CDCNET
TCP/IP

Usage

Publication Number 60000214

CDCNET

TCP/IP

Usage

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60000214

Manual History

Revision	System Version/ PSR Level	Date
A	1.2.5/688	September 1987
B	1.3/700	April 1988

This manual is revision B, printed in April 1988. It reflects CDCNET Version 1.3 at PSR level 700, for operation on NOS/VE Version 1.3.1 and NOS Version 2.6.1. New features and changes documented in this revision include the following:

- TCP/IP Address Resolution Protocol (ARP) support
- TCP/IP Defense Data Network (DDN) support
- TCP/IP Exterior Gateway Protocol (EGP) support
- TCP/IP File Transfer Protocol (FTP) on all CYBER mainframes
- TCP/IP TELNET on all CYBER mainframes

Contents

Figures

Tables

About This Manual

The TCP/IP Usage manual describes how to access the Control Data® network and host-based services that interface to terminals and hosts on a Transmission Control Protocol/Internet Protocols (TCP/IP) network, using the Control Data Distributed Communications Network (CDCNET).

The services that implement the TCP/IP protocols on CDCNET are as follows:

- CDCNET TELNET
- CYBER FTP

CDCNET TELNET provides two types of terminal-to-host connections:

- Terminals connected to a TCP/IP host (supporting TELNET), can connect and access the interactive services of a CYBER host.
- Terminals connected to CDCNET can connect and access the interactive services of a TCP/IP host (supporting TELNET), such as a Digital Equipment Corporation (DEC) VAX® host computer or Sun® workstation.

CYBER FTP is the CYBER host utility that implements the TCP/IP File Transfer Protocol (FTP) on CYBER hosts. Two FTP products exists:

- FTP/VE is the NOS/VE utility that implements FTP on NOS/VE hosts.
- FTP/NOS is a collection of Network Access Method (NAM) applications that implements FTP on NOS hosts.

FTP/VE and FTP/NOS provides file transfer capabilities to and from TCP/IP hosts connected to the same network.

Audience

This manual is for the terminal user who may be connected to a TCP/IP host and need to make a connection and access the services of a CYBER host.

Additionally, the terminal user may be connected to CDCNET and want to make a connection and access the services of another TCP/IP host, such as a VAX host computer or Sun workstation.

This manual is based on three assumptions: first, that you are familiar with the terminal and connection attributes discussed in the CDCNET Access Guide; second, you know the service title you want to access, and third, you have some working knowledge and understanding of TCP/IP protocols.

Organization

This manual is organized as follows:

Chapter 1 introduces you to CDCNET TELNET and CYBER FTP.

Chapter 2 describes CDCNET TELNET. CDCNET TELNET is Control Data's implementation of the Defense Data Network (DDN) TELNET protocol on CDCNET.

Chapter 3 describes CYBER FTP. CYBER FTP is Control Data's implementation of TCP/IP FTP.

Appendix A is a glossary of frequently used terms.

A list of acronyms is also provided later in this section.

Conventions

The following conventions are used in this manual.

boldface	Denotes the required parts of a format.
italics	Denotes the optional parts of a format.
UPPERCASE	In formats, denotes the parts of the format that must be entered exactly as shown. In text, names are shown in uppercase.
lowercase	In formats, denotes the parts of the format that the user supplies.
blue	In interactive examples, blue is used to show user input.
`nonproportional typeface`	Denotes examples (the nonproportional typeface simulates computer output).
number base	All numbers are decimal unless otherwise indicated.

Related Manuals

The following Control Data manuals describe in greater detail some of the topics covered in this manual:

Manual	Publication Number	Online Title
Background (Access as needed):		
CDCNET Conceptual Overview	60461540	
CDCNET Manuals:		
CDCNET Access Guide	60463830	CDCNET_ ACCESS
CDCNET Terminal Interface Usage	60463850	
CDCNET Configuration and Site Administration Guide	60461550	
CDCNET Network Operations	60461520	
CDCNET Diagnostic Messages	60461600	CDCNET_MSGS
SCL Manuals:		
NOS/VE System Usage	60464014	
NOS/VE Network Management	60463916	
NOS Manuals:		
NOS Version 2 Reference Manual Volume 3	60459680	
NOS Version 2 Installation Handbook	60459320	
Miscellaneous:		
NOS/VE Diagnostic Messages	60464613	MESSAGES

Ordering Manuals

Control Data manuals are available through Control Data sales offices or from:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

Submitting Comments

Control Data welcomes your comments about this manual. Your comments may include your opinion of the usefulness of this manual, your suggestions for specific improvements, and the reporting of any errors you have found.

You can submit your comments using the comment sheet on the last page of this manual. If the comment sheet has already been used, you can mail your comments to:

Control Data Corporation
Technology and Publications Division
P.O. Box 3492
Sunnyvale, California 94088-3492

You can also submit your comments through SOLVER, an online facility for reporting problems. To submit a documentation comment through SOLVER, do the following:

1. Select `Report a problem or change an existing PSR` from the main SOLVER menu.
2. Respond to the prompts for site-specific information.
3. Select `Write a comment about a manual` from the new menu.
4. Respond to the prompts.

Please indicate whether or not you would like a written response.

In Case You Need Assistance

Control Data's CYBER Software Support maintains a hotline to assist you if you have a problem using our products. If you need help beyond that provided in the documentation or find that the product does not perform as described, call us at one of the following numbers and a support analyst will work with you.

From the USA and Canada: (800) 345-9903

From other countries: (612) 851-4131

The preceding numbers are for help with product usage. Address questions about the physical packaging and/or distribution of printed manuals to Literature and Distribution Services at the following address:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

or you can call (612) 292-2101. If you are a Control Data employee, call CONTROLNET® 243-2100 or (612) 292-2100.

Acronyms

ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
CDCNET	Control Data Distributed Communications Network
CCL	CYBER Control Language
DARPA	Defense Advanced Research Projects Agency
DDN	Defense Data Network
DEC	Digital Equipment Corporation
DOD	Department of Defense
EGP	Exterior Gateway Protocol
FTP	File Transfer Protocol
FTP/NOS	File Transfer Protocol/Network Operating System
FTP/VE	File Transfer Protocol/Virtual Environment
IAF	Interactive Facility
IP	Internet Protocol
MILNET	Military Network
NAM	Network Access Method
NOS	Network Operating System
NOS/VE	Network Operating System/Virtual Environment
PSR	Programming System Report
RFC	Request for Comments
SCL	System Command Language
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDI	Terminal Device Interface

Products and Trademarks

Other companies' products documented in this manual are:

Digital Equipment Corp.: VAX Computer System

Sun Microsystems, Inc.: Sun Workstation

Trademarks documented in this manual are:

Sun Workstation® is a trademark of Sun Microsystems, Inc.

