gateway         Flags  Refcnt  Interface
-----------     --------        -----  ------  ---------
9.67.116.16     9.67.113.61     UGH    000000  TR1
Default         9.255.255.10    UG     000001  TR2
9.0.0.0         0.255.255.12    UH     000000  CTCD06
9.67.113.10     9.67.113.43     U      000004  CTCD05
```

Defining the onetstat Command Search Path

If the TCPIPjobname is not specified with the -p option, then onetstat uses the OE service _iptcpn() to retrieve the resolver-supplied TCPIPjobname. The _iptcpn() uses the following search order to locate the resolver configuration data set or file:

1. If the environment variable RESOLVER_CONFIG has been defined, the resolver uses the value of this environment variable as the name of the MVS data set or HFS file to access the resolver configuration data.

   The syntax for a MVS data set name is:

   ```
   "//'mvs.dataset.name'":
   ```

   The syntax for an HFS file name is:

   ```
   "/dir/subdir/file.name"
   ```

2. /etc/resolv.conf
3. Any MVS data set is pre-allocated to a DD-name of SYSTCPD. The use of this technique is discouraged because of restrictions for DD-name allocation during fork() processing.
4. userid.TCPIP.DATA for TSO/E or jobname.TCPIP.DATA for a batch request
5. SYS1.TCPPARMS(TCPDATA)

   For more about this dataset, see MVS/ESA Planning: OpenEdition MVS and the Accessing OS/390 OpenEdition MVS from the Internet.

   Two important keywords in the resolver file are DATASETPREFIX and TCPIPjobname. The value assigned to DATASETPREFIX will determine the high-level qualifier (hlq). The hlq is then used in the search order for the other configuraton files. If no DATSETPREFIX keyword is found in the resolver configuration dataset or file, a default of TCPIP is used.

   The value assigned to TCPIPjobname is used as the name of the TCP/IP image stack with which onestat attempts to establish a connection.

Using the OE PING (oping) Command—Send an Echo Request

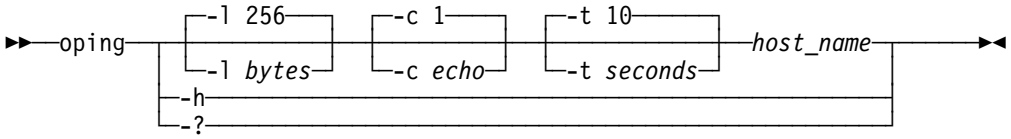
Purpose

The OE PING (oping) command sends an echo request to a foreign node (remote node) to determine if the computer is accessible.

When a response to an oping command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and the TCPIP address space.

Use the oping command to determine the accessibility of the foreign node.

Format



Parameters

host_name
Specifies the local or remote host to which you want to send the echo request. The host name is either a character-string name or the IP address in the standard format of the host.

-l *bytes*
Sets the number of bytes of the echo request. If *bytes* is not specified, an error occurs. If you do not specify the -l parameter, the default of 256 is used. The number of bytes must be between 8 and 32768.

-c *echo*
Sets the number of echo requests that are sent to the host. If you do not specify the -c parameter, the default of 1 is used. If *echo* is not specified, an error occurs. The *echo* value must be between 0 and $2^{31} - 1$, which is 2 147 483 647. If *echo* is 0, the oping command sends echo requests continually. To stop the oping command, see note 1 in the following section, "Usage."

-t *seconds*
Sets the number of seconds that the oping command waits for a response. If you do not specify the -t parameter, the default of 10 seconds is used. If *seconds* is not specified, an error occurs. The number of *seconds* must be between 1 and 100.

-h or -?
Provides help information about the oping command. You cannot place the -h or -? parameter on the oping command line with other parameters.

Usage

- To stop or interrupt the oping command, press **Ctrl** **c** . The interrupt key can be changed by using OMVS ESCAPE command for OE TSO shell, or the stty command for RAW shell. For more information about OMVS and stty commands, see the OpenEdition MVS Command Reference.
- You can place more than one parameter on the oping command line; however, the -h or -? parameter is an exception and cannot be placed on the oping command line with other parameters.

OE PING (oping) command return codes

The following is a list of the return codes generated by the oping command:

Code	Description
0	Response

4	No response
12	Socket API failure
100	Incorrect parameter

When a response to an oping command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and TCP/IP address space.

Resolving OE PING (oping) Command Problems

A host may fail to respond even after several oping commands for any of the following reasons:

- The host is not listening to the network.
- The host is inoperative, or some network or gateway leading from the user to the host is inoperative
- The host is slow because of activity.
- The packet is too large for the host.

The echo request sent by the oping command does not guarantee delivery. More than one oping command should be sent before you assume that a communication failure has occurred.

Use additional oping commands to communicate with other hosts in the network to determine the condition that is causing the communication failure. However, you should know the network topology to determine the location of the failure. Issue the oping commands in the following order until the failure is located.

1. Send an oping command to your local host.

   A successful oping command sent to a different host on the same network as the original host suggests that the original host is down, or is not listening to the network.

2. Send an oping command to a host other than your local host on your local network.
3. Send an oping command to each intermediate node that leads from your local host to the remote host, starting with the node closest to your local host.

   If you cannot get echoes from any host on that network, the trouble is usually somewhere along the path to the remote hosts. Direct an oping command to the gateway leading to the network in questions. If the oping command fails, continue to test along the network from the target, until you find the point of the communication breakdown.

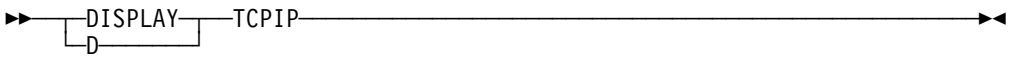
Using the DISPLAY TCP/IP Command for Client Information

Purpose

The DISPLAY TCPIP operator console command displays the status of the current TCP/IP images.

This is the general format of the DISPLAY command used to display the status of the current TCP/IP images:

Format



Examples

```
  User:   D TCPIP
System:   EZAOP501 TCPIP STATUS REPORT
           COUNT  TCPIP NAME  VERSION  STATUS
           -----  ----- ----  -------  ------
               1  TCPV32      V3R2     ACTIVE
               2  TCPV33      V3R3     ACTIVE
           *** END TCPIP STATUS REPORT ***
          EZAOP41I 'DISPLAY TCPIP' COMMAND COMPLETED SUCCESSFULLY
```

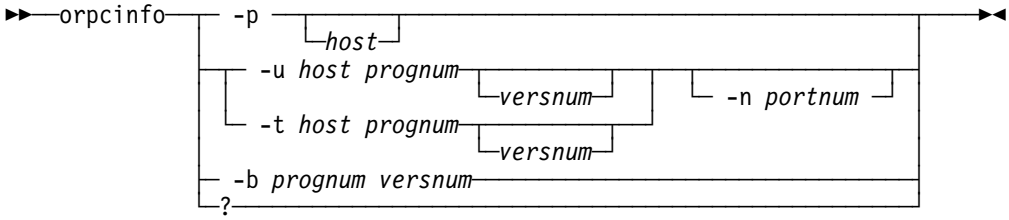
Displays the status of all current images of TCP/IP.

Using the OE RPCINFO (orpcinfo) Command—Display Server Information

Purpose

Use the orpcinfo command to display the servers that are registered and operational with any portmapper on your network. The orpcinfo command makes a remote procedure call (RPC) to an RPC server and displays the results.

Format



Parameters

-p *host*

Queries the portmapper on the specified host and prints a list of all registered RPC programs. If *host* is not specified, the system defaults to the local host name. For more information about how the local host name is defined, see the section on configuring TCP/IP in *OS/390 TCP/IP OpenEdition Configuration Guide*.

-n *portnum*
Specifies the port number to be used for the -t and -u options in place of the port number that is given by the portmapper.

-u *host prognum versnum*
Sends an RPC call to procedure 0 of *prognum* on the specified host using UDP, and reports whether a response is received. The variable *prognum* is the name or number of the RPC program.

-t *host prognum versnum*
Sends an RPC call to procedure 0 of *prognum* on the specified host using TCP, and reports whether a response is received.

-b *prognum versnum*
Sends an RPC broadcast to procedure 0 of the specified *prognum* and *versnum* using UDP, and reports all hosts that respond.

-?
Specifies the command help.

Usage

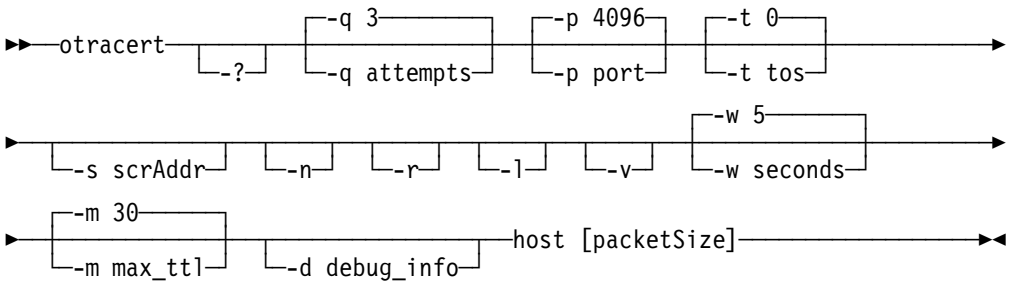
The version number is required for the -b parameter. If a version is specified, OE RPCINFO attempts to call that version of the specified program. If a version is not specified, OE RPCINFO prints error information.

Using the OE Traceroute (otracert) Command—Debug Network Problems

Purpose

The OE Traceroute (otracert) command is useful for debugging various network problems. The OE Traceroute function sends UDP requests with varying TTL (time to live) values and then waits for the routers between the local and remote hosts to send TTL-exceeded messages.

Format



Parameters

-?
Specifies the command help.

-q *attempts*
Specifies the number of times that a probe is sent with the same time-to-live value. This number reflects the total probe transmision (success or failure) per time-to-live increment. The range is 1-255. The default is 3.

-p *port*
Specifies the starting port number. The range for valid values is 4096-60000. The default is 4096.

-t *tos*
Specifies the type-of-service (tos) in the probe packets. The range for valid values is 0-255. The default is 0.

-s *scrAddr*
Specifies the source IP address. You must specify this address as an IP number and not a host name. On hosts with more than one IP address, you can set the source address to the IP address for another one of this machine's interface addresses.

-n
Specifies the print hop address. This address is numeric and saves a nameserver address-to-name lookup for each gateway on the path.

-r
Bypasses the normal routing tables and sends information directly to a host in an attached network. If the host is not in an adjacent network, an error is returned. You can use this option to ping a local host through an interface that has no active route.

-l
Specifies the time-to-live value in each received packet. This value can be used to help detect asymmetric routing.

-v
Specifies that additional information is to be sent.

-w *seconds*
Specifies how long to wait for a response. The range for valid values is 1–255. The default is 5 seconds.

-m *max_ttl*
Specifies the maximum time to live.

-d *debug info*
Specifies that extra messages and other debugging information are to be printed.

Examples

In these examples, an asterisk (*) represents a lost packet.

- The second hop in this example does not send TTL-exceeded messages.

```
otracert cyst.watson.ibm.com
Trace route to CYST.WATSON.IBM.COM (9.2.91.34)
Use escape C sequence to interrupt
1 (9.67.22.2) 67 ms 53 ms 60 ms
2 * * *
3 (9.67.1.5) 119 ms 83 ms 65 ms
4 (9.3.8.14) 77 ms 80 ms 87 ms
5 (9.158.1.1) 94 ms 89 ms 85 ms
6 (9.31.3.1) 189 ms 197 ms *
7 * * (9.31.16.2) 954 ms
8 (129.34.31.33) 164 ms 181 ms 216 ms
9 (9.2.95.1) 198 ms 182 ms 178 ms
10 (9.2.91.34) 178 ms 187 ms *
```

- Sometimes packets are lost (hops 6,7, and 10).

```
otracert 129.35.130.09
Trace route to 129.35.130.09 (129.35.130.9)
Use escape C sequence to interrupt
1 (9.67.22.2) 61 ms 62 ms 56 ms
2 * * *
3 (9.67.1.5) 74 ms 73 ms 80 ms
4 (9.3.8.1) 182 ms 200 ms 184 ms
5 (129.35.208.2) 170 ms 167 ms 163 ms
6 * (129.35.208.2) 192 ms !H 157 ms !H
```

- The network was found, but no host was found. The packet could not route to that network.

```
otracert 129.45.45.45
Trace route to 129.45.45.45 (129.45.45.45)
Use escape C sequence to interrupt
1 (9.67.22.2) 320 ms 56 ms 71 ms
2 * * *
3 (9.67.1.5) 67 ms 64 ms 65 ms
4 (9.67.1.5) 171 ms !N 68 ms !N 61 ms !N
```

- Traceroute uses the site tables for inverse name resolution rather than the domain name server. If a host name is found in the site table, it will be printed along with its IP address.