UNIX® is a trademark of AT&T Bell Laboratories.

VAX® is a trademark of Digital Equipment Corp.

Introduction 1

Introduction 1

This manual describes how to access the Control Data network and host-based services that interface to terminals and hosts on a Transmission Control Protocol/Internet Protocol (TCP/IP) network, using CDCNET.

TCP/IP refers to a suite of protocols, that have evolved to support the ARPANET community. ARPANET is a government funded network developed many years ago by the Defense Advanced Research Projects Agency (DARPA).

Over the last few years, ARPANET has evolved into the Department of Defense's (DOD) major communications network and has been split into two major networks, ARPANET and MILNET. ARPANET supports research and development and MILNET supports operational communication requirements.

The Department of Defense now requires that new proposals use implementations of TCP/IP to support connectability to ARPANET and MILNET. Therefore, Control Data supports TCP/IP as a standard product. CDCNET TCP/IP supports connectability to:

- Remote and local workstations that use the TCP/IP protocols
- The TCP/IP community including the Defense Data Network (DDN) MILNET, ARPANET, and other networks.
- Supported protocols include Address Resolution Protocol (ARP), Exterior Gateway Protocol (EGP), and DOD standard X.25.

The TCP/IP protocols discussed in this manual include:

- CDCNET TELNET
- CYBER FTP

CDCNET TELNET implements the DDN TELNET protocol on CDCNET.

CYBER FTP is a TCP/IP-based file transfer protocol. The CYBER implementation includes support for both client and server modes.

The FTP host resident software in conjunction with CDCNET provides the capability to interchange files with other hosts connected to a TCP/IP network.

Figure 1-1 shows a typical CDCNET and TCP/IP network and the services available.

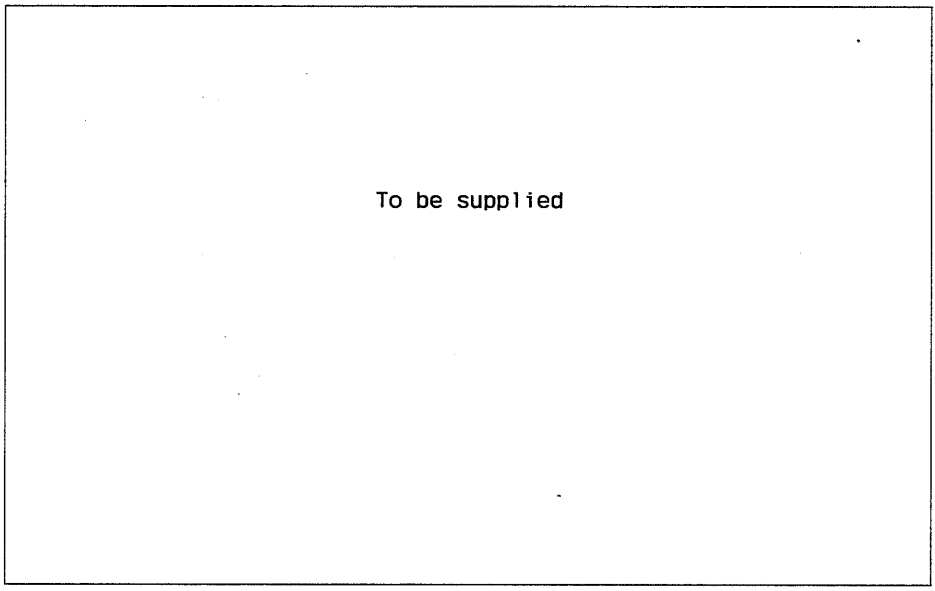

Figure 1-1. CDCNET and TCP/IP Network

CDCNET TELNET 2

CDCNET TELNET 2

CDCNET TELNET provides two types of terminal-to-host connections:

- Terminals connected to a TCP/IP host (supporting TELNET), can connect and access the interactive services of a CYBER host.
- Terminals connected to CDCNET can connect and access the interactive services of a TCP/IP host (supporting TELNET), such as a VAX host computer or Sun workstation.

The configurations listed below have been tested to verify their correct operation with a CYBER host. CDC supports other configurations by the Programming System Report (PSR) mechanism, but they have not been explicitly tested. The supported configurations are as follows:

- CYBER 930 host
- CYBER 180 host with NOS/VE
- CYBER 180 host with NOS/VE and NOS Dual State
- CYBER 180 host with NOS
- DEC VAX host with UNIX® 4.3 BSD
- Sun-3 workstations (Sun Release 3.4) with UNIX 4.2 BSD
- CYBER 910 workstations

CDCNET TELNET is based on the DARPA Internet protocols specified in Network Working Group Request for Comments (RFC) 854.

Introduction

CDCNET TELNET provides two gateways: the Server TELNET gateway and the User TELNET gateway.

The Server TELNET gateway allows TELNET host terminals to connect to the interactive services of a CYBER host. This manual refers to such a connection as a Server TELNET connection. The TELNET host involved in a Server TELNET connection is referred to as the User TELNET host.

The User TELNET gateway allows CDCNET terminals to connect to the interactive services of a TELNET host. This manual refers to this type of connection as a USER TELNET connection. The TELNET host involved in a User TELNET connection is referred to as the Server TELNET host.

The following sections describe how to make and manage Server TELNET and User TELNET connections.

CDCNET Server TELNET

A CDCNET Server TELNET connection connects a TELNET terminal to a CYBER host.

A Server TELNET connection initiates the TELNET terminal user, using the UNIX *telnet* command. The TELNET connection is made to the CDCNET Server TELNET gateway, which makes a corresponding connection to the CYBER host that it has been configured to support. The Server TELNET gateway maintains the association between the two connections, relaying commands and data between the two connections.

Figure 2-1 illustrates CDCNET's Server TELNET connection model.