```
otracert EVANS
Trace route to EVANS (129.45.45.45)
Use escape C sequence to interrupt
1 BART (9.67.60.85) 20 ms 56 ms 71 ms
2 BUZZ (9.67.60.84) 55 ms 56 ms 54 ms
3 EVANS (9.67.30.25) 67 ms 64 ms 65 ms
```

Usage

- The range of port numbers OE Traceroute uses is normally not valid but can be changed if the target host is using non-standard UDP port.

Chapter 5. Managing TCP/IP Network Resources with OE SNMP

This chapter describes how to use the OE Simple Network Management Protocol (osnmp) command and what support the OE TCP/IP SNMP agent and subagent provide.

SNMP defines an architecture that consists of network management stations (SNMP managers), network elements (hosts and gateways), and network management agents and subagents, which perform the information management functions.

SNMP allows managers and agents to communicate network management information.

The OE SNMP (osnmp) command is a management application that can be used to monitor and control network elements.

Note: Appendix C, “Management Information Base (MIB) Objects” on page 227 lists the MIB objects supported by the OE SNMP agent and subagent and the maximum access allowed. A limited number of variables are supported with read-write access. If an SNMP SET (write) is attempted against a variable for which the maximum access is read-only, an error code will be returned. For an SNMPv1 request, this error code will be noSuchName. For an SNMPv2 request, the error code will be noAccess or notWritable.

Using the OE SNMP (osnmp) Command

Purpose

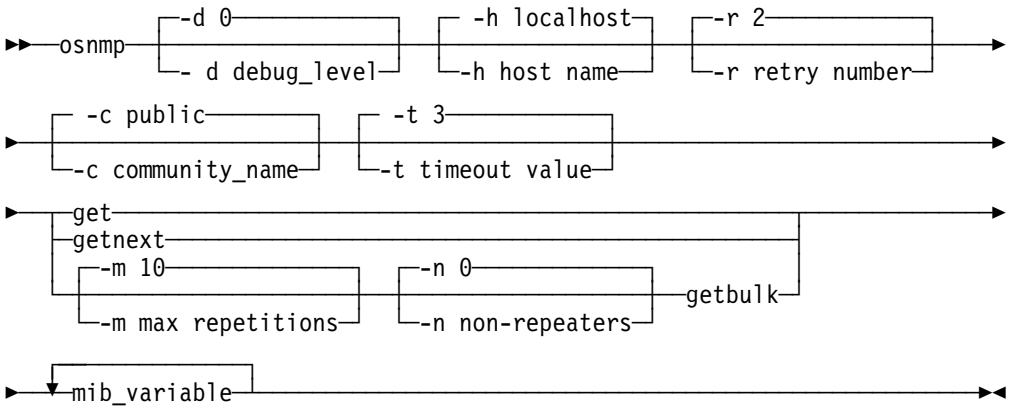
The osnmp command provides SNMP manager function to query SNMP agents for network management information.

Use the osnmp command to issue SNMP requests to agents and to process SNMP responses returned by agents. The osnmp commands supports issuance of SNMPv1, SNMPv2c, and SNMPv2u requests.

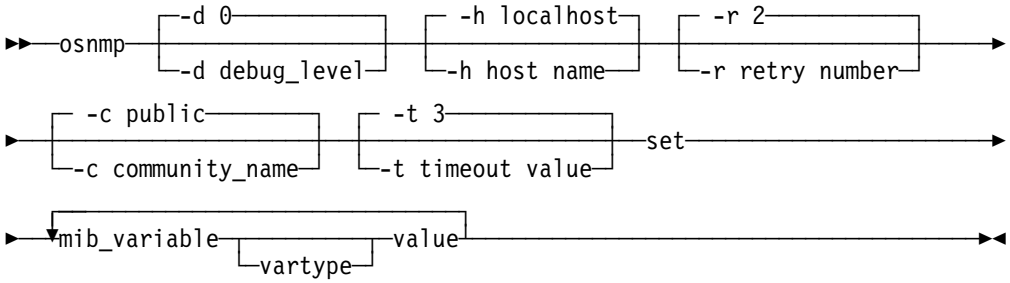
Note: The OS/390 TCP/IP OE SNMP agent only supports SNMPv1 and SNMPv2c requests.

Format

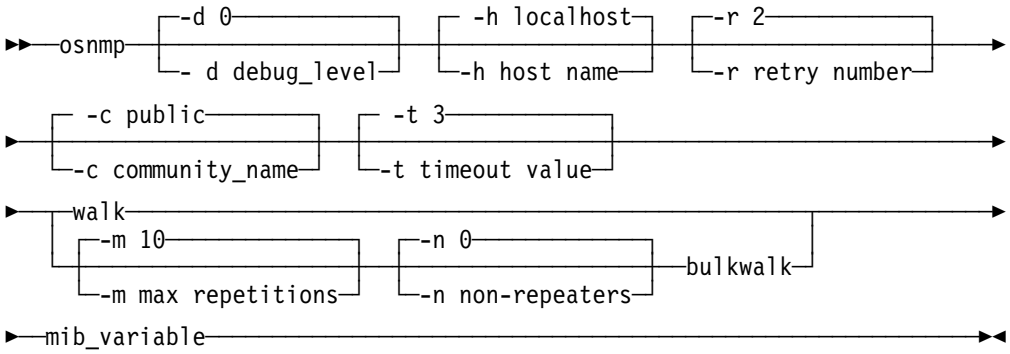
Getting MIB Variables



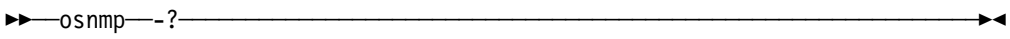
Setting the MIB Variables



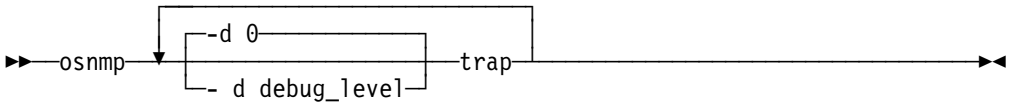
Walking the MIB Tree



Displaying osnmp Help



Receiving a Trap



Parameters

-c community_name
Specifies the community name used to access the specified variables at the destination SNMP agent. If you do not specify a community name, the default name is *public*.

Note: Community names are case sensitive.

-d debug_level
Specifies the debug level. The default level is 0, which means no debug. The higher the debug level, the more messages that display. The debug levels are 0–4.

-h host name
Specifies the destination host to which you want to send a request. This can be either an internet protocol address, a host name, or a winsnmp name in the /etc/snmpv2.conf configuration file. If you do not specify a host, the default is your local host.

-r retry number
Specify the maximum number of times to retry the command if it timed out. The default is 2.

-t timeout value
Specifies the amount of time (in seconds) that the osnmp command waits for a reply from the SNMP agent. The default is 3 seconds.

-m max repetitions
Only applies to GETBULK and BULKWALK requests. It will be ignored if the function request is not a GETBULK or BULKWALK. Maximum repetitions is the number of lexicographic successors to be returned for each variable binding pair after the first "-n number" successors. For example. starting with successor "-n number"+1, return "-m number" of successors for each variable binding pair. The default is 10.

-n non-repeaters
Only applies to GETBULK or BULKWALK requests. It will be ignored if the function request is not a GETBULK or BULKWALK. Non-repeaters is the number of variable binding pairs (name/value), starting with the first, for which only a single successor is returned. The default is 0.

-? Displays help information.

mib_variable
Specifies the MIB object, using its object descriptor (textual name), object identifier using ASN.1 notation, or a combination of the two. When used with **walk** and **bulkwalk**, this is the MIB object prefix. A prefix can be any leading portion of the complete object identifier.

vartype
The type of value being set. To complete an SNMP SET request, the SMI_type must be known. If no type is specified, OE SNMP searches first the /etc/mibs.data file and then the compiled MIB to determine the type. If the variable is not found, an error is returned. If a vartype is specified, the vartype takes precedence over any type that may be assigned in the MIB. The vartype and value must be compatible. For example, if you specify a type of “number”

and a value of “foo,” an error will be returned because "foo" is not a number. Vartype is not case sensitive. Valid variable types are:

- bitstring
- counter
- counter32
- counter64
- display or displaystring
- integer
- integer32
- ipaddress
- gauge
- gauge32
- nsapaddress
- null
- objectidentifier or OID
- octetstring
- opaque
- opaqueascii
- timeticks
- uinteger

value
Specifies the value to be set by the SET function. If white space is needed in the value, you must enclose the value in double quotes ("). If you want to set a variable to a value that is also a type, you must specify the type.

SNMP request types

get
Sends a request to an SNMP agent for a specific management information base (MIB) variable. OE SNMP then waits for a response or times out.

getbulk
A single GETBULK performs the same function as a series of GETNEXTs. Obtains the value of the variables in the MIB tree specified by the OID or MIB variable name.

getnext
Sends a request to an SNMP agent for the next MIB variable that lexicographically follows the mib_variable specified. OE SNMP then waits for a response or times out .

set
Sends a request to an SNMP agent to set a specific MIB variable. OE SNMP then waits for a response or times out.

trap
Listens on well-known port 162 for SNMP traps and displays trap information when they occur. The osnmp trap function continues to listen for traps until the process is killed or cancelled .

walk
Issues a **getnext** request for a specified prefix, then continues to issue **getnext** requests for as long as there are variables that match the specified prefix. A prefix can be any leading portion of the complete object identifier.

bulkwalk
Issues a GETBULK request for a specified prefix, then continues to issue GETBULK requests for as long as there are variables that match the specified prefix.

Usage

- SET is only supported if the agent or subagent managing the MIB the command is directed at supports SET.
- GETBULK and BULKWALK are SNMPv2 functions. If the target agent only supports SNMPv1, the target agent will ignore your request. As a result, your request will timeout.
- The function keywords are not case sensitive. The "-" options and variable names and values are case sensitive.
- In order to issue the osnmp trap command, you must be in superuser mode. This is required in order to bind to well-known port 162. If you are not in superuser mode, you will receive error EZZ3301I Error return from bind() : EDC5111I Permission denied.
- An osnmp command that is not authenticated (by using an acceptable community name) will time out.
- osnmp command uses three configuration files, /etc/mibs.data, /etc/snmpv2.conf, and /etc/pw.src.cli. For information about these files, see *OS/390 TCP/IP OpenEdition Configuration Guide*.
- The osnmp command supports sending both SNMPv1 and SNMPv2 requests. The file osnmp uses to determine whether it should send an SNMPv1 or and SNMPv2 request is the snmpv2.conf file. This file is only used for sending SNMPv2 requests and must be used if sending SNMPv2 requests. If the target specified via the -h parameter matches a winsnmp name in the snmpv2.conf file, osnmp sends an SNMPv2 request. Otherwise, osnmp sends an SNMPv1 request.

Examples

- Gets the MIB variable

```
osnmp get sysName.0
1.3.6.1.2.1.1.5.0 = MVS SNMP
```

- Gets the next MIB variable

```
osnmp getnext sysName.0
1.3.6.1.2.1.1.6.0 = Planet Earth, North America, USA
```

- Sets the MIB variable

```
osnmp set sysName.0  "MVSX SNMPv2 Agent"
1.3.6.1.2.1.1.5.0 = MVSX SNMPv2 Agent
```

- Walks the MIB tree

```
osnmp walk system
1.3.6.1.2.1.1.1.0 = Sysname: OS/390 Nodename: MVSVIC96 Release: 04.00 Version: 01
Machine: 9021
1.3.6.1.2.1.1.2.0 = 1.3.6.1.4.1.2.3.13
1.3.6.1.2.1.1.3.0 = 124900
1.3.6.1.2.1.1.4.0 = Fred Smith, system programmer
1.3.6.1.2.1.1.5.0 = MVS150
1.3.6.1.2.1.1.6.0 = Building 123, Floor 2, Raleigh, North Carolina
1.3.6.1.2.1.1.7.0 = 0
1.3.6.1.2.1.1.8.0 = 400
1.3.6.1.2.1.1.9.1.2.1 = 1.3.6.1.4.1.2.11.7.1
1.3.6.1.2.1.1.9.1.2.2 = 1.3.6.1.4.1.2.11.7.2
1.3.6.1.2.1.1.9.1.3.1 = TCPIP for MVS Agent Capabilities
1.3.6.1.2.1.1.9.1.4.1 = 0
1.3.6.1.2.1.1.9.1.4.2 = 400
```

- Shows a GET on the MIB object, myName, which is defined in the /etc/mibs.data file with the object ID for the sysName object.

```
osnmp get myName.0
1.3.6.1.2.1.1.5.0 = MVSX SNMPv2 Agent
```

OE SNMP Remote PING

OE SNMP remote PING is a function of the SNMP subagent that gives an SNMP manager the ability to obtain the round-trip response time for an ICMP echo request message (PING) from an SNMP agent to a destination IP address.

The SNMP remote PING function is a valuable tool in an enterprise network that provides centralized management services because it gives a third party (SNMP manager) system the ability to request that a PING operation be performed on a remote SNMP agent MVS system.

For example, if there are three hosts (A, B, and C) as shown in Figure 7 on page 205, you can obtain the response time between the 2 remote hosts. In this example, your host is running the SNMP manager function (Host A), Host B is running the SNMP agent function, and Host C is some arbitrary remote host. The standard PING function allows Host A to obtain the round-trip response time from A to B and from A to C, but not from B to C. With the SNMP remote PING function on the MVS SNMP agent, Host A can obtain the round-trip response time from B to C.

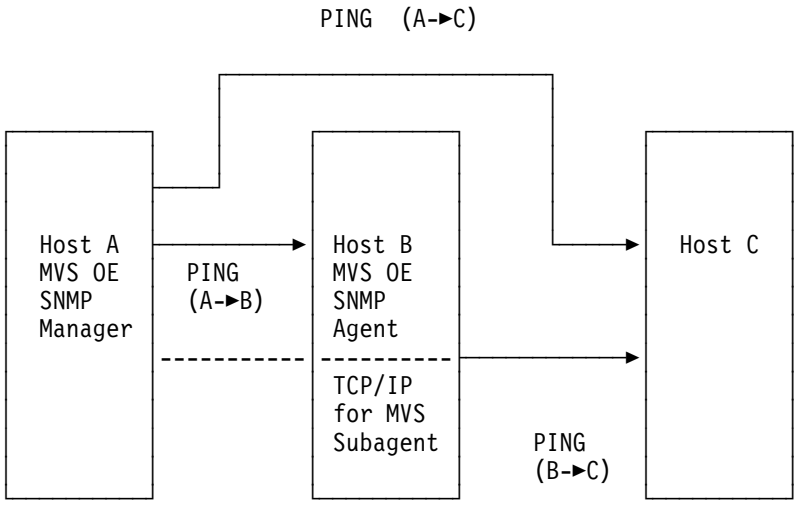
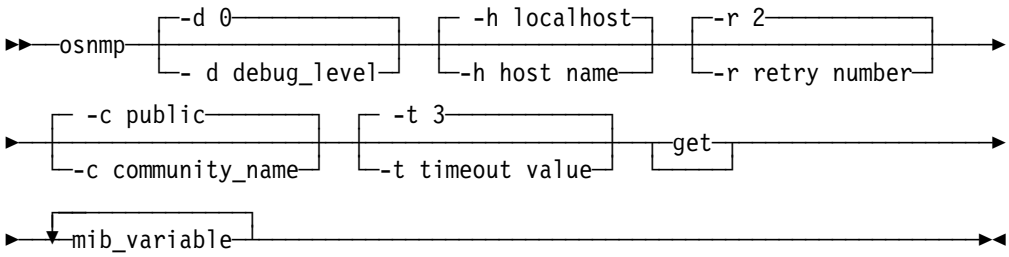


Figure 7. SNMP Remote PING Function

With the SNMP remote PING function, you can specify the size of the packet, in bytes, that is sent in the ICMP echo request message and the time period, in seconds, to wait for that ICMP echo request message to return from the requested destination address.

Format

To send a remote PING command, use the osnmp GET command in the following format:



Parameters

-d debug_level
Specifies the debug level. The default level is 0, which means no debugging. The higher the debug level, the more messages that display. The debug levels are 0 through 4.

-h host name
Specifies the destination host to which you want to send a request. This can be either an internet protocol address, a host name, or a winsnmp name in the snmpv2.conf configuration file. If you do not specify a host, the default is your local host.

-r retry number
Specify the maximum number of times to retry the command if it timed out. The default is 2.

-c community_name
Specifies the community name used to access the specified variables at the destination SNMP agent. If you do not specify a community name, the default name is *public*.

Note: Community names are case sensitive.

-t timeout value
Specifies the amount of time (in seconds) that the osnmp command waits for a reply from the SNMP agent. The default is 3 seconds.

get
Sends a request to an SNMP agent for a specific management information base (MIB) variable. OE SNMP then waits for a response or times out.

mib_variable
Specifies one or more MIB variable names to be retrieved. You can specify the names in textual form or in ASN.1 notation. Each variable name must consist of four instances, separated by dots (.), as in the following example:

```
osnmp -h host_name get ibmMvsRPingResponseTime.packet_size.time_out.ip_address
```

The following list describes the instances of *var_name*:

Instance Description

ibmMvsRPingResonseTime
Specifies the remote PING command

packet_size
Specifies the packet size of the PING request

time_out
Specifies the time-out value, in seconds, for the PING request

ip_address
Specifies the IP address of the remote host that receives the PING request

Example

The following is an example of using GET to perform a remote PING:

```
osnmp -h mvs1 -c mvs150 get ibmMvsRPingResponseTime.2048.5.9.37.33.175
```

where:

```
host_name = mvs1

community_name  = mvs150

mib_variable  = ibmMvsRPingResponseTime.2048.5.9.37.33.175  where:

          packet size     = 2048 bytes
          time-out        = 5 seconds
          ip_address      = 9.37.33.175
```

The expected response is as follows:

```
1.3.6.1.4.1.2.6.19.2.2.1.1.1.4.2048.5.9.37.33.175=33
```

The variable value in the previous example is a positive value (33) indicating a successful response. The variable number, when positive, is the round-trip response time, in milliseconds, from the OE SNMP agent host system to the requested destination IP address.

The variable value can be a negative integer indicating that a failure has occurred. A negative integer is a result of the SNMP agent or TCP/IP MVS subagent detecting either an internal error, an incorrect MIB instance format (f.e.a.b.c.d), an ICMP echo request time-out, an incorrect packet size value (f), an incorrect time-out value (e) or an incorrect destination IP address (a.b.c.d). See Table 19 for a description of the what the variable value can represent.

Table 19. SNMP Get Command Responses for Variable Value

Description	Returned value	Condition	Valid input
Round-Trip Response Time	>0 (milliseconds)	Success	N/A
Internal error	-1	Failure	N/A
ICMP echo request timed out	-2	Failure	N/A
Incorrect packet size (f)	-4	Failure	0, 16 - 4096 (bytes)
Incorrect time-out (e)	-5	Failure	0, 3 - 15 (seconds)
Unknown destination IP address (a.b.c.d)	-6	Failure	Dotted decimal IP address
Incorrect MIB instance format (f.e.a.b.c.d)	-7	Failure	packet size.timeout.IP address in dotted decimal format

Note: The packet size (f) and the time-out (e) in the *mib_variable* part of the osnmp get command can have a value of 0 indicating that the default values of 16 bytes and 10 seconds, respectively, are used.

The MIB variable `ibmMVSRPingResponseTime` is defined to access the OE SNMP Remote PING function and is included in the MVS Enterprise Specific MIB . This variable exists under the SNMP subtree `ibmRemotePingEntry`.

The ASN.1 for the SNMP subtree `ibmRemotePingEntry` is `1.3.6.1.4.1.2.6.19.2.2.1.1.1`, and the ASN.1 for the SNMP Remote PING variable `ibmMvsRPingResponseTime` is `1.3.6.1.4.1.2.6.19.2.2.1.1.1.4`.

Network Manager Considerations

The following sections describe:

- The data that the SNMP agent and subagent support
- Details on how devices are reflected in the interface table
- Retrieval of ATM interface data

SNMP MIB Support

The OS/390 TCP/IP OE SNMP agent and subagents support for MIB variables is defined in several files shipped with the product. These files are installed into the HFS in the /usr/lpp/tcpip/samples directory:

- mvstcpip.caps

  Contains the formal SNMPv2 definition of the MIBs supported by the SNMP agent and subagent shipped with TCP/IP.

- mvstcpip.mi2

  Contains the formal SNMPv2 syntax (SMI) of our IBM MVS Enterprise Specific MIB extension.

- mvstcpip.mib

  Contains the formal SNMPv1 syntax (SMI) of our IBM MVS Enterprise Specific MIB extension.

- samib.mi2

  Contains the SAMIB MIB objects

- rfc1592b.mi2

  Contains the additional information that expands the implementation of RFC 1592 in the OS/390 TCP/IP OE environment.

Interface Layering

In the SNMP framework the most fundamental MIB Table is the Interface table. RFC 1213 provides the latest SNMPv1 format of this table. OS/390 TCP/IP OE SNMP has also implemented the ifMib Internet Draft. For more information, see Appendix C, “Management Information Base (MIB) Objects” on page 227 for a list of supported ifMIB objects. The ifMib expands what is defined by RFC 1213:

- Extends the Interface table, ifTable and its ifEntry definition, through the ifXTable and its ifXEntry definition.
- Provides a new table, ifStackTable, that shows how interfaces are layered.

SNMP's interface layering implementation can best be explained by the following example. In this example, the first DEVICE statement is for an Open Systems Adapter 2 (ATM OSA-2) port . OS/390 TCP/IP OE does not support an ATM Port as a transport facility, but the SNMP subagent can interface with an ATM OSA-2 adapter through the Open Systems Adapter Support Facility (OSA/SF) for support of ATM Management. See the *OS/390 TCP/IP OpenEdition Configuration Guide* for information on configuring SNMP for ATM Management support.

Given the following DEVICE and LINK Profile statements:

```
DEVICE OSA1 ATM
LINK ATMPORT1 ATM OSA1
DEVICE LCS1 LCS 100
LINK TR1 IBMTR 0 LCS1
LINK ETH1 ETHERNET 0 LCS1
```

The following interface entries would be created:

ifIndex	ifType	Description
1	53 (propVirtual)	LOOPBACK device
2	24 (softwareLOOPBACK)	LOOPBACK link
3	53 (propVirtual)	ATM Device OSA1
4	1 (other)	ATM Link ATMPORT1
5	49 (aal5)	ATMPORT1's aal5 layer
6	37 (atm)	ATMPORT1's atm layer
7	53 (propVirtual)	LCS Device LCS1
8	9 (iso88025TokenRing)	LCS Link TR1
9	6 (ethernetCsmacd)	LCS Link ETH1

ifTypes are assigned by the Internet Assigned Numbers Authority (IANA) to indicate the type of interface. In OS/390 TCP/IP OE, a DEVICE has a corresponding interface entry with its LINKs defined as interface entries stacked below it. The ifStackTable is used to reflect the relationship between interfaces. Reference the ifMib for a detailed explanation of how the ifStackTable is used to reflect interface relationships. Essentially a DEVICE is stacked above its LINKs. Its ifEntry and ifXEntry counters reflect the sum of its LINKs.

In the previous example , a LOOPBACK DEVICE and LINK ifEntry were created when the links were not explicitly defined. TCP/IP automatically generated these entries. The ifEntry and ifXEntry counters can be retrieved from either the LOOPBACK LINK or DEVICE interface entry to determine LOOPBACK activity. The counters shown for a DEVICE interface entry will equal that of a subordinate LINK when there is only one LINK defined for the DEVICE.

When an ATM LINK is defined, two subordinate interface entries are created below it, AAL5 and ATM. AAL5 and ATM are UNI defined layers that exist physically in an ATM Port. The ifEntry and ifXEntry counters relect traffic though the port. Since OS/390 TCP/IP OE does not support an ATM Port as a transport facility, the ifEntry and ifXEntry counters at the ATM DEVICE and LINK layers will be zero. The AAL5 and ATM Layer counters will reflect Native ATM traffic.

ATM Considerations

ATM MIB Objects

OS/390 TCP/IP OE SNMP provides several MIB tables that form the basis of ATM Management support:

- osasfChannelTable

  An entry in this table will be created for every ATM DEVICE interface. Each ATM DEVICE statement represents one ATM OSA-2 adapter card. Externally through SNMP this table will be indexed by the ifIndex of the ATM DEVICE.

  Note: There is a maximum of 12 OSA ATM adapter cards per 390 host. Initially an ATM OSA-2 card can support only 1 ATM port.

- osasfPvcTable

  This table will be created at the same time that the osasfChannelTable is created. Indexing is by the ifIndex of the AAL5 layer and pvcName. One entry is created for every PVC defined for a given ATM Port. There is a limit of 256 PVCs per port per channel.

- osasfPortTable

An entry in this table is created for every ATM Link interface. Indexing is by the ifIndex of the AAL5 interface layer.

- atmInterfaceConfTable from RFC 1695 MIB

  One entry in this table is created for every ATM Link interface. It is however indexed by the ifIndex of AAL5 interface entry that corresponds to the ATM Link layer.

- Interface Table Data

  ifTable and ifXTable data will be retrieved from OSA/SF for the AAL5 and ATM interfaces subordinate to an ATM LINK interface. Both AAL5 and ATM layer data is retrieved at the same time.

ATM Port IP address Assignment

OS/390 TCP/IP OE SNMP provides a method for assigning an IP address to the ATM Port. The OSA ATM Port reports the IP address as its atmfMyIpNmAddress as specified by the ATM Forum User-Network Interface (UNI) Specification. UNI defines an Interim Local Management Interface (ILMI) layer that provides a MIB that can be accessed directly over an ATM Network via an SNMP Request.

To specify an IP address for an ATM Port, you can use the HOME statement. Once an IP address is set, the ATM Port remembers the IP address and it does not have to be reset. Set the IP address through the HOME statement at a TCP/IP instance running in the same MVS image as the customizing OSA/SF. After setting the IP address remove the definition to save system resources because the ATM OSA Port remembers its address. For information about the HOME statement, see the*OS/390 TCP/IP OpenEdition Configuration Guide*.

ATM Trap notification from OSA/SF

Asynchronous events will be forwarded from OSA/SF to SNMP subagents. These events will be converted to traps and sent to the OE SNMP agent associated with the TCP/IP instance receiving the notification for forwarding. The traps supported for ATM Management are:

- ATM Port enabled - LinkUp Trap
- ATM Port disabled - LinkDown Trap
- Permanent Virtual Circuit (PVC) creation - ibmMvsOsasfAtmPvcCreate Trap
- Permanent Virtual Circuit (PVC) deletion - ibmMvsOsasfAtmPvcDelete Trap

The LinkUp and LinkDown traps for ATM Ports will be generated by the Subagent for the AAL5 layer interface.

An ibmMvsAtmOsasfAtmPvcCreate notification is generated when OSA/SF sends an asynchronous notification to a subagent that a PVC was created for a given ATM Port. An ibmMvsAtmOsasfAtmPvcDelete notification is generated when a PVC is deleted.

Note: A subagent discards any notification received for an ATM PORT that is not properly defined through the DEVICE and LINK statements.

Part 3. Appendixes

Appendix A. Specifying Data Sets and Files

This appendix describes the file-naming formats for the MVS, AIX, UNIX, VM, OS/2, and AS/400 operating systems. Examples of each format are provided to show how the files appear to a TCP/IP user who is logged on to the different operating systems.

MVS Data Sets

FTP subcommands can require a data set or file name. The format used to name a data set depends on the host system. Some systems limit the length of a data set name, and some systems are case-sensitive.

Data set names in MVS consist of one or more names, called qualifiers, each from 1 to 8 characters long, that are delimited from one another by periods.

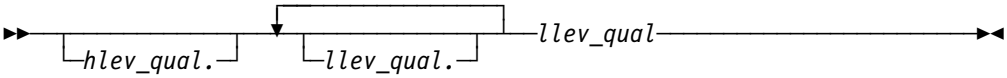
The leftmost qualifier in the data set name is the high-level qualifier.

The rightmost qualifier in the data set name is the low-level qualifier. Partitioned data sets may be further qualified with a member name in the rightmost position.

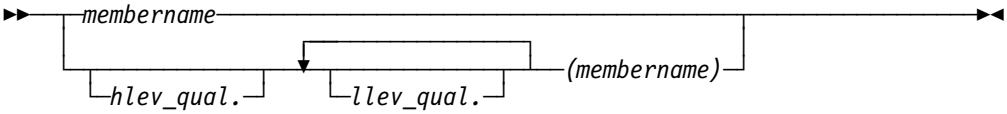
Qualifiers lying between them are called intermediate-level qualifiers.

For example, in the data set name `dog.bulldog.winston` `dog` is the high-level qualifier, `bulldog` is the intermediate-level qualifier, and `winston` is the low-level qualifier.

Specify TSO sequential data sets in the following format:



Specify TSO partitioned data sets in the following format:



hlev_qual

Specifies the high-level qualifier of the data set. The default is the current working directory. If you specify this parameter, the complete data set name must be enclosed within single quotation marks (').

llev_qual

Specifies the low-level qualifier of the data set. You must specify this qualifier for sequential data sets.

membername

Specifies the member name of a partitioned data set (PDS). You must include parentheses around *membername* only when you also specify *llev_qual* or *hlev_qual*.

Sequential Data Sets

A sequential data set is a single file that can be allocated with any record length specified. The naming requirements for a sequential data set on an MVS host are minimal, and most of the requirements apply to any data set name under MVS.

The naming requirements for a sequential data set are:

- No part of the name can start with a numeric.
- No part of the name can be more than 8 characters in length.
- Each part of the name is separated by a period.
- A sequential data set name can have a minimum of 2 and a maximum of 44 characters.
- If single quotation marks are not used when specifying the data set name, the MVS system appends the current working directory as the first part of the name.

The following examples show the naming conventions for sequential data sets on an MVS host.

To access the sequential data set `KC00852.SEQ.NAMES`, the user, with the current working directory KC00852, enters one of the following:

```
'KC00852.SEQ.NAMES'
```

Or

```
SEQ.NAMES
```

Either of these formats is acceptable for accessing a sequential data set.

Partitioned Data Sets

A partitioned data set (PDS) is a group of files contained in a library. The individual files that make up a PDS are called members. You can access an entire PDS or any individual member of a PDS.

The naming requirements for a partitioned data set are:

- No part of the name can start with a numeric.
- No part of the name can be more than 8 characters in length.
- Each part of the name is separated by a period.
- If single quotation marks are not used when specifying the PDS name, the MVS system appends the current working directory as the first part of the name.

The difference between a sequential and partitioned data set specification is that the partitioned data set user accesses the directory of members in the PDS, and the sequential data set user accesses an individual file.

The following examples show the naming conventions for partitioned data sets on an MVS host.

To access the partitioned data set KC00852.PDS.NAMES, the user, with the current working directory 'KC00852', enters one of the following:

```
'KC00852.PDS.NAMES'
```

Or

```
PDS.NAMES
```

Either of these formats is acceptable to access a partitioned data set.

Note: You can use the special character asterisk (*) as a global name character (wild card) for pattern matching when you specify a data set name, with the following restrictions:

- The asterisk must be the last, or only, character specified for a level of qualifier.
- When the data set name is enclosed in quotation marks, you cannot use the asterisk as a wild card in the high-level qualifier of the data set name. Data set names not enclosed in quotation marks will use the setting of the current directory as the high-level qualifier.
- You can use the asterisk more than once in the complete data set name, but the asterisk must be the last character for each level of data set name qualifier.
- If you specify a member name, you cannot use an asterisk anywhere in the data set name.
- If you use an asterisk as all or part of the member name, you cannot use an asterisk anywhere else in the data set name.

To access an individual member of a PDS, the member name is entered in parentheses.

To access the member PROPER in the PDS KC00852.PDS.NAMES, the user, with the current working directory KC00852, enters one of the following:

```
'KC00852.PDS.NAMES(PROPER)'
```

Or

```
PDS.NAMES(PROPER)
```

Either of these formats is acceptable to access an individual member of a partitioned data set.

Transferring Data between Partitioned and Sequential Data Sets

When transferring data between partitioned and sequential data sets, ensure that the *local_file* and *foreign_file* parameters of the FTP subcommands are compatible with the type of data set you are transferring to or from. For example, if your local working directory is a partitioned data set and you want to GET the sequential file TEST.FILE1, you cannot issue the subcommand GET TEST.FILE1 to retrieve the file, because this subcommand by default tries use the local PDS member name TEST.FILE1, which is not a valid member name.

To keep the local and remote file names compatible with the type of data set used, either specify both the local and remote file names, for example:

```
GET TEST.FILE TESTFL1
PUT TESTPDS(FILE1) FILE1
```

or change the directory to the lowest level qualifier. For example, to transfer between the PDS 'USER14.TESTPDS(NAME1)' and the sequential data set 'USER17.SEQ.NAME1', LCD to 'USER14.TESTPDS' and CD to 'USER17.SEQ'; then you can enter `GET NAME1` to get 'USER17.SEQ.NAME1' as 'USER14.TESTPDS(NAME1)', or `PUT NAME1` to put 'USER14.TESTPDS(NAME1)' as 'USER17.SEQ.NAME1'.

Data Transfer Methods

You must use the appropriate transmission attributes to preserve the content and structure of the data when you transfer data sets or files between 2 hosts. Use the FTP MODE subcommand to specify how the bits of data are to be transmitted, and the FTP TYPE subcommand to define the way that data is represented during the data transfer.

See "MODE Subcommand—Set the Data Transfer Mode" on page 120 for information about the MODE subcommand, and "TYPE Subcommand—Set the Data Transfer Type" on page 168 for information about the TYPE subcommand.

TCP/IP only supports the data transfer of a data set or file structured as a continuous sequence of data bytes. This ensures that the correct record format is preserved across MVS hosts.

Table 20 shows how to set the transmission attributes for different host systems. IBM mainframe operating systems (VM or MVS) are identified as EBCDIC transfer types. Systems with ASCII storage are identified as ASCII transfer types. A text file of an ASCII transfer type contains standard, displayable characters; a carriage return (ASCII X'0D' and EBCDIC X'15'); and line feed characters (ASCII X'0A' and EBCDIC X'25'). A text file of an EBCDIC transfer type contains standard, displayable characters only. A binary file can contain any characters.

Table 20. Recommended Methods for Data Transfer

Transfer Between Host Types	Transfer Type	Mode
EBCDIC to EBCDIC — text data	**EBCDIC**	**Stream**
EBCDIC to EBCDIC — binary data	**EBCDIC**	**Block**
EBCDIC to ASCII — text data	**ASCII**	**Stream**
ASCII to EBCDIC — text data	**ASCII**	**Stream**
ASCII to EBCDIC — binary data	**Image (binary)**	**Stream**
ASCII to EBCDIC to ASCII — all data	**Image (binary)**	**Stream**

Note: The EBCDIC host is used for storage only. Data is not used on the EBCDIC host.

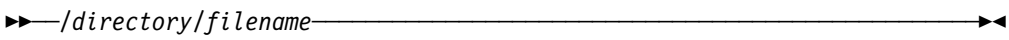
Transferring PDS Directory Information

When a PDS member is transmitted in block or compressed data transfer mode with a representation type of EBCDIC, the user data associated with the PDS member is also transferred to the directory on the target host, only when using an MVS client.

AIX and UNIX Files

For the Advanced Interactive Executive (AIX) and UNIX operating systems, data is stored in files. Related files are stored in a directory. OS/390 OE MVS files are UNIX files.

Specify AIX and UNIX files in the following format:



directory
Specifies a directory name. Directories contain the names of files, other directories, or both.

filename
Specifies a file name. It can be up to 14 characters long.

The complete name of an AIX and UNIX file contains the directory name and the file name. The following is an example:

```
/mailfiles/cooks
```

Where:

mailfiles Is the directory name.
cooks Is the file name.

In the AIX and UNIX operating systems, you specify the first slash (/) only when you begin at the root directory. If you are specifying a file in the current directory, enter only the file name. For example, if you are in the current directory *mailfiles* and you want to access the *cooks* file, specify:

```
cooks
```

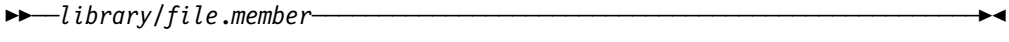
The directory name and file name can each be up to 14 characters. The AIX and UNIX operating systems distinguishes between uppercase and lowercase letters in file names.

A directory name and file name should not include characters such as backslash (\), ampersand (&), and period (.) that have a special meaning to the shell.

AS/400 Operating System

For the AS/400 operating system, data is stored in files.

Specify AS/400 files in the following format:



library
Is a library name. Libraries contain the names of programs, files, and commands.

file.member
Is the file name.

In the AS/400 operating system, files can have one or more members. Each file can consist of data records, source programs, or database definitions.

The FTP subcommand PUT is used to copy a local file member into a file at the remote host. The following is an example:

```
PUT PDS.DATA(MBR1)  LIB1/FILEA.MBR1
```

In this example, the PUT subcommand copies the file member `MBRA` in file `FILEA` into library `TCPA` at the local host to `MBRA` in `FILEA` in library `TCPB` at the remote host. If the member already exists at the remote host, it is overwritten.

OS/2 Files

OS/2 files are stored on a storage device such as a diskette or hard disk. A storage device is identified by a drive letter. Within the storage device, files can be assigned to a directory, which is a group of related files.

The complete name of an OS/2 file contains the storage device identifier, directory name, file name, and an extension. The following is an example of a complete name for an OS/2 file.