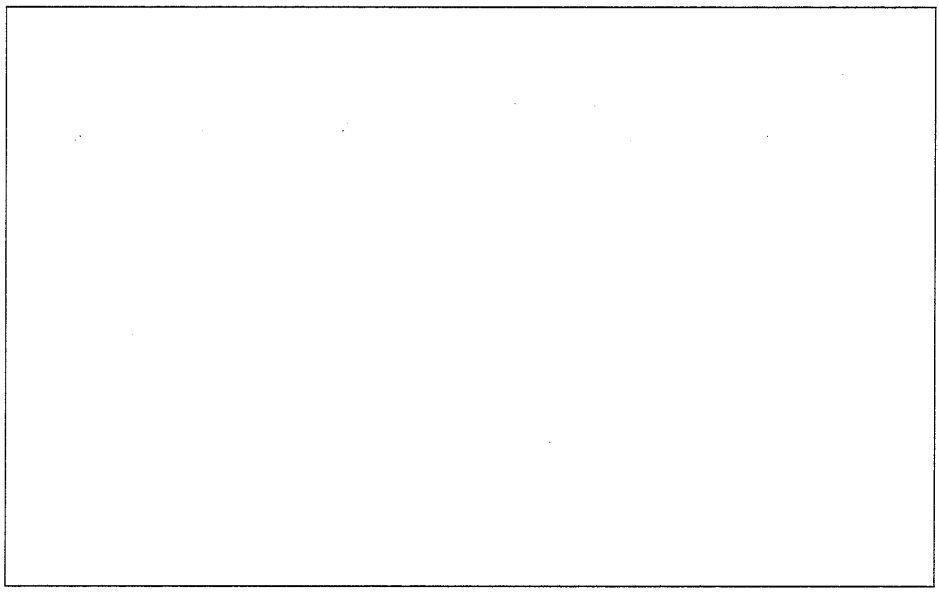

Figure 2-1. CDCNET's Server TELNET Connection Model

Connecting and Disconnecting From a CYBER Host

To connect to a CYBER host from a TELNET terminal, you use the UNIX *telnet* command. Your site administrator can provide you with the service name for each of the available CYBER hosts. Use the CYBER host service name as the value for the host parameter of the UNIX *telnet* command. Enter the following *telnet* command to access a CYBER host:

```
telnet name
```

After connecting to the CYBER host, you can login to the CYBER. Enter any validation information (such as username and password) required by the CYBER host and then proceed with your session.

When you are ready to stop processing on the CYBER host, logoff by entering the LOGOUT command. The CYBER host provides you with a summary of your session and then disconnects the connection for you.

Server TELNET Terminal and Connection Attributes

Table 2-1 summarizes the terminal and connection attributes that are supported when using a Server TELNET connection. Default settings are indicated in parentheses.

Table 2-1. Server TELNET CDCNET Attributes

Terminal Attributes	Connection Attributes
Attention_Character (NUL)	Attention_Character_Action (2)
Backspace_Character (BS)	Break_Key_Action (0)
End_Line_Character (CR)	Input_Block_Size (160)
End_Partial_Character (LF)	Input_Editing_Mode
End_Line_Positioning (NONE)	Partial_Character_Forwarding (OFF)
End_Partial_Positioning (NONE)	Transparent_Character_Mode (TERMINATE)
Echoplex (OFF)	Transparent_Forward_ Character (CR 8D(16))
Network_Command_Character (%)	Transparent_Length_Mode (NONE)
Terminal_Model (NONE)	Transparent_Message_Length (255)
	Transparent_Terminate_ Character (CR 8D(16))

Server TELNET Input Editing Modes

When the INPUT_EDITING_MODE attribute is set to TRANSPARENT, you cannot enter CDCNET terminal user interrupts because the NETWORK_COMMAND_CHARACTER is not recognized in TRANSPARENT mode.

A Server TELNET connection controls the host application with the INPUT_EDITING_MODE attribute. Applications, such as the NOS/VE SCL command interpreter, use the INPUT_EDITING_MODE attribute set to NORMAL. Full-Screen applications, such as NOS/VE EDIT_FILE and NOS Full Screen Editor, are set to TRANSPARENT. For more information on input editing modes, refer to the CDCNET Terminal Interface Usage manual.

Normal Input Editing Mode

When you activate an application that uses the INPUT_EDITING_MODE attribute set to NORMAL, the Server TELNET gateway puts your TELNET connection in line mode or character mode, depending on the setting of your ECHOPLEX terminal attribute.

If ECHOPLEX is OFF, which is the default, the Server gateway puts the TELNET connection in line mode. This causes your TELNET host to locally edit and accumulate your input until you enter a carriage return.

If ECHOPLEX is ON, the Server gateway puts the TELNET connection in character mode. This causes your TELNET host to transmit each character you enter to the Server gateway. The Server gateway accumulates and edits your input characters according to your terminal attribute settings. When the Server TELNET gateway detects an input forwarding character (END_LINE_CHARACTER or END_PARTIAL_CHARACTER), the accumulated data is sent to your host application.

Transparent Input Editing Mode

When you activate an application that uses the INPUT_EDITING_MODE attribute set to TRANSPARENT, the Server TELNET gateway puts your TELNET connection in character mode. This causes your TELNET host to transmit each character you enter to the Server gateway. The Server gateway accumulates the characters until one of the transparent forwarding conditions, selected by your application, is detected. Then your accumulated input is sent to your host application by the Server gateway.

Managing Your Server TELNET Connection

To manage a Server TELNET connection, the terminal user can:

- Issue the NOS/VE CHANGE_TERMINAL_ATTRIBUTES command or the NOS TRMDEF command.
- Issue commands directly to the Server TELNET host, such as the TELNET SEND command which can be used to send TELNET commands such as Are You There (AYT) and Interrupt Process (IP) to the Server TELNET gateway. The Server gateway converts these commands to the corresponding CDCNET user interrupts.
- Issue CDCNET terminal user interrupts to the Server TELNET gateway. These interrupts are sent on to the CYBER host application.

NOTE

The CDCNET terminal user commands are not available when using a Server TELNET connection. If entered, the Server gateway discards them and responds with the message Invalid User Interrupt.

Using the NOS/VE CHANGE_TERMINAL_ ATTRIBUTES Command

To change your CDCNET terminal attributes, use the NOS/VE command CHANGE_TERMINAL_ATTRIBUTES.

To change the network command character, enter:

```
/change_terminal_attributes network_command_character=esc
```

For more information on the CHANGE_TERMINAL_ATTRIBUTES command and the available terminal attributes, refer to the NOS/VE System Usage manual.

Using the NOS TRMDEF Command

To change your CDCNET terminal attributes, use the NOS command TRMDEF.

For example, to change the network command character, enter:

```
TRMDEF,NCC=X1B
```

For more information on the TRMDEF command and the available terminal attributes, refer to the NOS Reference Manual, Volume 3.

Using the UNIX stty Command

The UNIX command to set terminal options, *stty*, can set a number of terminal options, such as echo control and new line characters. Refer to UNIX documentation for details on how to use this command. The section titled Resolving Communication Problems in this chapter describes some circumstances when the *stty* command is used.