```
C:\WP\MAIL.LST
```

In this example, the device identifier is C: and the directory name is WP, which could be a group of word processing files. The file name is MAIL, and the file extension is .LST.

In the OS/2 operating system, the device identifier is a drive letter that is assigned by the file system, followed by a colon. The drive letter, such as A, B, or C, can represent either a physical device or a logical device. If a device identifier is not specified, the default is the device from which the system was booted or the current drive.

You assign the directory name, which consists of a character string of 1 to 8 characters, preceded by a backslash. A directory name is optional. If a directory name is not specified, the system searches the current directory.

Normally, you also assign the file name, which consists of a character string of 1 to 8 characters. A file name is required.

The file extension is a character string that consists of 1 to 3 characters, preceded by a period. A file extension is optional.

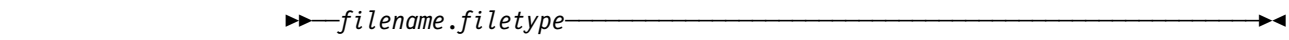
If you use an incorrect file name, the system gives you an error message.

The following are examples of different OS/2 file names that are valid.

```
TEMP
START.BAT
A:COMMAND.COM
C:ABC\ABCPC.HLP
```

VM Files

Data is stored in files on VM hosts. Specify VM files in the following format:



filename

Specifies the file name.

filetype

Specifies 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 not used in TCP/IP commands.

For example, if you want to specify a file named `accounts` with a file type `cprog` enter the following:

```
accounts.cprog
```

Where *filename* is `accounts` and *filetype* is `cprog`.

All VM file specifications are treated as if they are entered in uppercase. The file name and the file type consist of 1 to 8 alphanumeric characters. Other valid characters are ($), number sign (#), at sign (@), plus (+), hyphen (-), and underscore (_).

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

Appendix B. Capability Statement

This appendix includes the capability statement for OS390 TCP/IP OE. The capability statement defines the level of support that an agent provides to a MIB group. For example, a capability statement may indicate that some objects have restricted or augmented syntax or access levels. This information is in the HFS directory /usr/lpp/tcpip/samples. The file name is 'mvstcpip.caps'.

```
IBMTCPIPMVS-CAPS DEFINITION ::= BEGIN

IMPORTS
   enterprises,
   AGENT-CAPABILITIES
         FROM SNMPv2-SMI ;

   ibm               OBJECT IDENTIFIER ::= { enterprises 2 }
   ibmAgentCapabilities OBJECT IDENTIFIER ::= { ibm 11 }

ibmTcpIpMvsCaps MODULE-IDENTITY
     LAST-UPDATED "9612220000Z"
     ORGANIZATION "IBM TCP/IP MVS Development"
     CONTACT-INFO
         "        Kenneth White

          Postal: International Business Machines Corporation
                  P.O. Box 12195
                  Dept. G80/Bldg. 503
                  Research Triangle Park, NC 27709
                  USA

             Tel: +1 919 254 0102
             Fax: +1 919 254 4027

          E-mail: kennethw@vnet.ibm.com"
     DESCRIPTION
         "The IBM TCP/IP for MVS Products capabilities
          statements"
::= { ibmAgentCapabilities 7 }

ibmTcpIpMvsAgtCaps AGENT-CAPABILITIES
   PRODUCT-RELEASE  "IBM TCP/IP for MVS V3R3 SNMPv2 Agent"
   STATUS              current
   DESCRIPTION        "TCPIP for MVS Agent's Capabilities"

   SUPPORTS            SNMPv2-MIB
       INCLUDES        { systemGroup, snmpGroup, snmpSetGroup,
                         snmpBasicNotificationsGroup,
                         snmpCommunityGroup }
       VARIATION       coldStart
           DESCRIPTION "A coldStart trap is generated on all
                        reboots."

   SUPPORTS            DPI20-MIB
       VARIATION       dpiPathNameForUnixStream
           DESCRIPTION "This object was added the the dpiMib
                        defined by RFC1592 in order to support
                        AF_UNIX DPI connections."

   SUPPORTS            SUBAGENT-MIB

   ::= { ibmTcpIpMvsCaps 1 }

ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
   PRODUCT-RELEASE  "IBM TCP/IP for MVS V3R3 SNMPv2 DPI Subagent"
   STATUS              current
   DESCRIPTION        "TCPIP for MVS DPI Subagent's Capabilities"
```

```
SUPPORTS             IBMTCPIPMIV-MIB
    INCLUDES         { ibmRemotePingGroup, ibmTcpipMvsSystem,
                       ibmTcpipMvsAtmGroup, ibmTcpipMvsTcpGroup,
                       ibmTcpipMvsUdpGroup }

SUPPORTS             IF-MIB
    INCLUDES         { ifGeneralInformationGroup, ifStackGroup2,
                       ifPacketGroup, ifHCFixedLengthGroup }
    VARIATION        ifAdminStatus
        SYNTAX       INTEGER { up(1), down(2) }
        ACCESS        read-only
        DESCRIPTION  "Test mode (testing(3)) not supported. Don't
                      support SET operation. Currently read-only."
    VARIATION        ifOperStatus
        SYNTAX       INTEGER { up(1), down(2) }
        DESCRIPTION  "Information limited to up or down. Don't
                      support testing(3), unknown(4), dormant(5),
                      or notPresent(6) not supported."
    VARIATION        ifLastChange
        DESCRIPTION  "Use time that TCP/IP was started instead of
                      sysUpTime to calculate this value, since
                      sysUpTime represents time relative to the
                      agent's IPL not TCP/IP's."
    VARIATION        ifLinkUpDownTrapEnable
        SYNTAX       INTEGER { enabled(1) }
        ACCESS       read-only
        DESCRIPTION  "SET operation not supported. Currently
                      defaults to enabled."
    VARIATION        ifPromiscuousMode
        ACCESS        read-only
        DESCRIPTION  "Write access is not required, nor supported."
    VARIATION       ifAlias
        ACCESS       read-only
        DESCRIPTION  "Write access is not required, nor supported.
                      Will return a zero length octet string."
    VARIATION        ifStackStatus
        SYNTAX       INTEGER { active(1) } -- subset of RowStatus
        ACCESS       read-only
        DESCRIPTION  "Write access is not required, nor supported.
                      Only one enumerated values for the RowStatus
                      textual convention is supported."
    VARIATION        ifStackLastChange
        DESCRIPTION  "Not supported"
    VARIATION        ifTableLastChange
        DESCRIPTION  "Not supported"
    VARIATION        ifHCInOctets
        DESCRIPTION  "Supported only for AAL5 and ATM Interface
                      entries as required by the ifMib. Indexing
                      with a non-AAL5 or ATM ifIndex will result
                      in the return of noSuchInstance."
    VARIATION        ifHCOutOctets
        DESCRIPTION  "Supported only for AAL5 and ATM Interface
                      entries as required by the ifMib. Indexing
                      with a non-AAL5 or ATM ifIndex will result
                      in the return of noSuchInstance."

SUPPORTS             IP-MIB
```

```
        INCLUDES         { ipGroup, icmpGroup }
        VARIATION        ipForwarding
            ACCESS        read-only
            DESCRIPTION "write operation not supported."
        VARIATION        ipDefaultTTL
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaIfIndex
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaPhysAddress
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaNetAddress
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaType
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."

    SUPPORTS             RFC1354-MIB
        INCLUDES         { ipForwardGroup }
        VARIATION        ipForwardMask
             ACCESS   read-only
             DESCRIPTION "Write access not supported."
        VARIATION        ipForwardPolicy
             DESCRIPTION "Not used in this release. Will always return
                          a zero."
        VARIATION        ipForwardIfIndex
             ACCESS   read-only
             DESCRIPTION  "write access not supported."
        VARIATION        ipForwardType
             ACCESS   read-only
             DESCRIPTION  "write access not spported."
        VARIATION        ipForwardInfo
             ACCESS   read-only
             DESCRIPTION "write access not supported.
                          Will always return a zero"
        VARIATION        ipForwardNextHopAS
             ACCESS   read-only
             DESCRIPTION "write access not supported.
                          Will always return a zero."
        VARIATION       ipForwardMetric1
             ACCESS   read-only
             DESCRIPTION
                "An alternate routing metric  for  this  route."
        VARIATION       ipForwardMetric2
             ACCESS   read-only
             DESCRIPTION "not supported"
        VARIATION      ipForwardMetric3
             ACCESS   read-only
             DESCRIPTION "not supported"
        VARIATION     ipForwardMetric4
             ACCESS   read-only
             DESCRIPTION "not supported"
        VARIATION     ipForwardMetric5
             ACCESS    read-only
```

```
            DESCRIPTION "not supported"

    SUPPORTS             TCP-MIB
        INCLUDES         { tcpGroup }
        VARIATION        tcpConnState
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."

    SUPPORTS             UDP-MIB
        INCLUDES         { udpGroup }

    SUPPORTS             ATM-MIB
        INCLUDES         { atmInterfaceConfGroup }

    ::= { ibmTcpIpMvsCaps 2 }

END
```

Appendix C. Management Information Base (MIB) Objects

This appendix lists the objects defined by the Management Information Base (MIB) which are supported by the OMVS agent.

The object types are defined using the following fields:

Object Descriptor	A textual name for the object type, along with its corresponding OBJECT IDENTIFIER.
Object Identifier	The name for the object type, using ASN.1 notation.
Supported by	Support by the agent or subagent.
Defined by	The location of the description of the object. The generic traps supported by OS/390 TCP/IP OE are defined in RFC 1907. OS/390 TCP/IP OE provides support of two enterprise-specific traps which are defined in its Enterprise Specific MIB, mvstcpip.m12. The non-IEFT derived MIBs supported by OS/390 TCP/IP OE are installed in HFS at /usr/lpp/tcpip/samples HFS directory and consist of: • mvstcpip.mi2 • mvstcpip.mib (SNMPv1 version of mvstcpip.mi2) • samib.mi2 • rfc1592b.mi2
Access Allowed	Read-only, read-write, write-only, or not-accessible.

Object Types

Table 21 on page 228 shows the object types for the SNMP agent and subagents.

Table 21 (Page 1 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
sysDescr	1.3.6.1.2.1.1.1	Agent	RFC1907	R/O
sysObjectID	1.3.6.1.2.1.1.2	Agent	RFC1907	R/O
sysUpTime	1.3.6.1.2.1.1.3	Agent	RFC1907	R/O
sysContact	1.3.6.1.2.1.1.4	Agent	RFC1907	R/W
sysName	1.3.6.1.2.1.1.5	Agent	RFC1907	R/W
sysLocation	1.3.6.1.2.1.1.6	Agent	RFC1907	R/W
sysServices	1.3.6.1.2.1.1.7	Agent	RFC1907	R/O
sysORLastChange	1.3.6.1.2.1.1.8	Agent	RFC1907	R/O
sysORTable	1.3.6.1.2.1.1.9	Agent	RFC1907	N/A
sysOREntry	1.3.6.1.2.1.1.9.1	Agent	RFC1907	N/A
sysORIndex	1.3.6.1.2.1.1.9.1.1	Agent	RFC1907	N/A
sysORID	1.3.6.1.2.1.1.9.1.2	Agent	RFC1907	R/O
sysORDescr	1.3.6.1.2.1.1.9.1.3	Agent	RFC1907	R/O
sysORUpTime	1.3.6.1.2.1.1.9.1.4	Agent	RFC1907	R/O
ifTable	1.3.6.1.2.1.2.2	Subagent	ifMib Internet Draft	N/A
ifEntry	1.3.6.1.2.1.2.2.1	Subagent	ifMib Internet Draft	N/A
ifIndex	1.3.6.1.2.1.2.2.1.1	Subagent	ifMib Internet Draft	R/O

Table 21 (Page 2 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ifDescr	1.3.6.1.2.1.2.2.1.2	Subagent	ifMib Internet Draft	R/O
ifType	1.3.6.1.2.1.2.2.1.3	Subagent	ifMib Internet Draft	R/O
ifMtu	1.3.6.1.2.1.2.2.1.4	Subagent	ifMib Internet Draft	R/O
ifSpeed	1.3.6.1.2.1.2.2.1.5	Subagent	ifMib Internet Draft	R/O
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Subagent	ifMib Internet Draft	R/O
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Subagent	ifMib Internet Draft	R/O
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Subagent	ifMib Internet Draft	R/O
ifLastChange	1.3.6.1.2.1.2.2.1.9	Subagent	ifMib Internet Draft	R/O
ifInOctets	1.3.6.1.2.1.2.2.1.10	Subagent	ifMib Internet Draft	R/O
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Subagent	ifMib Internet Draft	R/O
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Subagent	ifMib Internet Draft	R/O
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Subagent	ifMib Internet Draft	R/O
ifInErrors	1.3.6.1.2.1.2.2.1.14	Subagent	ifMib Internet Draft	R/O
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Subagent	ifMib Internet Draft	R/O
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Subagent	ifMib Internet Draft	R/O
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Subagent	ifMib Internet Draft	R/O
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Subagent	ifMib Internet Draft	R/O

Table 21 (Page 3 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Subagent	ifMib Internet Draft	R/O
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Subagent	ifMib Internet Draft	R/O
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Subagent	ifMib Internet Draft	R/O
ifSpecific	1.3.6.1.2.1.2.2.1.22	Subagent	ifMib Internet Draft	R/O
ipForwarding	1.3.6.1.2.1.4.1	Subagent	RFC1213	R/O
ipDefaultTTL	1.3.6.1.2.1.4.2	Subagent	RFC1213	R/O
ipInReceives	1.3.6.1.2.1.4.3	Subagent	RFC1213	R/O
ipInHdrErrors	1.3.6.1.2.1.4.4	Subagent	RFC1213	R/O
ipInAddrErrors	1.3.6.1.2.1.4.5	Subagent	RFC1213	R/O
ipForwDatagrams	1.3.6.1.2.1.4.6	Subagent	RFC1213	R/O
ipInUnknownProtos	1.3.6.1.2.1.4.7	Subagent	RFC1213	R/O
ipInDiscards	1.3.6.1.2.1.4.8	Subagent	RFC1213	R/O
ipInDelivers	1.3.6.1.2.1.4.9	Subagent	RFC1213	R/O
ipOutRequests	1.3.6.1.2.1.4.10	Subagent	RFC1213	R/O
ipOutDiscards	1.3.6.1.2.1.4.11	Subagent	RFC1213	R/O
ipOutNoRoutes	1.3.6.1.2.1.4.12	Subagent	RFC1213	R/O
ipReasmTimeout	1.3.6.1.2.1.4.13	Subagent	RFC1213	R/O

Table 21 (Page 4 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ipReasmReqds	1.3.6.1.2.1.4.14	Subagent	RFC1213	R/O
ipReasmOKs	1.3.6.1.2.1.4.15	Subagent	RFC1213	R/O
ipReasmFails	1.3.6.1.2.1.4.16	Subagent	RFC1213	R/O
ipFragOKs	1.3.6.1.2.1.4.17	Subagent	RFC1213	R/O
ipFragFails	1.3.6.1.2.1.4.18	Subagent	RFC1213	R/O
ipFragCreates	1.3.6.1.2.1.4.19	Subagent	RFC1213	R/O
ipAddrTable	1.3.6.1.2.1.4.20	Subagent	RFC1213	N/A
ipAddrEntry	1.3.6.1.2.1.4.20.1	Subagent	RFC1213	N/A
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	Subagent	RFC1213	R/O
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	Subagent	RFC1213	R/O
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	Subagent	RFC1213	R/O
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	Subagent	RFC1213	R/O
ipAdEntReasmMaxSize	1.3.6.1.2.1.4.20.1.5	Subagent	RFC1213	R/O
ipNetToMediaTable	1.3.6.1.2.1.4.22	Subagent	RFC1213	N/A
ipNetToMediaEntry	1.3.6.1.2.1.4.22.1	Subagent	RFC1213	N/A
ipNetToMediaIfIndex	1.3.6.1.2.1.4.22.1.1	Subagent	RFC1213	R/O
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	Subagent	RFC1213	R/O

Table 21 (Page 5 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3	Subagent	RFC1213	R/O
ipNetToMediaType	1.3.6.1.2.1.4.22.1.4	Subagent	RFC1213	R/O
ipRoutingDiscards	1.3.6.1.2.1.4.23	Subagent	RFC1213	R/O
ipForward	1.3.6.1.2.1.4.24	Subagent	RFC1354	N/A
ipForwardNumber	1.3.6.1.2.1.4.24.1	Subagent	RFC1354	R/O
ipForwardTable	1.3.6.1.2.1.4.24.2	Subagent	RFC1354	N/A
ipForwardEntry	1.3.6.1.2.1.4.24.2.1	Subagent	RFC1354	N/A
ipForwardDest	1.3.6.1.2.1.4.24.2.1.1	Subagent	RFC1354	R/O
ipForwardMask	1.3.6.1.2.1.4.24.2.1.2	Subagent	RFC1354	R/O
ipForwardPolicy	1.3.6.1.2.1.4.24.2.1.3	Subagent	RFC1354	R/O
ipForwardNextHop	1.3.6.1.2.1.4.24.2.1.4	Subagent	RFC1354	R/O
ipForwardIfIndex	1.3.6.1.2.1.4.24.2.1.5	Subagent	RFC1354	R/O
ipForwardType	1.3.6.1.2.1.4.24.2.1.6	Subagent	RFC1354	R/O
ipForwardProto	1.3.6.1.2.1.4.24.2.1.7	Subagent	RFC1354	R/O
ipForwardAge	1.3.6.1.2.1.4.24.2.1.8	Subagent	RFC1354	R/O
ipForwardInfo	1.3.6.1.2.1.4.24.2.1.9	Subagent	RFC1354	R/O
ipForwardNextHopAS	1.3.6.1.2.1.4.24.2.1.10	Subagent	RFC1354	R/O

Table 21 (Page 6 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ipForwardMetric1	1.3.6.1.2.1.4.24.2.1.11	Subagent	RFC1354	R/O
ipForwardMetric2	1.3.6.1.2.1.4.24.2.1.12	Subagent	RFC1354	R/O
ipForwardMetric3	1.3.6.1.2.1.4.24.2.1.13	Subagent	RFC1354	R/O
ipForwardMetric4	1.3.6.1.2.1.4.24.2.1.14	Subagent	RFC1354	R/O
ipForwardMetric5	1.3.6.1.2.1.4.24.2.1.15	Subagent	RFC1354	R/O
icmpInMsgs	1.3.6.1.2.1.5.1	Subagent	RFC1213	R/O
icmpInErrors	1.3.6.1.2.1.5.2	Subagent	RFC1213	R/O
icmpInDestUnreachs	1.3.6.1.2.1.5.3	Subagent	RFC1213	R/O
icmpInTimeExcds	1.3.6.1.2.1.5.4	Subagent	RFC1213	R/O
icmpInParmProbs	1.3.6.1.2.1.5.5	Subagent	RFC1213	R/O
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	Subagent	RFC1213	R/O
icmpInRedirects	1.3.6.1.2.1.5.7	Subagent	RFC1213	R/O
icmpInEchos	1.3.6.1.2.1.5.8	Subagent	RFC1213	R/O
icmpInEchoReps	1.3.6.1.2.1.5.9	Subagent	RFC1213	R/O
icmpInTimestamps	1.3.6.1.2.1.5.10	Subagent	RFC1213	R/O
icmpInTimestampReps	1.3.6.1.2.1.5.11	Subagent	RFC1213	R/O
icmpInAddrMasks	1.3.6.1.2.1.5.12	Subagent	RFC1213	R/O

Table 21 (Page 7 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	Subagent	RFC1213	R/O
icmpOutMsgs	1.3.6.1.2.1.5.14	Subagent	RFC1213	R/O
icmpOutErrors	1.3.6.1.2.1.5.15	Subagent	RFC1213	R/O
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	Subagent	RFC1213	R/O
icmpOutTimeExcds	1.3.6.1.2.1.5.17	Subagent	RFC1213	R/O
icmpOutParmProbs	1.3.6.1.2.1.5.18	Subagent	RFC1213	R/O
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	Subagent	RFC1213	R/O
icmpOutRedirects	1.3.6.1.2.1.5.20	Subagent	RFC1213	R/O
icmpOutEchos	1.3.6.1.2.1.5.21	Subagent	RFC1213	R/O
icmpOutEchoReps	1.3.6.1.2.1.5.22	Subagent	RFC1213	R/O
icmpOutTimestamps	1.3.6.1.2.1.5.23	Subagent	RFC1213	R/O
icmpOutTimestampReps	1.3.6.1.2.1.5.24	Subagent	RFC1213	R/O
icmpOutAddrMasks	1.3.6.1.2.1.5.25	Subagent	RFC1213	R/O
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	Subagent	RFC1213	R/O
tcpRtoAlgorithm	1.3.6.1.2.1.6.1	Subagent	RFC1213	R/O
tcpRtoMin	1.3.6.1.2.1.6.2	Subagent	RFC1213	R/O
tcpRtoMax	1.3.6.1.2.1.6.3	Subagent	RFC1213	R/O

Table 21 (Page 8 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
tcpMaxConn	1.3.6.1.2.1.6.4	Subagent	RFC1213	R/O
tcpActiveOpens	1.3.6.1.2.1.6.5	Subagent	RFC1213	R/O
tcpPassiveOpens	1.3.6.1.2.1.6.6	Subagent	RFC1213	R/O
tcpAttemptFails	1.3.6.1.2.1.6.7	Subagent	RFC1213	R/O
tcpEstabResets	1.3.6.1.2.1.6.8	Subagent	RFC1213	R/O
tcpCurrEstab	1.3.6.1.2.1.6.9	Subagent	RFC1213	R/O
tcpInSegs	1.3.6.1.2.1.6.10	Subagent	RFC1213	R/O
tcpOutSegs	1.3.6.1.2.1.6.11	Subagent	RFC1213	R/O
tcpRetransSegs	1.3.6.1.2.1.6.12	Subagent	RFC1213	R/O
tcpConnTable	1.3.6.1.2.1.6.13	Subagent	RFC1213	N/A
tcpConnEntry	1.3.6.1.2.1.6.13.1	Subagent	RFC1213	N/A
tcpConnState	1.3.6.1.2.1.6.13.1.1	Subagent	RFC1213	R/O
tcpConnLocalAddress	1.3.6.1.2.1.6.13.1.2	Subagent	RFC1213	R/O
tcpConnLocalPort	1.3.6.1.2.1.6.13.1.3	Subagent	RFC1213	R/O
tcpConnRemAddress	1.3.6.1.2.1.6.13.1.4	Subagent	RFC1213	R/O
tcpConnRemPort	1.3.6.1.2.1.6.13.1.5	Subagent	RFC1213	R/O
tcpInErrs	1.3.6.1.2.1.6.14	Subagent	RFC1213	R/O

Table 21 (Page 9 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
tcpOutRsts	1.3.6.1.2.1.6.15	Subagent	RFC1213	R/O
udpInDatagrams	1.3.6.1.2.1.7.1	Subagent	RFC1213	R/O
udpNoPorts	1.3.6.1.2.1.7.2	Subagent	RFC1213	R/O
udpInErrors	1.3.6.1.2.1.7.3	Subagent	RFC1213	R/O
udpOutDatagrams	1.3.6.1.2.1.7.4	Subagent	RFC1213	R/O
udpTable	1.3.6.1.2.1.7.5	Subagent	RFC1213	N/A
udpEntry	1.3.6.1.2.1.7.5.1	Subagent	RFC1213	N/A
udpLocalAddress	1.3.6.1.2.1.7.5.1.1	Subagent	RFC1213	R/O
udpLocalPort	1.3.6.1.2.1.7.5.1.2	Subagent	RFC1213	R/O
snmpInPkts	1.3.6.1.2.1.11.1	Agent	RFC1907	R/O
snmpInBadVersions	1.3.6.1.2.1.11.3	Agent	RFC1907	R/O
snmpInBadCommunityNames	1.3.6.1.2.1.11.4	Agent	RFC1907	R/O
snmpInBadCommunityUses	1.3.6.1.2.1.11.5	Agent	RFC1907	R/O
snmpInASNParseErrs	1.3.6.1.2.1.11.6	Agent	RFC1907	R/O
snmpEnableAuthenTraps	1.3.6.1.2.1.11.30	Agent	RFC1907	R/W
snmpSilentDrops	1.3.6.1.2.1.11.31	Agent	RFC1907	R/O
snmpProxyDrops	1.3.6.1.2.1.11.32	Agent	RFC1907	R/O

Table 21 (Page 10 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ifXTable	1.3.6.1.2.1.31.1.1	Subagent	ifMib Internet Draft	N/A
ifXEntry	1.3.6.1.2.1.31.1.1.1	Subagent	ifMib Internet Draft	N/A
ifName	1.3.6.1.2.1.31.1.1.1.1	Subagent	ifMib Internet Draft	R/O
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	Subagent	ifMib Internet Draft	R/O
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	Subagent	ifMib Internet Draft	R/O
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	Subagent	ifMib Internet Draft	R/O
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5	Subagent	ifMib Internet Draft	R/O
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	Subagent	ifMib Internet Draft	R/O
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	Subagent	ifMib Internet Draft	R/O
ifLinkUpDownTrapEnable	1.3.6.1.2.1.31.1.1.1.14	Subagent	ifMib Internet Draft	R/O
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15	Subagent	ifMib Internet Draft	R/O
ifPromiscuousMode	1.3.6.1.2.1.31.1.1.1.16	Subagent	ifMib Internet Draft	R/O
ifConnectorPresent	1.3.6.1.2.1.31.1.1.1.17	Subagent	ifMib Internet Draft	R/O
ifAlias	1.3.6.1.2.1.31.1.1.1.18	Subagent	ifMib Internet Draft	R/O
ifStackTable	1.3.6.1.2.1.31.1.2	Subagent	ifMib Internet Draft	N/A
ifStackEntry	1.3.6.1.2.1.31.1.2.1	Subagent	ifMib Internet Draft	N/A
ifStackStatus	1.3.6.1.2.1.31.1.2.1.3	Subagent	ifMib Internet Draft	R/O

Table 21 (Page 11 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
atmInterfaceConfTable	1.3.6.1.2.1.37.1.2	Subagent	RFC1695	N/A
atmInterfaceConfEntry	1.3.6.1.2.1.37.1.2.1	Subagent	RFC1695	N/A
atmInterfaceMaxVpcs	1.3.6.1.2.1.37.1.2.1.1	Subagent	RFC1695	R/O
atmInterfaceMaxVccs	1.3.6.1.2.1.37.1.2.1.2	Subagent	RFC1695	R/O
atmInterfaceConfVpcs	1.3.6.1.2.1.37.1.2.1.3	Subagent	RFC1695	R/O
atmInterfaceConfVccs	1.3.6.1.2.1.37.1.2.1.4	Subagent	RFC1695	R/O
atmInterfaceMaxActiveVpiBits	1.3.6.1.2.1.37.1.2.1.5	Subagent	RFC1695	R/O
atmInterfaceMaxActiveVciBits	1.3.6.1.2.1.37.1.2.1.6	Subagent	RFC1695	R/O
atmInterfaceIlmiVpi	1.3.6.1.2.1.37.1.2.1.7	Subagent	RFC1695	R/O
atmInterfaceIlmiVci	1.3.6.1.2.1.37.1.2.1.8	Subagent	RFC1695	R/O
atmInterfaceAddressType	1.3.6.1.2.1.37.1.2.1.9	Subagent	RFC1695	R/O
atmInterfaceAdminAddress	1.3.6.1.2.1.37.1.2.1.10	Subagent	RFC1695	R/O
atmInterfaceMyNeighborIpAddress	1.3.6.1.2.1.37.1.2.1.11	Subagent	RFC1695	R/O
atmInterfaceMyNeighborIfName	1.3.6.1.2.1.37.1.2.1.12	Subagent	RFC1695	R/O
dpiPort	1.3.6.1.4.1.2.2.1.1.0	Agent	RFC1592	R/O
dpiPortForTCP	1.3.6.1.4.1.2.2.1.1.1.0	Agent	RFC1592	R/O
dpiPortForUDP	1.3.6.1.4.1.2.2.1.1.2.0	Agent	RFC1592	R/O

Table 21 (Page 12 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
dpiPathNameForUnixStream	1.3.6.1.4.1.2.2.1.1.3.0	Agent	RFC1592B	R/O
saDefaultTimeout	1.3.6.1.4.1.2.4.12.1	Agent	SAMIB	R/W
saMaxTimeout	1.3.6.1.4.1.2.4.12.2	Agent	SAMIB	R/W
saAllowDuplicateIDs	1.3.6.1.4.1.2.4.12.3	Agent	SAMIB	R/W
saNumber	1.3.6.1.4.1.2.4.12.4	Agent	SAMIB	R/O
saAllPacketsIn	1.3.6.1.4.1.2.4.12.5	Agent	SAMIB	R/O
saAllPacketsOut	1.3.6.1.4.1.2.4.12.6	Agent	SAMIB	R/O
saTable	1.3.6.1.4.1.2.4.12.7	Agent	SAMIB	R/O
saEntry	1.3.6.1.4.1.2.4.12.7.1	Agent	SAMIB	R/O
saIndex	1.3.6.1.4.1.2.4.12.7.1.1	Agent	SAMIB	R/O
saIdentifier	1.3.6.1.4.1.2.4.12.7.1.2	Agent	SAMIB	R/O
saDescription	1.3.6.1.4.1.2.4.12.7.1.3	Agent	SAMIB	R/O
saStatus	1.3.6.1.4.1.2.4.12.7.1.4	Agent	SAMIB	R/W
saStatusChangeTime	1.3.6.1.4.1.2.4.12.7.1.5	Agent	SAMIB	R/O
saProtocol	1.3.6.1.4.1.2.4.12.7.1.6	Agent	SAMIB	R/O
saProtocolVersion	1.3.6.1.4.1.2.4.12.7.1.7	Agent	SAMIB	R/O
saProtocolRelease	1.3.6.1.4.1.2.4.12.7.1.8	Agent	SAMIB	R/O

Table 21 (Page 13 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
saTransport	1.3.6.1.4.1.2.4.12.7.1.9	Agent	SAMIB	R/O
saTransportAddress	1.3.6.1.4.1.2.4.12.7.1.10	Agent	SAMIB	R/O
saTimeout	1.3.6.1.4.1.2.4.12.7.1.11	Agent	SAMIB	R/W
saMaxVarBinds	1.3.6.1.4.1.2.4.12.7.1.12	Agent	SAMIB	R/O
saPacketsIn	1.3.6.1.4.1.2.4.12.7.1.13	Agent	SAMIB	R/O
saPacketsOut	1.3.6.1.4.1.2.4.12.7.1.14	Agent	SAMIB	R/O
saTreeTable	1.3.6.1.4.1.2.4.12.8	Agent	SAMIB	R/O
saTreeEntry	1.3.6.1.4.1.2.4.12.8.1	Agent	SAMIB	R/O
saTsubtree	1.3.6.1.4.1.2.4.12.8.1.1	Agent	SAMIB	R/O
saTpriority	1.3.6.1.4.1.2.4.12.8.1.2	Agent	SAMIB	R/O
saTindex	1.3.6.1.4.1.2.4.12.8.1.3	Agent	SAMIB	R/O
saTstatus	1.3.6.1.4.1.2.4.12.8.1.4	Agent	SAMIB	R/W
saTtimeout	1.3.6.1.4.1.2.4.12.8.1.5	Agent	SAMIB	R/W
ibmRemotePingTable	1.3.6.1.4.1.2.6.19.2.2.1.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmRemotePingEntry	1.3.6.1.4.1.2.6.19.2.2.1.1.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmMvsRPingResponseTime	1.3.6.1.4.1.2.6.19.2.2.1.1.1.4	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsSubagentCacheTime	1.3.6.1.4.1.2.6.19.2.2.2.1	SubAgent	ibmTCPIPmvsMIB	R/W

Table 21 (Page 14 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
osasfChannelTable	1.3.6.1.4.1.2.6.19.2.2.6.1	Subagent	ibmTCPIPmvsMIB	N/A
osasfChannelEntry	1.3.6.1.4.1.2.6.19.2.2.6.1.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfChannelNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.1	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.2	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelSubType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.3	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmO.sasfChannelMode	1.3.6.1.4.1.2.6.19.2.2.6.1.1.4	SubAgent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelHwModel	1.3.6.1.4.1.2.6.19.2.2.6.1 .1.5	SubAgent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelState	1.3.6.1.4.1.2.6.19.2.2.6.1.1.6	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelShared	1.3.6.1.4.1.2.6.19.2.2.6.1.1.7	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelNumPorts	1.3.6.1.4.1.2.6.19.2.2.6.1.1.8	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelDeterNodeDesc	1.3.6.1.4.1.2.6.19.2.2.6.1.1.9	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelControlUnitNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.10	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCodeLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.11	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelEcLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.12	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.13	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.14	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelManParnName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.15	Subagent	ibmTCPIPmvsMIB	R/O

Table 21 (Page 15 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ibmMvsAtmOsasfChannelManParnNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.16	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelDate	1.3.6.1.4.1.2.6.19.2.2.6.1.1.17	Subagent ibmTCPIPmvsMIB	R/O	
ibmMvsAtmOsasfChannelTime	1.3.6.1.4.1.2.6.19.2.2.6.1.1.18	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelFlashLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.19	Subagent	ibmTCPIPmvsMIB	R/O