Signaling Your CYBER Host Application

The usual interaction with a CYBER host application involves a number of terminal input and application output sequences. NOS/VE, NOS, and CDCNET provide a mechanism for sending signals to the application which are different than normal input data. When using a Server TELNET connection, you can signal your application using any of the following methods:

- Entering the TELNET SEND command
- Entering a CDCNET Terminal User Interrupt
- Entering the CDCNET ATTENTION_CHARACTER

The following sections discuss these capabilities in further detail.

Using the TELNET SEND Command

The TELNET SEND command can send the following TELNET commands to the Server TELNET gateway:

Options are as follows:

Are You There (send AYT)

When the Server TELNET gateway receives this command, it is converted to a %E user interrupt and sent to the CYBER host. The CYBER host responds to this interrupt by sending you the status of your application.

Break (send BK)

When the Server TELNET gateway receives this command, it is converted to the user interrupt specified by the BREAK_KEY_ACTION (BKA) attribute and sent to the CYBER host. The response is dependent on the value of BKA. The default value for BKA is 0, resulting in the Server TELNET gateway issuing a %0 interrupt to the CYBER host.

Interrupt Process (send IP)

When the Server TELNET gateway receives this command, it is converted to the user interrupt specified by the ATTENTION_CHARACTER_ACTION (ACA) attribute and sent to the CYBER host. The response is dependent on the value of ACA. The default value for ACA is 2, resulting in the Server TELNET gateway issuing a %2 interrupt to the CYBER host.

Abort Output (send AO)

When the Server TELNET gateway receives this command, it discards the current output message from your application. If the output message being discarded consists of many network packets, some time is required to discard the complete message and you may experience some delay before receiving responses to subsequent commands.

Using CDCNET Terminal User Interrupts

The following user interrupts can be sent to the Server TELNET gateway.

Options are as follows:

%0

When the Server TELNET gateway receives this interrupt, all typed-ahead input is discarded.

%1

When the Server TELNET gateway receives this interrupt, it is sent on to the CYBER host. The CYBER host responds to this interrupt by suspending your application and sending you a suspended notification.

%2 through %9

When the Server TELNET gateway receives this interrupt, it is sent on to the CYBER host. The CYBER host responds to this interrupt by terminating your application and sending you a termination notification.

%A through %Z

When the Server TELNET gateway receives this interrupt, it is sent on to the CYBER host. The response is dependent on the alpha character entered. For example, %E causes an application status to be returned; %D causes the current application to be detached; %L causes the application log to be returned.

NOTE

Terminal user interrupts cannot be entered when the application has selected transparent mode.

Using the CDCNET ATTENTION_CHARACTER

An ATTENTION_CHARACTER attribute can be defined using the NOS/VE CHANGE_TERMINAL_ATTRIBUTES command or the NOS TRMDEF command. When the Server TELNET gateway receives the attention character, it converts it to the user interrupt specified by the ATTENTION_CHARACTER_ACTION (ACA) attribute and sends the interrupt to the CYBER host. The response is dependent on the value of ACA. The default value of ACA is 2, resulting in the Server TELNET gateway issuing a %2 interrupt to the CYBER host.

Echoing

The Server TELNET gateway echoes your input under the following conditions:

- You enter the UNIX *stty -echo* command. This instructs your TELNET host not to echo and causes it to request the Server TELNET gateway to echo. The Server TELNET gateway honors this request.
- You activate a CYBER host application that uses transparent mode. This causes the Server TELNET gateway to put your connection in character mode, which causes your TELNET host not to echo.
- You enter the NOS/VE CHANGE_TERMINAL_ATTRIBUTES command or the NOS TRMDEF command with the ECHOPLEX attribute set to ON, which is an explicit request to have the Server gateway echo.
- The CYBER host application requests secured input (no echo). This occurs only during your login sequence. In this case, the Server gateway does the echoing, but suppresses the echoing of your password.

Resolving Communication Problems

The following are communication problems that could occur while using a Server TELNET session.

Problems When Using CYBER Full Screen Applications

When a Server TELNET connection is being used to access a CYBER full-screen application, the connection could hang when the full-screen application is activated. The problem occurs if the CDCNET TELNET Server attempts to negotiate the TELNET binary option and receives no response from the User TELNET system. If your User TELNET system does not respond to TELNET's binary option, you will not be able to use CYBER full screen products with this release of CDCNET TELNET.

Some User TELNET Systems Do Not Process XON/XOFF Locally

When the TELNET connection is in character mode, some TELNET implementations do not process XON/XOFF flow control locally. Instead, the XOFF signal is sent on the TELNET connection as data. When this occurs, it usually results in lost data and/or locked keyboards. If your User TELNET system does not process XON/XOFF flow control locally when in character mode, and you encounter the symptoms described above, reducing the terminal's line speed may resolve the problem.

Response to %2 Through %9 Commands Can Be Delayed

The response to a %2 through %9 command may be delayed if entered while output is being transmitted to the terminal. Any output data in the TELNET connection must be flushed and is delivered to the terminal before the response to the %2 command. The delay varies, but usually no more than 2 screens.

Use Attention Character If Interrupt Process Is Not Available

When executing a CYBER full-screen application, you cannot enter any of the user interrupts sent using the % character. However, if your TELNET host allows you to send a TELNET IP command, the Server TELNET gateway maps it into a corresponding CDCNET user interrupt. If your TELNET host does not allow you to send the TELNET IP command, define a CDCNET ATTENTION_CHARACTER so you can terminate CYBER full screen applications.

CDCNET User TELNET

A CDCNET User TELNET connection connects a CDCNET terminal to a Server TELNET host.

The CDCNET CREATE_CONNECTION command initiates a User TELNET connection. You specify the title of a Server TELNET host on this command.

This command creates a connection to the CDCNET User TELNET gateway which, in turn, initiates a TELNET connection to the Server host. The User TELNET gateway maintains the association between the two connections, relaying data and commands between the two connections.

Figure 2-2 illustrates CDCNET's User TELNET Connection Model

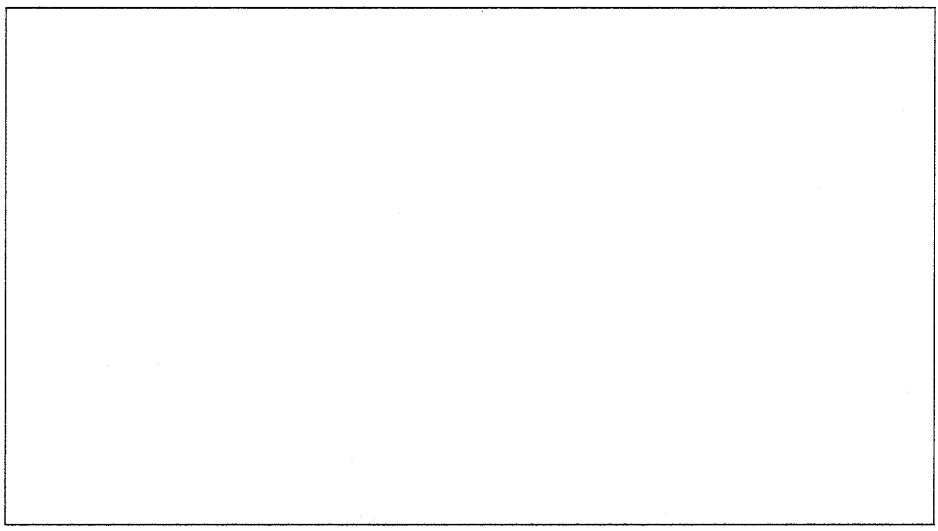

Figure 2-2. CDCNET User TELNET Connection Model

Connecting and Disconnecting From a Server TELNET Host

To connect to a Server TELNET host from a CDCNET terminal, your site administrator can provide you with a unique service name for each of the available Server TELNET hosts. Enter the following CDCNET command to access a service.

```
%create_connection service_name=name
```

After connecting to the Server TELNET host, you can login to the Server TELNET host in the normal manner. Enter any validation information (such as username and password) required by the service, and then proceed as though you were directly connected to that host.

When you are ready to stop processing on the Server TELNET host, enter the logoff command provided by the Server TELNET host. Alternatively, you may enter the CDCNET DELETE_CONNECTION command, but doing so prevents the User TELNET gateway from restoring your ECHOLPLEX attribute to its preconnection value.

For information on syntax conventions for the Server TELNET host, refer to that host's documentation.

User TELNET Connection Attributes

Table 2-2 summarizes the terminal and connection attributes that are supported when using a User TELNET connection. Default settings are indicated in parentheses.

Table 2-2. User TELNET CDCNET Attributes

Connection Attributes
Attention_Character_Action (2)
Break_Key_Action (0)
Input_Block_Size (160)
Input_Editing_Mode
Input_Output_Mode (FULLDUPLEX)
Partial_Character_Forwarding (OFF)
Store_Backspace_Character (OFF)
Store_Nuls_Dels (OFF)
Transparent_Character_Mode (FORWARD)
Transparent_Forward_Character (CR 8D(16))
Transparent_Length_Mode (NONE)
Transparent_Message_Length (255)
Transparent_Terminate_Character (CR 8D(16))
Transparent_Timeout_Mode (FORWARD)

User TELNET Input Editing Modes

When the INPUT_EDITING_MODE attribute is set to TRANSPARENT, you cannot enter CDCNET terminal user interrupts because the NETWORK_COMMAND_CHARACTER is not recognized in TRANSPARENT mode.

The INPUT_EDITING_MODE attribute of a User TELNET connection is controlled indirectly by the Server TELNET host selecting either line mode or character mode operation of the TELNET connection.

Normal Input Editing Mode

When a Server TELNET host places the TELNET connection in line mode, the User TELNET gateway sets the INPUT_EDITING_MODE attribute to NORMAL. This causes the CDCNET terminal device interface (TDI) to accumulate your input characters until you complete an input line by entering a carriage return. The completed input line is transmitted to the gateway and then onto the Server TELNET host. This mode of operation is appropriate for line oriented sessions like command interactions with the UNIX Shell program. Line mode does not work well with highly interactive programs because in order to send a single character to the TELNET host application for processing, the character must be followed by a carriage return.

Transparent Input Editing Mode

When a Server TELNET host places the TELNET connection in character mode, the User TELNET gateway sets the INPUT_EDITING_MODE attribute to TRANSPARENT and requests that the TDI forward each input character to the User TELNET gateway. The User gateway immediately forwards each character to the Server TELNET host. This mode allows the TELNET host application to process each character as it is entered. Character mode is appropriate for both line oriented sessions and full screen sessions like running the UNIX vi editor. For this reason, most Server TELNET hosts maintain the TELNET connection in character mode. One drawback of character mode is that CDCNET commands and user interrupts cannot be entered.

Managing Your User TELNET Connection

To manage a User TELNET connection, the terminal user can:

- Issue commands directly to the Server TELNET host, such as the UNIX *stty* command to set terminal options.
- Issue CDCNET terminal user commands to the CDCNET TDI. For example, the CHANGE_TERMINAL_ATTRIBUTE and CHANGE_CONNECTION_ATTRIBUTE commands can change terminal and connection attribute values.
- Issue CDCNET terminal user interrupts to the CDCNET TDI (e.g., %E and %2). These interrupts are sent to the User TELNET gateway which converts them into corresponding TELNET commands, such as AYT and IP.

Using the UNIX stty Command

The UNIX *stty* command can set a number of terminal options such as echo control and new line characters. Refer to UNIX documentation for details on the use of this command. The section titled Resolving Communication Problems later in this chapter describes some circumstances when it m ay be useful to set certain options using the *stty* command.

Using CDCNET Terminal User Commands

It is not necessary to change any of your terminal or connection attribute settings when using a User TELNET connection, in most cases. However, if you need to examine and/or change the values of terminal and connection attributes to accommodate characteristics of the Server TELNET host, you can do so using the following CDCNET commands:

- DISPLAY_TERMINAL_ATTRIBUTES
- DISPLAY_CONNECTION_ATTRIBUTES
- CHANGE_TERMINAL_ATTRIBUTES
- CHANGE_CONNECTION_ATTRIBUTES

The sections titled Echoing and Resolving Communication Problems identify situations when it may be appropriate to adjust terminal and/or connection attribute settings.

You can also enter any of the other CDCNET Terminal User commands. The CDCNET Terminal User commands are documented in the CDCNET Access Guide.

NOTE

When the User TELNET connection is being operated in transparent mode, it is necessary to break out of transparent mode before entering any CDCNET terminal user commands. Refer to the section titled Using CDCNET Terminal User Break Sequences later in this chapter.

Signaling Your Server TELNET Host

To send TELNET commands to the Server TELNET host, either:

- Enter a CDCNET Terminal User Interrupt
- Enter the Break Key or Attention Character

Using CDCNET Terminal User Interrupts

The following terminal user interrupts can be sent to the User TELNET gateway:

%1

When the User TELNET gateway receives this interrupt, it sends a TELNET Break command to the Server TELNET host.