osasfPortTable	1.3.6.1.4.1.2.6.19.2.2.6.2	Subagent	ibmTCPIPmvsMIB	N/A
osasfPortEntry	1.3.6.1.4.1.2.6.19.2.2.6.2.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPortNumber	1.3.6.1.4.1.2.6.19.2.2.6.2.1.1	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.2	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortHardwareState	1.3.6.1.4.1.2.6.19.2.2.6.2.1.3	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMediaType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.4	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.5	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniVersion	1.3.6.1.4.1.2.6.19.2.2.6.2.1.6	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.7	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.8	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.9	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortCodeLoadStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.10	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMacAddrBurntIn	1.3.6.1.4.1.2.6.19.2.2.6.2.1.11	Subagent	ibmTCPIPmvsMIB	R/O

Table 21 (Page 16 of 20). MIB Objects				
Object Descriptor	**Object Identifier**	**Supported by**	**Defined by**	**Access Allowed**
ibmMvsAtmOsasfPortMacAddrActive	1.3.6.1.4.1.2.6.19.2.2.6.2.1.12	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMaxPcmConnections	1.3.6.1.4.1.2.6.19.2.2.6.2.1.13	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortPcmName	1.3.6.1.4.1.2.6.19.2.2.6.2.1.14	Subagent	ibmTCPIPmvsMIB	R/O
osasfPvcTable	1.3.6.1.4.1.2.6.19.2.2.6.3	Subagent	ibmTCPIPmvsMIB	N/A
osasfPvcEntry	1.3.6.1.4.1.2.6.19.2.2.6.3.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPvcName	1.3.6.1.4.1.2.6.19.2.2.6.3.1.1	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBestEffort	1.3.6.1.4.1.2.6.19.2.2.6.3.1.2	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.3	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.4	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.5	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.6	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.7	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.8	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVpi	1.3.6.1.4.1.2.6.19.2.2.6.3.1.9	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVci	1.3.6.1.4.1.2.6.19.2.2.6.3.1.10	Subagent	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsTcpConnTable	1.3.6.1.4.1.2.6.19.2.2.7.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsTcpConnEntry	1.3.6.1.4.1.2.6.19.2.2.7.1.1	Subagent	ibmTCPIPmvsMIB	N/A

Table 21 (Page 17 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ibmMvsTcpConnLastActivity	1.3.6.1.4.1.2.6.19.2.2.7.1.1.1	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBytesIn	1.3.6.1.4.1.2.6.19.2.2.7.1.1.2	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBytesOut	1.3.6.1.4.1.2.6.19.2.2.7.1.1.3	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnActiveOpen	1.3.6.1.4.1.2.6.19.2.2.7.1.1.4	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIpTos	1.3.6.1.4.1.2.6.19.2.2.7.1.1.5	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOptions	1.3.6.1.4.1.2.6.19.2.2.7.1.1.6	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.7	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnUsrSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.8	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.9	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndUna	1.3.6.1.4.1.2.6.19.2.2.7.1.1.10	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.11	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.12	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.13	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSendWindowSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.14	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSendWindowAck	1.3.6.1.4.1.2.6.19.2.2.7.1.1.15	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.16	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.17	Subagent	ibmTCPIPmvsMIB	R/O

Table 21 (Page 18 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ibmMvsTcpConnUsrRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.18	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.19	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.20	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.21	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.22	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnMaxSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.23	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmtCount	1.3.6.1.4.1.2.6.19.2.2.7.1.1.24	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnCongestionWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.25	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSSThresh	1.3.6.1.4.1.2.6.19.2.2.7.1.1.26	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripTime	1.3.6.1.4.1.2.6.19.2.2.7.1.1.27	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripVariance	1.3.6.1.4.1.2.6.19.2.2.7.1.1.28	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitSndSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.29	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitRcvSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.30	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSendMSS	1.3.6.1.4.1.2.6.19.2.2.7.1.1.31	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl1	1.3.6.1.4.1.2.6.19.2.2.7.1.1.32	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl2	1.3.6.1.4.1.2.6.19.2.2.7.1.1.33	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.34	Subagent	ibmTCPIPmvsMIB	R/O

Table 21 (Page 19 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ibmMvsTcpConnPendTcpRecv	1.3.6.1.4.1.2.6.19.2.2.7.1.1.35	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvBufSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.36	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceName	1.3.6.1.4.1.2.6.19.2.2.7.1.1.37	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSubtask	1.3.6.1.4.1.2.6.19.2.2.7.1.1.38	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceId	1.3.6.1.4.1.2.6.19.2.2.7.1.1.39	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSockOpt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.40	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpTimer	1.3.6.1.4.1.2.6.19.2.2.7.1.1.41	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpSig	1.3.6.1.4.1.2.6.19.2.2.7.1.1.42	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpSel	1.3.6.1.4.1.2.6.19.2.2.7.1.1.43	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRttSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.44	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBackoffCount	1.3.6.1.4.1.2.6.19.2.2.7.1.1.45	Subagent	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsUdpTable	1.3.6.1.4.1.2.6.19.2.2.8.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsUdpEntry	1.3.6.1.4.1.2.6.19.2.2.8.1.1	Subagent	ibmTCPIPmvsMIB	N/A
ibmMvsUdpLastAct	1.3.6.1.4.1.2.6.19.2.2.8.1.1.1	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpTos	1.3.6.1.4.1.2.6.19.2.2.8.1.1.2	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpIpOpts	1.3.6.1.4.1.2.6.19.2.2.8.1.1.3	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.4	Subagent	ibmTCPIPmvsMIB	R/O

Table 21 (Page 20 of 20). MIB Objects

Object Descriptor	Object Identifier	Supported by	Defined by	Access Allowed
ibmMvsUdpBytesIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.5	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.6	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpBytesOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.7	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceName	1.3.6.1.4.1.2.6.19.2.2.8.1.1.8	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSubtask	1.3.6.1.4.1.2.6.19.2.2.8.1.1.9	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceId	1.3.6.1.4.1.2.6.19.2.2.8.1.1.10	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSockOpt	1.3.6.1.4.1.2.6.19.2.2.8.1.1.11	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSendLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.12	Subagent	ibmTCPIPmvsMIB	R/O
ibmMvsUdpRecvLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.13	Subagent	ibmTCPIPmvsMIB	R/O
snmpSetSerialNo	1.3.6.1.6.3.1.1.6.1	Agent	RFC1907	R/W

Appendix D. OE SNMP Trap Types

This appendix lists the generic trap and enterprise-specific types that can be received by OE SNMP.

OE SNMP Generic Trap Types

This section lists the generic trap types that can be received by OE SNMP.

Table 22. Generic Trap Types

Value	Type	Description

0	**coldStart**	**A coldStart trap signifies that the sending protocol entity is reinitializing itself so that the agent's configuration or the protocol entity implementation can be altered.**
1	**warmStart**	**A warmStart trap signifies that the sending protocol entity is reinitializing itself so that neither the agent configuration nor the protocol entity implementation can be altered.**
2	**linkDown**	**A linkDown trap signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration.** A Trap-PDU of type linkDown contains, as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.
3	**linkUp**	**A linkUp trap signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up.** A Trap-PDU of type linkUp contains, as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.
4	**authenticationFailure**	**An authenticationFailure trap signifies that the sending protocol entity is the addressee of a protocol message that is not properly authenticated.**
5	**egpNeighborLoss**	**An egpNeighborLoss trap signifies that an EGP neighbor for whom the sending protocol entity was an EGP peer has been marked down and the peer relationship no longer exists.** The Trap-PDU of the egpNeighborLoss contains, as the first element of its variable-bindings, the name and value of the egpNeighAddr instance for the affected neighbor.
6	**enterpriseSpecific**	**An enterpriseSpecific trap signifies that the sending protocol entity recognizes that some enterprise-specific event has occurred. The specific-trap field identifies the particular trap that occurred.**

OE SNMP MVS Enterprise—Specific Trap Types

This section lists the MVS enterprise trap types.

Table 23. MVS Enterprise Trap Types

Value	Type	Description

6	ibmMvsAtmOsasfAtmPvcDelete	This trap is generated when OSA/SF sends a asyn notification to the TCP/IP DPI Subagent that a PVC was deleted for a given OSA Port. This notification contains the corresponding ibmMvsAtmOsasfPortName instance. Representation of this will contain the port's (aal5 layer interface) 'ifIndex.pvcNameOctetCount.pvcNameInASCIINvt'.
6	ibmMvsAtmOsasfAtmPvcCreate	This trap is generated when OSA/SF sends a asyn notification to the TCP/IP DPI Subagent that a PVC was created for a given OSA Port. This notification contains the corresponding ibmMvsAtmOsasfPortName instance. Representation of this will contain the port's (aal5 layer interface) 'ifIndex.pvcNameOctetCount.pvcNameInASCIINvt'.

Appendix E. Related Protocol Specifications

This appendix lists the related protocol specifications for TCP/IP for MVS. The Internet suite of protocols is still evolving through Requests for Comments (RFC). New protocols are being designed and implemented by researchers, and are brought to the attention of the Internet community in the form of RFCs. Some of these are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement this particular function or protocol. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

Many features of TCP/IP for MVS are based on the following RFCs:

RFC	Title and Author
768	*User Datagram Protocol* J.B. Postel
791	*Internet Protocol* J.B. Postel
792	*Internet Control Message Protocol* J.B. Postel
793	*Transmission Control Protocol* J.B. Postel
821	*Simple Mail Transfer Protocol* J.B. Postel
822	*Standard for the Format of ARPA Internet Text Messages* D. Crocker
823	*DARPA Internet Gateway* R.M. Hinden, A. Sheltzer
826	*Ethernet Address Resolution Protocol: or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware* D.C. Plummer
854	*Telnet Protocol Specification* J.B. Postel, J.K. Reynolds
855	*Telnet Option Specification* J.B. Postel, J.K. Reynolds
856	*Telnet Binary Transmission* J.B. Postel, J.K. Reynolds
857	*Telnet Echo Option* J.B. Postel, J.K. Reynolds
858	*Telnet Suppress Go Ahead Option* J.B. Postel, J.K. Reynolds
859	*Telnet Status Option* J.B. Postel, J.K. Reynolds
860	*Telnet Timing Mark Option* J.B. Postel, J.K. Reynolds
861	*Telnet Extended Options —List Option* J.B. Postel, J.K. Reynolds
862	*Echo Protocol* J.B. Postel
863	*Discard Protocol* J.B. Postel
864	*Character Generator Protocol* J.B. Postel
877	*Standard for the Transmission of IP Datagrams over Public Data Networks* J.T. Korb
885	*Telnet End of Record Option* J.B. Postel
903	*Reverse Address Resolution Protocol* R. Finlayson, T. Mann, J.C. Mogul, M. Theimer
904	*Exterior Gateway Protocol Formal Specification* D.L. Mills
919	*Broadcasting Internet Datagrams* J.C. Mogul

922 *Broadcasting Internet Datagrams in the Presence of Subnets* J.C. Mogul

950 *Internet Standard Subnetting Procedure* J.C. Mogul, J.B. Postel

952 *DoD Internet Host Table Specification* K. Harrenstien, M.K. Stahl, E.J. Feinler

959 *File Transfer Protocol* J.B. Postel, J.K. Reynolds

974 *Mail Routing and the Domain Name System* C. Partridge

1009 *Requirements for Internet Gateways* R.T. Braden, J.B. Postel

1013 *X Window System Protocol, Version 11: Alpha Update* R.W. Scheifler

1014 *XDR: External Data Representation Standard* Sun Microsystems Incorporated

1027 *Using ARP to Implement Transparent Subnet Gateways* S. Carl-Mitchell, J.S. Quarterman

1032 *Domain Administrators Guide* M.K. Stahl

1033 *Domain Administrators Operations Guide* M. Lottor

1034 *Domain Names—Concepts and Facilities* P.V. Mockapetris

1035 *Domain Names—Implementation and Specification* P.V. Mockapetris

1042 *Standard for the Transmission of IP Datagrams over IEEE 802 Networks* J.B. Postel, J.K. Reynolds

1044 *Internet Protocol on Network System's HYPERchannel: Protocol Specification* K. Hardwick, J. Lekashman

1055 *Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP* J.L. Romkey

1057 *RPC: Remote Procedure Call Protocol Version 2 Specification* Sun Microsystems Incorporated

1058 *Routing Information Protocol* C.L. Hedrick

1073 *Telnet Window Size Option* D. Waitzman

1079 *Telnet Terminal Speed Option* C.L. Hedrick

1091 *Telnet Terminal-Type Option* J. VanBokkelen

1094 *NFS: Network File System Protocol Specification* Sun Microsystems Incorporated

1096 *Telnet X Display Location Option* G. Marcy

1118 *Hitchhikers Guide to the Internet* E. Krol

1122 *Requirements for Internet Hosts—Communication Layers* R.T. Braden

1123 *Requirements for Internet Hosts—Application and Support* R.T. Braden

1155 *Structure and Identification of Management Information for TCP/IP-Based Internets* M.T. Rose, K. McCloghrie

1156 *Management Information Base for Network Management of TCP/IP-based Internets* K. McCloghrie, M.T. Rose

1157 *Simple Network Management Protocol (SNMP)* J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin

1179 *Line Printer Daemon Protocol* The Wollongong Group, L. McLaughlin III

1180 *TCP/IP Tutorial* T.J. Socolofsky, C.J. Kale

1183 *New DNS RR Definitions* C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris, (Updates RFC 1034, RFC 1035)

1184 *Telnet Linemode Option* D. Borman

1187 *Bulk Table Retrieval with the SNMP* M.T. Rose, K. McCloghrie, J.R. Davin

1188 *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks* D. Katz

1198 *FYI on the X Window System* R.W. Scheifler

1207 *FYI on Questions and Answers: Answers to Commonly Asked :q.Experienced Internet User:eq. Questions* G.S. Malkin, A.N. Marine, J.K. Reynolds

1208 *Glossary of Networking Terms* O.J. Jacobsen, D.C. Lynch

1213 *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II,* K. McCloghrie, M.T. Rose

1215 *Convention for Defining Traps for Use with the SNMP* M.T. Rose

1228 *SNMP-DPI Simple Network Management Protocol Distributed Program Interface* G.C. Carpenter, B. Wijnen

1229 *Extensions to the Generic-Interface MIB* K. McCloghrie

1230 *IEEE 802.4 Token Bus MIB IEEE 802 4 Token Bus MIB* K. McCloghrie, R. Fox

1231 *IEEE 802.5 Token Ring MIB IEEE 802.5 Token Ring MIB* K. McCloghrie, R. Fox, E. Decker

1267 *A Border Gateway Protocol 3 (BGP-3)* K. Lougheed, Y. Rekhter

1268 *Application of the Border Gateway Protocol in the Internet* Y. Rekhter, P. Gross

1269 *Definitions of Managed Objects for the Border Gateway Protocol (Version 3)* S. Willis, J. Burruss

1270 *SNMP Communications Services* F. Kastenholz, ed.

1340 *Assigned Numbers* J.K. Reynolds, J.B. Postel

1348 *DNS NSAP RRs* B. Manning

1350 *TFTP Protocol* K.R. Sollins

1351 *SNMP Administrative Model* J. Davin, J. Galvin, K. McCloghrie

1352 *SNMP Security Protocols* J. Galvin, K. McCloghrie, J. Davin

1353 *Definitions of Managed Objects for Administration of SNMP Parties* K. McCloghrie, J. Davin, J. Galvin

1354 *IP Forwarding Table MIB* F. Baker

1356 *Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode* A. Malis, D. Robinson, R. Ullmann

1372 *Telnet Remote Flow Control Option* D. Borman, C. L. Hedrick

1374 *IP and ARP on HIPPI* J. Renwick, A. Nicholson

1381 *SNMP MIB Extension for X.25 LAPB* D. Throop, F. Baker

1382 *SNMP MIB Extension for the X.25 Packet Layer* D. Throop

1387 *RIP Version 2 Protocol Analysis* G. Malkin