%2 through %9

When the User TELNET gateway receives one of these interrupts, it sends a TELNET Interrupt Process command to the Server TELNET host.

%A

When the User TELNET gateway receives this interrupt, it sends a TELNET Abort Output command to the Server TELNET host.

%E

When the User TELNET gateway receives this interrupt, it sends a TELNET Are You There command to the Server TELNET host.

All other user interrupts received by the User TELNET gateway are discarded with no notification to the terminal user.

Using the Break Key and Attention Character

The Break Key and Attention Character can be used to send the TELNET Break and Interrupt Process commands to the Server TELNET host. Both the Break Key and Attention Character have corresponding connection attributes, BREAK_KEY_ACTION and ATTENTION_CHARACTER_ACTION. It is the value of the action attributes that determines whether a TELNET Break or TELNET Interrupt command is sent. If the action attribute has a value of 1, then a TELNET Break command is sent when the corresponding key is entered. If the action attribute has a value of 2, then a TELNET Interrupt Process command is sent when the corresponding key is entered.

Using CDCNET Terminal User Break Sequences

When you want to interrupt your application, the following break sequences can be used when you want to enter a CDCNET command or user interrupt but your User TELNET connection is in transparent mode:

```
<break><ctrl-x>      Transparent escape sequence.
<break><ctrl-c>      Switch to $command sequence.
```

These sequences can be entered using the Attention Character instead of the Break Key.

```
<attention_character><ctrl-x>      Transparent escape sequence.
<attention_character><ctrl-c>      Switch to $command sequence.
```

All of the terminal user break sequences are entered without a carriage return.

The following examples illustrate how these sequences can be used.

Figure 2-3 illustrates escaping from transparent mode, entering CDCNET commands, and then returning to transparent mode.

<break><ctrl-x>	Escape from transparent mode.
%dista<CR>	Display terminal attributes.
%e<CR>	Send AYT to server system.
%chaca iem=t<CR>	Return to transparent mode.
<CR>	Resync with server system.

Figure 2-3. Escaping From Transparent Mode

Figure 2-4 is the same command sequence as figure 2-4, except echoing is turned on before the commands are entered and turned off before returning to transparent mode. This enables you to see the commands as they are being entered.

<break><ctrl-x>	Escape from transparent mode.
%chata e=on<ELC>	Turn CDCNET local echo on.
%dista<CR>	Display terminal attributes.
%e<CR>	Send AYT to server system.
%chata e=off<CR>	Turn CDCNET local echo off.
%chaca iem=t<CR>	Return to transparent mode.
<CR>	Resync with server system.

Figure 2-4. Escaping From Transparent Mode With Echo

Figure 2-5 illustrates switching a User TELNET connection from character mode to line mode.

`stty -echo<CR>`	Turn server remote echo off.
`<break><ctrl-x>`	Escape from transparent mode.
`%chata e=on<CR>`	Turn CDCNET local echo on.

Figure 2-5. Switching From Character Mode to Line Mode

Figure 2-6 illustrates switching a User TELNET connection back to character mode.

`stty -echo<CR>`	Turn server remote echo off.
`%chaca e=off<CR>`	Turn CDCNET local echo off.
`%chata iem=t<CR>`	Return to transparent mode.

Figure 2-6. Switching From Line Mode to Character Mode

Figure 2-7 illustrates how you can display your CDCNET connections when your User TELNET connection is in the transparent mode.

`<break><ctrl-c>`	Switch to $command connection.
`disc`	Displays your connections.
`chawc $A`	Switch back to the User TELNET connection.

Figure 2-7. Displaying CDCNET Connections in Transparent Mode

Echoing

The User TELNET gateway never echos your input. Instead, it notifies the Server TELNET host when it should echo your input.

To indicate the type of echoing you prefer, set your ECHOPLEX terminal attribute before making the User TELNET connection.

If you set your ECHOPLEX attribute to ON before making the User TELNET connection, you indicate to the gateway that you want the Server TELNET host to echo your input.

If you set your ECHOLPEX attribute to OFF before making the User TELNET connection, you indicate to the gateway that you want the Server TELNET host not to echo your input.

When you make your User TELNET connection, the gateway reads your ECHOLPEX attribute in order to determine how to respond to echoing offers by the Server TELNET host.

When the gateway finds that your ECHOLPEX attribute is ON, it allows the Server TELNET host to echo. The gateway temporarily changes your ECHOLPEX attribute to OFF so that double echoing does not occur. When you logout from the Server TELNET host, the gateway turns your ECHOLPEX attribute back ON. Using the CDCNET DELETE_CONNECTION command prevents the User gateway from turning your ECHOPLEX attribute back ON.

When the gateway finds that your ECHOLPEX attribute is OFF, it assumes that you want either your terminal or the CDCNET TDI to echo your input. The gateway does not allow the Server TELNET system to echo.

Once you have made your User TELNET connection, you no longer can communicate your echoing preference to the gateway. You can however, issue commands to your TDI and Server TELNET host to achieve the desired echoing.

Resolving Communication Problems

The following are communication problems that could occur while using a User TELNET connection.

Line Folding Should be Turned Off

The Server system could transmit output data that includes embedded control characters and/or escape sequences. When output is formatted in this manner, CDCNET cannot properly perform the output operations of line folding, hold page, and end of page processing. The CDCNET attributes that control these output operations should be set to the following values before making the User TELNET connection:

```
Hold_Page        = OFF        (same as CDCNET default)
End_Page_Action  = NONE       (same as CDCNET default)
Fold_Line        = OFF        (CDCNET default is FL=ON)
```

Server System Must Terminate Output Lines With CR/LF

The default in some UNIX implementations is to terminate output lines with a line feed (LF) character only. This causes the output of a User TELNET connection to be incorrectly formatted since the lines of output are transmitted to the terminal without first positioning each line to the left margin. The problem can be corrected by entering the UNIX set terminal options *stty* command. The syntax is:

```
stty -nl<CR>
```

CYBER FTP 3

CYBER FTP 3

CYBER FTP provides file transfer capabilities to and from TCP/IP hosts connected to the same network. Two FTP products exist, the NOS/VE utility FTP/VE and a collection of NAM applications called FTP/NOS.

The configurations listed below have been tested to verify their correct operation with CYBER FTP. CDC supports other configurations using the PSR mechanism, but have they not been explicitly tested. The supported configurations are as follows:

- CYBER 930 host
- CYBER 180 host with NOS/VE
- CYBER 180 host NOS/VE and NOS Dual State
- CYBER 180 host with NOS
- DEC VAX host with UNIX 4.3 BSD
- Sun-3 workstations (Sun Release 3.4) with UNIX 4.2 BSD
- CYBER 910 workstations

Introduction

The File Transfer Protocol (FTP) is a DDN-defined protocol. It provides file transfer capabilities to and from connected TCP/IP hosts.

CYBER FTP supports the FTP Client and the FTP Server protocols. FTP Client provides CYBER users access to file systems on remote hosts. FTP Server provides access from remote hosts to the CYBER file systems.

The hosts where the FTP Client and the associated FTP Server execute are referred to as local host and remote host, respectively. CYBER FTP allows the CDCNET user to transfer files between hosts on the same network. A remote host can be any host system that supports the FTP protocol. Most implementations of TCP/IP support the FTP protocol.

In addition, CYBER FTP allows you to access directories/files on a remote host and to perform common operations, such as list and change working directories, list files at various levels, and rename directories and files.

To use CYBER FTP to communicate with a remote host on the network:

- Both the local and remote hosts must support DDN's standard FTP.
- The FTP Server program must be running on the remote host.
- You must use the remote host's conventions for specifying files.

Figure 3-1 illustrates how files are transferred from a local host (CYBER) to a remote host (VAX, Sun) using CYBER FTP.

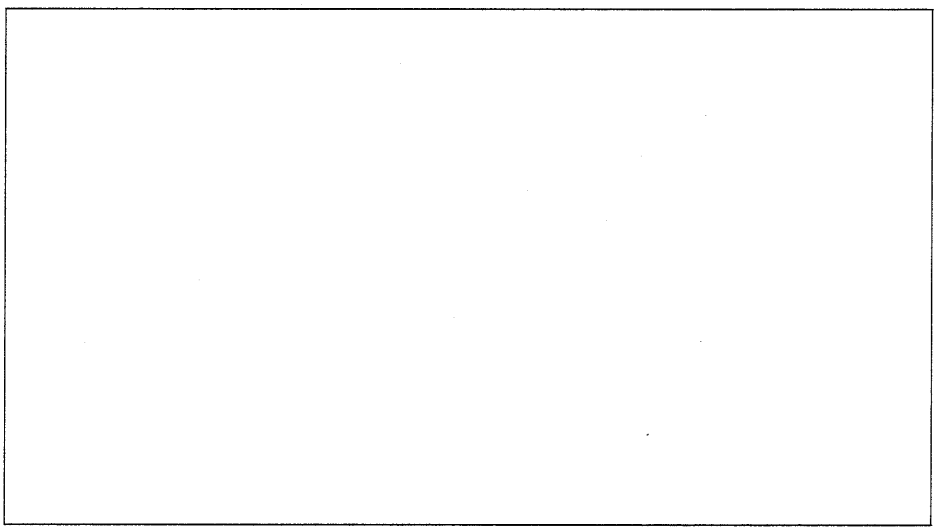

Figure 3-1. Transferring Files Using the CYBER FTP Client

Figure 3-2 illustrates how files are transferred from a local host (VAX, Sun) to a remote host (CYBER) using CYBER FTP.

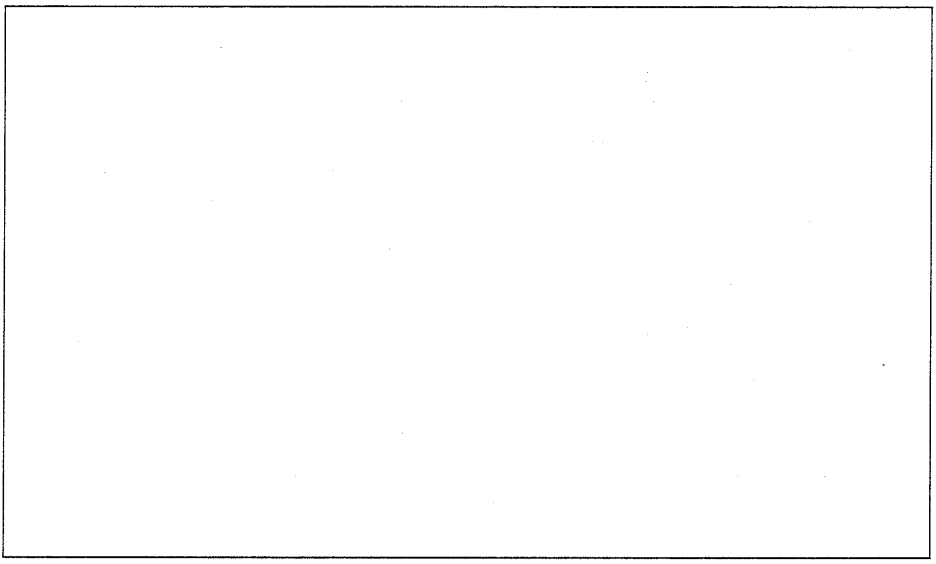

Figure 3-2. Transferring Files Using the CYBER FTP Server

All CYBER FTP operations, interactive and batch, are initiated by first invoking the FTP Client. To invoke the FTP Client, use the CREATE_CLIENT_FTP_CONNECTION command utility for NOS/VE, and the FTP control statement for NOS.

Appropriate CYBER FTP subcommands are then supplied to the FTP Client to perform the desired operation.

The CYBER FTP subcommands are as follows:

APPEND_FILE
CHANGE_AUTO_LOGIN_MODE
CHANGE_BELL_MODE
CHANGE_DEBUG_MODE
CHANGE_EXPRESSION_EVALUATION (FTP/VE only)
CHANGE_FILE_NAME
CHANGE_INPUT_FILE
CHANGE_OUTPUT_FILE
CHANGE_PORT_COMMAND_MODE
CHANGE_TRANSFER_TYPE
CHANGE_VERBOSE_MODE
CHANGE_WORKING_DIRECTORY
CREATE_DIRECTORY
CREATE_HOST_CONNECTION
DEFINE_AUTO_LOGIN
DELETE_DIRECTORY
DELETE_FILE
DELETE_HOST_CONNECTION
DISPLAY_COMMAND_INFORMATION
DISPLAY_DIRECTORY
DISPLAY_FILE_NAMES
DISPLAY_FTP_OPTIONS
DISPLAY_LOCAL_HELP
DISPLAY_REMOTE_HELP
DISPLAY_WORKING_DIRECTORY
ELEVATE_WORKING_DIRECTORY
GET_FILE
LOGIN_USER
QUIT
REPLACE_FILE
SEND_FTP_COMMAND

CYBER FTP Client User Interface

This section documents the use of the SCL utility CREATE_CLIENT_FTP_CONNECTION (CRECFC) for NOS/VE, the control statement FTP (CRECFC) for NOS and describes each subcommand supported by CYBER FTP.