1388 *RIP Version 2 — Carrying Additional Information* G. Malkin

1389 *RIP Version 2 MIB Extension* G. Malkin

1390 *Transmission of IP and ARP over FDDI Networks* D. Katz

1393 *Traceroute Using an IP Option* G. Malkin

1397 *Default Route Advertisement In BGP2 And BGP3 Versions of the Border Gateway Protocol* D. Haskin

1398 *Definitions of Managed Objects for the Ethernet-like Interface Types* F. Kastenholz

1540 *IAB Official Protocol Standards* J.B. Postel

1571 *Telnet Environment Option Interoperability Issues* D. Borman

1572 *Telnet Environment Option* S. Alexander

1592 *Simple Network Management Protocol Distributed Protocol Interface Version 2.0* B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

1594 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* A.N. Marine, J. Reynolds, G.S. Malkin

1695 *Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2*M. Ahmed, K. Tesink

1901 *Introduction to Community-Based SNMPv2* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1902 *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1903 *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1904 *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1905 *Protocols Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1906 *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1907 *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1908 *Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1909 *An Administrative Infrastructure for SNMPv2* K. McCloghrie

1910 *User-based Security Model for SNMPv2* G. Waters

These documents can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or on a subscription basis. Online copies are available using FTP from the NIC at `nic.ddn.mil`. Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

Where:

nnnn	Is the RFC number.
`TXT`	Is the text format.
`PS`	Is the PostScript format.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC` *nnnn* for text versions or a subject line of `RFC` *nnnn*`.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

Appendix F. Abbreviations and Acronyms

This appendix lists the abbreviations and acronyms used throughout this book.

AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
API	Application program interface
APPC	Advanced program-to-program communications
APPN	Advanced Peer-to-Peer Networking
ARP	Address Resolution Protocol
ASCII	American National Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
BIOS	Basic Input/Output System
CCITT	Comite Consultatif International Telegraphique et Telephonique. The International Telegraph and Telephone Consultative Committee
CCSID	Coded character set identifier
CETI	Continuously Executing Transfer Interface
CICS	Customer Information Control System
CLAW	Common Link Access to Workstation
CLIST	Command list
CMS	Conversational Monitor System
CP	Control Program
CPI	Common Programming Interface
CREN	Corporation for Research and Education Networking
CSD	Corrective service diskette
CSI	Control sequence indicator
CTC	Channel-to-channel
CU	Control unit
CUA	Common User Access
DASD	Direct access storage device
DBCS	Double-byte character set
DB2	DATABASE 2
DLL	Dynamic link library
DNS	Domain Name System
DOS	Disk Operating System
DPI	Distributed Program Interface
EBCDIC	Extended binary-coded decimal interchange code
ESCON	Enterprise Systems Connection

EUC	Extended UNIX Code
FAT	File allocation table
FDDI	Fiber Distributed Data Interface
FTAM	File Transfer Access Management
FTP	File Transfer Protocol
FTP API	File Transfer Protocol Applications Programming Interface
GCS	Group Control System
GDF	Graphics Data File
HCH	HYPERchannel device
HFS	Hierarchical file system
HIPPI	High Performance Parallel Interface
HPFS	High Performance File System
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IMS	Information Management System
IP	Internet Protocol
IPL	Initial program load
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
JES	Job Entry Subsystem
JIS	Japanese Institute of Standards
JCL	Job control language
KSC	Korean Standard Code
LAN	Local area network
LAPS	LAN Adapter Protocol Support
LCS	IBM LAN Channel Station
LU	Logical unit
Mbps	Megabits per second
MBps	Megabytes per second
MIB	Management Information Base
MIH	Missing Interrupt Handler
MTU	Maximum transmission unit
MVS	Multiple Virtual Storage
NDIS	Network Driver Interface Specification
NFS	Network File System
NIC	Network Information Center

NLS	National Language Support
OS/2	Operating System/2
OSF	Open Software Foundation, Inc.
OSI	Open Systems Interconnection
PC	Personal computer
PCA	Parallel Channel Adapter
PDS	Partitioned data set
PDU	Protocol Data Units
PING	Packet Internet Groper
PIOAM	Parallel I/O Access Method
PU	Physical unit
RACF	Resource Access Control Facility
RARP	Reverse Address Resolution Protocol
REXEC	Remote execution
REXX	Restructured Extended Executor Language
RFC	Request For Comments
RIP	Routing Information Protocol
RISC	Reduced Instruction Set Computer
RPC	Remote procedure call
RSCS	Remote Spooling Communications Subsystem
SBCS	Single-byte character set
SDLC	Synchronous Data Link Control
SJIS	Shift-Japanese Institute of Standards
SMS	Storage Management Subsystem
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SO/SI	Shift-Out/Shift-In
SPOOL	Simultaneous Peripheral Operations Online
SQL	Structured Query Language
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
TSO	Time Sharing Option
TTL	Time-to-Live
UDP	User Datagram Protocol
VGA	Video Graphic Array
VM	Virtual machine

VMCF	Virtual machine communication facility
VM/SP	Virtual Machine/System Product
VM/XA	Virtual Machine/Extended Architecture
VTAM	Virtual Telecommunications Access Method
WAN	Wide area network
XDR	eXternal Data Representation

Appendix G. How to Read a Syntax Diagram

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and Punctuation

The following symbols are used in syntax diagrams:

Symbol	Description
►►	Marks the beginning of the command syntax.
►	Indicates that the command syntax is continued.
|	Marks the beginning and end of a fragment or part of the command syntax.
►◄	Marks the end of the command syntax.

You must include all punctuation such as colons, semicolons, commas, quotation marks, and minus signs that are shown in the syntax diagram.

Parameters

The following types of parameters are used in syntax diagrams.

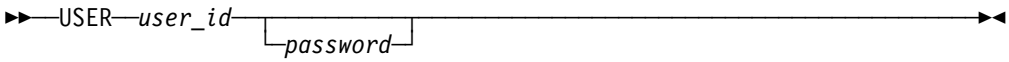
Parameter	Description
Required	Required parameters are displayed on the main path.
Optional	Optional parameters are displayed below the main path.
Default	Default parameters are displayed above the main path.

Parameters are classified as keywords or variables. Keywords are displayed in uppercase letters and can be entered in uppercase or lowercase. For example, a command name is a keyword.

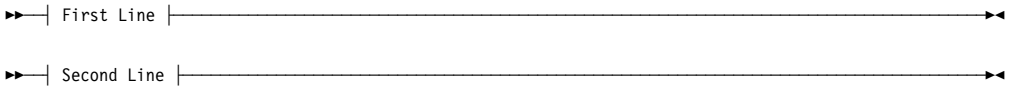
Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set is a variable.

Syntax Examples

In the following example, the USER command is a keyword. The required variable parameter is *user_id*, and the optional variable parameter is *password*. Replace the variable parameters with your own values.



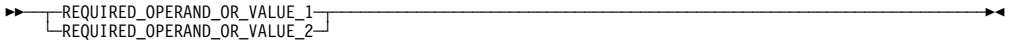
Longer than one line: If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.



Required operands: Required operands and values appear on the main path line.

You must code required operands and values.

Choose one required item from a stack: If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically in alphanumeric order.



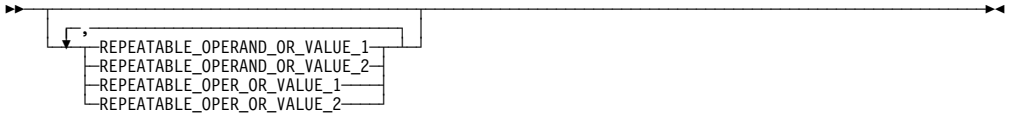
Optional values: Optional operands and values appear below the main path line.

You can choose not to code optional operands and values.

Choose one optional operand from a stack: If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.

Repeating an operand: An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The commad means that each operand or value must be separated from the next by a comma.

Selecting more than one operand: An arrow returning to the left above a group of operands or values means more than one can be selected, or a single one can be repeated.



If an operand or value can be abbreviated, the abbreviation is described in the text associated with the syntax diagram.

Nonalphanumeric characters: If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code OPERAND=(001,0.001).

Blank spaces in syntax diagrams: If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code OPERAND=(001 FIXED).

Default operands: Default operands and values appear above the main path line. TCP/IP uses the default if you omit the operand entirely.

Variables: A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

Syntax fragments: Some diagrams contain syntax fragments, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

Syntax Fragment:

Appendix H. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California.

Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
AD/Cycle
AIX
AIX/ESA
BookManager
C/370
CICS
DB2
DFSMS
DFSMS/MVS
ESCON
ES/9000
EtherStreamer
Extended Services
GDDM
Hardware Configuration Definition
IBM
LANStreamer
Library Reader
MVS/ESA
MVS/SP
MVS/XA
NetView
OpenEdition
OS/2
OS/390
PS/2
RACF
RISC System/6000
RS/6000
SAA
System/370
System/390
VTAM
3090

The following terms are trademarks of other companies:

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Bibliography

This bibliography lists the publications for IBM TCP/IP products.

IBM TCP/IP Publications

The following sections describe the books associated with IBM TCP/IP products.

OS/390 TCP/IP OpenEdition Publications

- *OS/390 TCP/IP OpenEdition Configuration Guide*, SC31-8304-00.

  This book is for people who want to configure, customize, administer, and maintain OS/390 TCP/IP OpenEdition. Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition Diagnosis Guide*, SC31-8492-00.

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the OS/390 TCP/IP OpenEdition product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *OS/390 TCP/IP OpenEdition Messages and Codes*, SC31-8307-00.

  This book explains the informational and error messages issued by OS/390 TCP/IP OpenEdition. It can help users, operators, or system programmers to diagnose and fix problems identified by error messages.

- *OS/390 TCP/IP OpenEdition Planning and Release Guide*, SC31-8303-00.

  This book is intended to help you plan for OS/390 TCP/IP OpenEdition whether you are migrating from a previous version or installing TCP/IP for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with OS/390 TCP/IP OpenEdition.

- *OS/390 TCP/IP OpenEdition Programmer's Reference*, SC31-8308-00

  This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing.

  This book is for people who want to use the supplied interfaces while writing application programs that access OS/390 TCP/IP OpenEdition. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition User's Guide*, GC31-8305-00.

  This book is for people who want to use OS/390 TCP/IP OpenEdition for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for MVS Publications

- *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, SC31-8069-00.

  This book explains how to plan for, install, customize, and use the OpenEdition MVS Applications Feature. The Feature consists of applications and interfaces for direct access to the OpenEdition MVS environment. For example, users of the Feature can use MVS, UNIX, or AIX commands to transfer files, log in to the OpenEdition environment without going through TSO, and run commands remotely. This book also explains how to improve performance and diagnose problems when using the Feature.

- *TCP/IP for MVS: Application Programming Interface Reference*, SC31-7187-02.

  This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

- *TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*, SC31-7131-02.

  This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

- *TCP/IP for MVS: Customization and Administration Guide*, SC31-7134-03.

  This book is for people who want to con customize, administer, and maintain TCP/IP for MVS. Familiarity with MVS operating system, TCP/IP protocols,

and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: Diagnosis Guide*, LY43-0105-02.

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the IBM TCP/IP for MVS product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference*, SC31-7186-02.

  This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

- *TCP/IP for MVS: Messages and Codes*, SC31-7132-03.

  This book explains the informational a nd error messages issued by IBM TCP/IP for MVS. It can help users, operators, or system programmers to diagnose and fix problems identified by TCP/IP for MVS error messages.

- *TCP/IP for MVS: Network Print Facility*, SC31-8074-03.

  This book is for system programmers an d network administrators who need to prepare their network to route VTAM, JES2, or JES3 printer output to remote printers using TCP/IP for MVS.

- *TCP/IP for MVS: Offloading TCP/IP Processing*, SC31-7133-02.

  This book is for people who want to install and configure the Offload feature on IBM 3172 Model 3 Interconnect Controllers. This book is also for people who want to use and customize the Offload feature of TCP/IP for MVS.

- *TCP/IP for MVS: Planning and Migration Guide*, SC31-7189-01.

  This book is intended to help you plan for TCP/IP for MVS whether you are migrating from a previous version or installing TCP/IP for MVS for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with TCP/IP for MVS.

- *TCP/IP: Performance Tuning Guide*, SC31-7188-02.

  This book describes how to improve the performance of your network operations.

- *TCP/IP for MVS: Programmer's Reference*, SC31-7135-02.

  This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing.

  This book is for people who want to use the supplied interfaces while writing application programs that access TCP/IP for MVS. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: User's Guide*, SC31-7136-02.

  This book is for people who want to use TCP/IP for MVS for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for VM Publications

The following list describes books in the IBM TCP/IP for VM library.

- *IBM TCP/IP Version 2 Release 4 for VM: Messages and Codes*, SC31-6151-03.

  This book is for system programmers who want to diagnose and fix problems identified by TCP/IP for VM error messages.

- *IBM TCP/IP Version 2 Release 4 for VM: Planning and Customization*, SC31-6082-03.

  This book is for system programmers who want to plan and customize the TCP/IP for VM environment.

- *IBM TCP/IP Version 2 Release 4 for VM: Programmer's Reference*, SC31-6084-03.

  This book is for application and system programmers who want to write application programs that use TCP/IP for VM. Application programmers should know the VM operating system.

- *IBM TCP/IP Version 2 Release 4 for VM: User's Guide*, SC31-6081-03.

  This book is for people who want to use TCP/IP for VM for data communication. Familiarity with VM operating system, IBM Command Processor (CP), and IBM Conversational Monitor System (CMS) is recommended.

TCP/IP for OS/2 Publication

IBM TCP/IP Version 3.0 for OS/2: Programmer's Reference, SC31-6077.

This book provides application and system programmers with the information required to write application programs that use TCP/IP for OS/2. Programmers should know the OS/2 operating system.

TCP/IP for DOS Publications

The following list describes books in the IBM TCP/IP for DOS library.

- *IBM TCP/IP Version 2.1.1 for DOS: Command Reference*, SX75-0083.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, understand DOS operating system concepts, and be familiar with the *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*

- *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration*, SC31-7047.

  This book provides system programmers, network administrars, and workstation users responsible for installing TCP/IP for DOS with the information required to plan and implement the installation of TCP/IP for DOS. The topics include hardware and software requirements, pre-installation system performance considerations, instructions for installing TCP/IP for DOS, instructions for customizing the TCP/IP for DOS environment, and installation examples.

- *IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference*, SC31-7046.

  This book is for application and system programmers to aid them in writing application programs that use TCP/IP for DOS on a workstation. Application programmers should know the DOS operating system and multitasking operating system concepts. Application programmers should be knowledgeable in the C programming language.

- *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*, SC31-745.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, and also understand DOS operating system concepts.

TCP/IP for AIX (RS/6001, PS/2, RT, 370) Publications

The following list shows books in the TCP/IP for AIX library.

- *AIX Operating System TCP/IP User's Guide*, SC23-2309.
- *AIX PS/2 TCP/IP User's Guide*, SC23-2047.
- *TCP/IP for IBM X-Windows on DOS 2.1*, SC23-2349.

TCP/IP for AS/400 Publications

The following list shows books in the TCP/IP for AS/400 library.

- *IBM AS/400 Communications: TCP/IP Guide*, SC41-9875.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.

Other IBM TCP/IP Publications

The following list shows other available IBM TCP/IP books.

- *IBM Local Area Network Technical Reference*, SC30-3383.
- *IBM TCP/IP for VM and MVS: Diagnosis Guide*, LY43-0013.
- *TCP/IP and National Language Support*, GG24-3840.
- *TCP/IP Introduction*, GC31-6080.
- *TCP/IP Tutorial and Technical Overview*, GG24-3376.

IBM Operating System Publications

The following lists show books about various IBM operating systems.

AIX Publications

- *AIX Communications Concepts and Procedures for IBM RISC System/6001*, GC23-2203.
- *AIX Communications Programming Concepts*, SC23-2206.
- *IBM AIX Operating System Technical Reference, Volume 1*, SC23-2300.
- *IBM AIX Operating System Technical Reference, Volume 2*, SC23-2301.

AS/400 Publications

- *IBM AS/400 CL Reference Manual Volume 1*, SC21-9775.
- *IBM AS/400 CL Reference Manual Volume 2*, SC21-9776.
- *IBM AS/400 CL Reference Manual Volume 3*, SC21-9777.
- *IBM AS/400 CL Reference Manual Volume 4*, SC21-9778.

- *IBM AS/400 CL Reference Manual Volume 5*, SC21-9779.
- *IBM AS/400 Communications: APPN Network User's Guide*, SC21-8188.
- *IBM AS/400 Communications: Programmer's Guide*, SC21-9590.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.
- *IBM AS/400 Device Configuration Guide*, SC21-8106.
- *IBM AS/400 Programming: Command Reference Summary*, SC21-8076.
- *IBM AS/400 Programming: Data Management Guide*, SC21-9658.
- *IBM AS/400 System Operations: Database Coordinator' Guide*, SC21-8086.
- *IBM AS/400 System Operations: Operator's Guide*, SC21-8082.

DOS Publications

- *DOS Getting Started Version 5.00*, SA40-0637.
- *DOS 5.02 Technical Reference*, S16G-4559.
- *DOS/Windows Client Getting Started*, SC09-3001.
- *PC DOS 6.1 Command Reference*, S71G-3634.

MVS Publications

For a complete description of the library for MVS/ESA Version 5, see *OS/390 Information Roadmap*, GC28-1727-02. See also “JES Publications” on page 274.

OS/2 Publications

- *IBM OS/2 Warp Server Up and Running!*, S25H-8004
- *IBM Official Guide to Using OS/2 Warp*, ISBN 1-56884-466-2 (Karla Stagray and Linda S. Rogers; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)
- *IBM OS/2 Warp Internet Connection: Your Key to Cruising the Internet and the World Wide Web*, ISBN 1-56884-465-4 (Deborah Morrison; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)

OS/390 Publications

- *OS/390 Information Roadmap*, GC28-1727-02

  This book describes the documentation for the specific elements included in OS/390.
- *OS/390 Planning for Installation Release 3*, GC28-1726-02

  This book is intended to help you plan for the installation of OS/390. It describes migration, installation, hardware and software requirements, and coexistence considerations.
- *OS/390 OpenEdition Introduction*, GC28-1889-01.
- *OS/390 OpenEdition Planning*, SC28-1890-02.
- *OS/390 OpenEdition User's Guide*, SC28-1891-02.
- *OS/390 OpenEdition Command Reference*, SC28-1892-02.
- *OS/390 OpenEdition Messages and Codes*, SC28-1908-02.
- *OS/390 Language Environment Programming Guide*, SC28-1939-02.
- *OS/390 Language Environment Programming Reference*, SC28-1940-02.
- *OS/390 OpenEdition Programming: Assembler Callable Services Reference*, SC28-1899-02.
- *OS/390 Open Systems Adapter Support Facility Users's Guide*, SC28-1855.
- *Planning for the System/390 Open Systems Adapter Feature*, GC23-3870.

VM Publications

- *VM/ESA CMS Command Reference Summary*, SX24-5249.
- *VM/ESA CP Planning and Administration for 370*, SC24-5430.
- *VM/ESA CP Programming Services for 370*, SC24-5435.
- *VM/ESA Group Control System Reference for 370*, SC24-5426.
- *VM/ESA: Library Guide and Master Index*, GC23-0367.
- *VM/ESA: Master Index for 370*, GC24-5436.
- *VM/ESA Service Introduction and Reference*, SC24-5444.
- *VM/SP CMS Command Reference*, ST00-1981.
- *VM/SP Group Control System Macro Reference*, SC24-5250.
- *VM/SP Installation Guide*, SC24-5237.
- *VM/SP High Performance Option:*

Library Guide and Master Index, GC23-0187.

- *VM/SP System Facilities for Programming*, SC24-5288.
- *VM/XA CP Programming Services*, SC23-0370.
- *VM/XA Diagnosis Reference*, LY27-8054.
- *VM/XA Installation and Service*, SC23-0364.
- *VM/XA SP Group Control System Command and Macro Reference*, SC23-0433.

IBM Software Publications

The following sections describe the books associated with IBM software products.

ACF/VTAM Publications

The following list shows books in the VTAM Version 4 Release 4 library.

- *VTAM Installation and Migration Guide*, GC31-8367-00.
- *VTAM Release Guide*, GC31-6545-00.
- *VTAM Network Implementation Guide*, SC31-8370-00.
- *VTAM Resource Definition Reference*, SC31-8377-00.
- *VTAM Resource Definition Samples*, SC31-8378-00.
- *VTAM Customization*, LY43-0075-00.
- *VTAM Operation*, SC31-8372-00.
- *VTAM Messages*, GC31-8368-00.
- *VTAM Codes*, GC31-8369-00.
- *VTAM Programming*, SC31-8373-00.
- *VTAM Guide to Programming for LU 6.2*, SC31-8374-00.
- *VTAM Programming Reference for LU 6.2*, SC31-8375-00.
- *VTAM Programming for CSM*, SC31-8420-00.
- *VTAM CMIP Services and Topology Agent Programming Guide*, SC31-8365-00.
- *VTAM Diagnosis*, LY43-0078-00.
- *VTAM Data Areas for MVS/ESA Volume 1*, LY43-0076-00.
- *VTAM Data Areas for MVS/ESA Volume 2*, LY40-0077-00.
- *APPC Application Suite User's Guide*, SC31-6532-00.
- *APPC Application Suite Administration*, SC31-6533-00.
- *APPC Application Suite Programming*, SC31-6534-00.
- *VTAM AnyNet Guide to Sockets over SNA*, SC31-8371-00.
- *VTAM AnyNet Guide to SNA over TCP/IP*, SC31-8376-00.
- *VTAM Glossary*, GC31-8366-00.
- *Planning for NetView, NCP, and VTAM*, SC31-8063-00.
- *Planning for Integrated Networks*, SC31-8062-00.
- *VTAM Licensed Program Specifications*, GC31-8379-00.
- *VTAM Operation Quick Reference*, SX75-0208-00.

DATABASE 2 Publications

The following lists show books in the DATABASE 2 library.

DATABASE 2 Version 2

- *IBM DATABASE 2 Version 2: Administration Guide*, SC26-4374.
- *IBM DATABASE 2 Version 2: Application Programming and SQL Guide*, SC26-4377.
- *IBM DATABASE 2 Version 2: Messages and Codes*, SC26-4379.
- *IBM DATABASE 2 Version 2: Reference Summary*, SX26-3771.
- *IBM DATABASE 2 Version 2: SQL Reference*, SC26-4380.

DATABASE 2 Version 3

- *IBM DATABASE 2 Version 3: DB2 Administration Guide*, SC26-4888.
- *IBM DATABASE 2 Version 3: DB2 Application Programming and SQL Guide*, SC26-4889.
- *IBM DATABASE 2 Version 3: DB2 Messages and Codes*, SC26-4892.
- *IBM DATABASE 2 Version 3: DB2 Reference Summary*, SX26-3801.
- *IBM DATABASE 2 Version 3: DB2 SQL Reference*, SC26-4890.

ISPF Publication

ISPF Dialog Management Guide and Reference, SC34-4266.

JES Publications

- *MVS/ESA Library Guide with JES2*, GC28-1423.
- *MVS/ESA Library Guide with JES3*, SC28-1424

MVS/DFP Publications

- *MVS/DFP Version 3 Release 3: Customizing and Operating the Network File System Server*, SC26-4832.
- *MVS/DFP Version 3 Release 3: Macro Instructions for Data Sets*, S26-4747.
- *MVS/DFP Version 3 Release 3: Using Data Sets*, SC26-4749.
- *MVS/DFP Version 3 Release 3: Using the Network File System Server*, SC26-4732.

Network Control Program (NCP) Publications

- *ACF/NCP V7R1 IP Router Planning and Installation Guide*, GG24-3974.
- *NCP and EP Reference*, LY43-0029.
- *NCP, SSP, and EP Generation and Loading Guide*, SC31-6221.
- *NCP, SSP, and EP Resource Definition Guide*, SC31-6223.
- *NCP, SSP, and EP Resource Definition Reference*, SC31-6224.

TME 10 NetView for OS/390 Publications

For a complete description of the TME 10 NetView for OS/390 library, see the *TME 10 NetView for OS/390 Library Reference*, SC31-8249.

Networking Systems Cross-Product Library

The following list shows books in the Networking Systems cross-product library.

- *Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM*, SX75-0092.
- *Planning for Integrated Networks*, SC31-8062.
- *Planning for NetView, NCP, and VTAM*, SC31-8063.

OpenEdition MVS Publications

The following list shows selected books in the OpenEdition MVS library.

- *OS/390 OpenEdition Introduction*, GC28-1889-01
- *OS/390 OpenEdition Planning*, SC28-1890-02

Programming Publications

The following list shows books about various programming applications.

- *IBM C/370 Diagnosis Guide and Reference* LY09-1804 (feature 8082).
- *IBM C/370 General Information Manual* GC09-1386.
- *IBM C/370 Installation and Customization Guide Version 2 Release 1.0*, GC09-1387.
- *IBM C/370 Programming Guide*, SC09-1384.
- *IBM C/370 Reference Summary*, SX09-1211.
- *IBM C/370 User's Guide*, SC09-1264.
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663-01.
- *IBM TSO Extensions CLISTs*, SC28-1876.
- *IBM TSO Extensions Command Language Reference* GX23-0015.
- *IBM TSO Extensions Interactive Data Transmission Facility: User's Guide*, SC28-1104.
- *IMS/ESA V3R1 Application Programming: DL/I Calls* SC26-4274.
- *HiPPI User's Guide and Programmer's Reference*, SA23-0369.
- *Parallel I/O Access Methods Programmer's Guide*, SC26-4648.
- *VS Pascal Application Programming Guide* SC26-4319.
- *VS Pascal Diagnosis Guide and Reference* LY27-9525.
- *VS Pascal General Information*, GT00-2664.
- *VS Pascal Installation and Customization for MVS* SC26-4321.
- *VS Pascal Installation and Customization for VM* SC26-4342.
- *VS Pascal Language Reference*, SC26-4320.

RACF Publications

The following list shows books in the RACF library.

- *IBM Resource Access Control Facility (RACF): General Information Manual*, GT00-2820.
- *IBM Resource Access Control Facility (RACF): User's Guide*, SC28-1341.
- *External Security Interface (RACROUTE) Macro Reference*, GC28-1366.
- *RACF Publications Order Guide*, GX22-0012.
- *Resource Access Control Facility (RACF) Security Administrator's Guide*, SC28-1340.
- *System Programming Library: RACF*, SC28-1343.

SMP/E Publications

The following list shows books in the SMP/E Release 8 library.

- *SMP/E Diagnosis Guide*, SC23-3130.
- *SMP/E Messages and Codes*, SC28-1107.
- *SMP/E Reference*, SC28-1107.
- *SMP/E Reference Summary*, SX22-0016.
- *SMP/E User's Guide*, SC28-1302.

VSAM Publication

MVS/370 VSAM Administration Guide, GC26-4066.

X.25 NPSI Publications

The following list shows books in the X.25 NPSI library.

- *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning Version 3*, LY30-5610.
- *X.25 Network Control Program Packet Switching Interface Host Programming*, SC30-3502.
- *X.25 Network Control Program Packet Switching Interface Planning and Installation*, SC30-3470.

IBM Hardware Publications

The following sections describe the books associated with IBM hardware products.

System/370 and System/390 Publications

The following list shows the principles of operation manuals for the System/370 and System/390 processors.

- *IBM ESA/370 Principles of Operation*, SA22-7200.
- *IBM ESA/390 Principles of Operation*, SA22-7201.
- *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085.
- *IBM System/370 Principles of Operation*, GA22-7001.
- *S/360, S/370, and S/390 I/O Interface Channel to Channel Control Unit OEMI*, GA22-6974.

3172 Interconnect Controller Publications

The following list shows books in the IBM 3172 Interconnect Controller library.

- *IBM Interconnect Controller Program User's Guide*, SC30-3525.
- *IBM 3172 Interconnect Controller Installation and Service Guide*, GA27-3861.
- *IBM 3172 Interconnect Controller Operator's Guide*, GA27-3860.
- *IBM 3172 Interconnect Controller Planning Guide*, GA27-3867.
- *IBM 3172 Interconnect Controller Status Codes*, GA27-3951.

3270 Information Display System Publication

3270 Information Display System: 3270 Data Stream Programmer's Reference, GA23-0059.

8232 LAN Channel Station Publications

The following list shows books in the IBM 8232 LAN Channel Station library.

- *IBM LAN Channel Support Program: Version 1.0 User's Guide*, SC30-3458.
- *IBM 8232 LAN Channel Station: Installation and Testing*, GA27-3796.
- *IBM 8232 LAN Channel Station: Operating Guide*, GA27-3785.

9370 Publications

The following list shows books in the 9370 library.

- *IBM 9370 Information System: Using the X.25 Communications Subsystem*, SA09-1742.
- *IBM 9370 Information System X.25 Communications Subsystem Description*, SA09-1743.
- *VM/ESA: Connectivity Planning, Administration, and Operation Release 1*, SC24-5448.

Other TCP/IP-Related Publications

The following sections describe other books associated with TCP/IP.

- *The Art of Distributed Application: Programming Techniques for Remote Procedure Calls* John R. Corbin, Springer-Verlog, 1991.
- *CAE Specification: X/Open Transport Interface (XTI),* X/Open Company Ltd., U. K., 1992, SC31-8005.
- *IEEE Network Magazine*, July 1990.
- *TCP/IP Illustrated Volume I: The Protocols*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1994, SR28-5586.
- *TCP/IP Illustrated Volume II: The Implementation*, Gary R. Wright and Richard Stevens, Addison-Wesley Publishing Company, Inc., 1995, SR28-5630.
- *TCP/IP Illustrated Volume III*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1996, SR23-7289
- *Interoperability Report*, Volume 3, No. 3, March 1989.
- "MIB II Extends SNMP Interoperability," C. Vanderberg, *Data Communications*, October 1990.
- *"Network Management and the Design of SNMP,"* J.D. Case, J.R. Davin, M.S. Fedor, M.L. Schoffstall.
- *"Network Management of TCP/IP Networks: Present and Future,"* A. Ben-Artzi, A. Chandna, V. Warrier.
- *The Simple Book: An Introduction to Management of TCP/IP-based Internets*, Marshall T Rose, Prentice Hall, Englewood Cliffs, New Jersey,1993.
- "Special Issue: Network Management and Network Security," *ConneXions-The Interoperability Report* Volume 4, No. 8, August 1990.
- *UNIX Programmer's Reference Manual* (4.3 Berkeley Software Distribution, Virtual VAX-11 Version). Department of Electrical Engineering and Computer Science. University of California, Berkeley, 1988.

OSF/Motif Publications

The following list shows OSF/Motif books.

- *OSF/Motif Application Environment Specifications* (AES), Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640483-9.
- *OSF/Motif Programmer's Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640509-6.
- *OSF/Motif Programmer's Reference* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640517-7.
- *OSF/Motif Style Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640491-X.
- *OSF/Motif User's Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640525-8.

Sun (RPC) Publications

The following list shows Sun Microsystems books.

- *Networking on the Sun Workstation: Remote Procedure Call Programming Guide* (800-1324-03), Sun Microsystems, Inc.
- *Network Programming* (800-1779-10), Sun Microsystems, Inc.

X Window System Publications

The following list shows X Window System books.

- *Introduction to the X Window System*, Oliver Jones, Prentice-Hall, 1988, ISBN 0-13-499997-5.
- *PEXlib Specification and C Language Binding* Jeff Stevenson, Hewlett-Packard Company, 1992, SR28-5116.
- *The X Window System Series* (6 volumes), O'Reilly & Associates, 1988, 1989, 1990, ISBN 0-937175-40-4, 0-937175-27-7, 0-937175-28-5, 0-937175-35-6, 0-937175-33-1, 0-937175-35-8.
- *X Protocol Reference Manual* Adrian Nye, ed. O'Reilly & Associates, Inc., 1990, ISBN 0-937175-50-1.
- *X Window System: C Library and Protocol Reference* Robert Scheifler, James Gettys, and Ron Newman, DEC Press, 1988, ISBN 1-55558-012-2.
- *X Window System: Programming and Applications with Xt*, Douglas A. Young, Prentice-Hall, 1989, ISBN 0-13-972167-3.

- *X Window System: Programming and Applications with Xt, OSF/Motif Edition* Douglas A. Young, Prentice-Hall, 1990, ISBN 0-13-497074-8.
- *X Window System Technical Reference*, Steven Mikes, Addison-Wesley, 1990, ISBN 0-201-52370-1.
- *X Window System User's Guide* Valerie Quercia and Tim O'Reilly, O'Reilly & Associates, Inc., 1990, ISBN 0-937175-14-5.

Network Architecture Publications

The following sections list books associated with network architecture.

Open Systems Interconnection (OSI) Publication

Open Systems Interconnection, Z320-9757.

Systems Network Architecture (SNA) Publications

The following list shows books in the SNA library.

- *Systems Network Architecture: Sessions between Logical Units*, GC20-1868.
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112.
- *Systems Network Architecture Format and Protocol Reference Manual: Management Services*, SC30-3346.
- *Systems Network Architecture Formats* GA27-3136.
- *Systems Network Architecture Network Product Formats*, LY43-0081.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Communicating Your Comments to IBM

OS/390 TCP/IP OpenEdition
User's Guide

Publication No. GC31-8305-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:

  United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**
  - Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

OS/390 TCP/IP OpenEdition
User's Guide

Publication No. GC31-8305-00

If your concern is service related, you can reach Service at 1-800-992-4777 in the United States. Outside the United States, please check your phone listing for the IBM Service Center nearest you.

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Your Internet Address: _____

Name

Address

Company or Organization

Phone No.

Help us help you!
GC31-8305-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

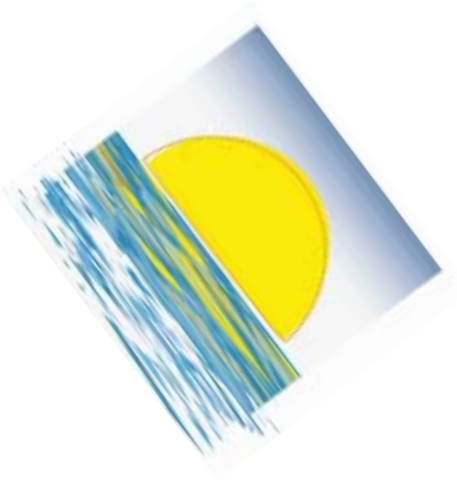
Information Development
Department CGMD
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S390-50
Program Number: 5645-001

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.



GC31-8305-00

Spine information:

OS/390 TCP/IP OpenEdition User's Guide