The utility session begins when you enter the CREATE_CLIENT_FTP_CONNECTION (CRECFC) command for NOS/VE, or the control statement FTP (CRECFC) for NOS, and ends when you enter a QUIT subcommand. The subcommand descriptions are presented in alphabetical order.

File Name Processing

Many FTP subcommands must specify file names. On NOS/VE, the interpretation of the file name value specified depends on the value specified for the EXPRESSION_EVALUATION parameter on the subcommand. File names are evaluated as parameter type application value (LOCAL_NAME or FOREIGN_NAME) if EXPRESSION_EVALUATION is OFF, and as parameter type STRING, if it is ON.

Both types preserve case, which is significant for file names in certain operating systems, and allow some remote host file names that do not conform to SCL conventions to be specified without quotation marks.

For FTP/VE, the following characters cannot be present in a file name unless the file name is enclosed in quotation marks:

```
( ) , ; " space
```

For FTP/NOS, the following characters cannot be present in a file name unless the file name is enclosed in quotation marks:

```
, " space
```

One effect of disabling the EXPRESSION_EVALUATION parameter is that expressions such as $value(file) within SCL procedures are not evaluated when used as a parameter to a CRECFC subcommand.

FTP/NOS does not support the EXPRESSION_EVALUATION parameter because the SCL substitutions affected by this parameter are not available on NOS. File name specification in either mode's format is accepted directly.

Local Host File Names

For FTP/VE, local host file names must conform to type LOCAL_NAME. The name can be any NOS/VE file name or the UNIX acronym - (hyphen). The hyphen character causes either $INPUT or $OUTPUT to be used, depending on the context.

Local host file names must conform to type FOREIGN_NAME.

NOTE

Local host files refer to files residing on the local host, as opposed to foreign files residing on the remote host. Local host files do not refer to files residing in the NOS/VE catalog $LOCAL.

Remote Host File Names

Remote host file names must conform to type FOREIGN_NAME.

FTP/NOS File Format

All files manipulated by the FTP/NOS Client, with the possible exception of the file being transferred, are in ASCII 6/12 zero-byte-terminated system default record type format.

Prolog Execution

When CYBER FTP is invoked, it attempts to execute a user prolog file. The default CYBER FTP prolog file is $USER.FTP_PROLOG for FTP/VE and FTPPRLG for FTP/NOS. The prolog can be used when establishing alternate default values for the CYBER FTP operating environment and for defining the automatic login data required for access to a connected host. This may include user name, password, family, account number and project number.

The family and project number are not supported by FTP/VE for the R1.3 release. For more information, refer to the description of the AUTO_LOGIN_MODE parameter.

Command Formats

The CYBER FTP commands and subcommands use the SCL command syntax. For a complete description of the SCL command syntax, refer to the NOS/VE System Usage manual. The format of the CYBER FTP commands and subcommands as follows:

CYBER FTP COMMAND or
CYBER ALIAS or
BERKELEY UNIX ALIASES or
RFC 959 ALIAS
  REQUIRED PARAMETER_NAMES = scl_type
  OPTIONAL PARAMETER_NAMES = scl_type

NOTE

Only some commands have an RFC 959 alias.

RFC (Request for Comments) specifies a standard used by the ARPA (Advanced Research Projects Agency) community.

If a Berkeley alias matches an RFC alias, only one alias is specified in the parameter list.

Sometimes a single FTP command has more than one Berkeley UNIX alias; in these cases, all aliases are specified.

Command Descriptions

Each command or subcommand description in this manual provides the following information.

- Command or subcommand name.
- Brief statement of the command or subcommand function.
- Format including both singular and plural forms of the command or subcommand name with its abbreviation and the position of each parameter in the parameter list.
- Parameter value types. Within the command or subcommand format, each parameter name is equated to a word indicating the parameter value type. For more information on parameter value types, refer to the NOS/VE System Usage manual.
- Parameter name forms. Each parameter description lists the singular and plural forms of the parameter name and any abbreviations.
- Brief statement of the function of each parameter.
- Indication of whether the parameter is **required** or *optional* and, if optional, the default value.
- Additional remarks on command or subcommand options and processing details.
- Brief example using the command or subcommand.

CYBER FTP Initiation

Purpose

CYBER FTP is initiated by entry of the CREATE_CLIENT_FTP_CONNECTION (CRECFC) command for NOS/VE and the FTP control statement (command) for NOS. Its execution is controlled by a number of command parameters, some of which allow selection of the defaults in effect during execution. Their value can be changed at any time within CYBER FTP since each command parameter that sets a default is also associated with a subcommand.

Each command to CYBER FTP is processed as an SCL syntax subcommand for the duration of its execution. For FTP/VE each of the subcommands has an SCL STATUS parameter, allowing interception and processing of errors at the command level. The STATUS parameter is not supported by FTP/NOS.

For FTP/NOS, the processor of the FTP command does not support positional parameter value specification. Parameter names can only be specified on the FTP command using the abbreviated form, and the INPUT, OUTPUT, and PROLOG parameters are the only ones supported on the command.

Format

FTP/VE Format:

CREATE_CLIENT_FTP_CONNECTION or
CRECFC or
FTP
HOST or *KEY_HELP* = *name*
AUTO_LOGIN_MODE = *boolean*
VERBOSE_MODE = *boolean*
BELL_MODE = *boolean*
DEBUG_MODE = *boolean*
PORT_COMMAND_MODE=*boolean*
EXPRESSION_EVALUATION = *boolean*
OUTPUT = *file*
INPUT = *file*
PROLOG = *file*
STATUS = *var of status*

Format

FTP/NOS Format:

FTP or
CRECFC
HOST or *KEY_HELP* = *name*
AUTO_LOGIN_MODE = *boolean*
VERBOSE_MODE = *boolean*
BELL_MODE = *boolean*
DEBUG_MODE = *boolean*
PORT_COMMAND_MODE = *boolean*
EXPRESSION_EVALUATION = *boolean*
OUTPUT = *file*
INPUT = *file*
PROLOG = *file*
STATUS = *var of status*

Parameters

HOST (H)

Allows a remote host to be specified.

If this parameter is omitted, a connection can be established with a remote host later using the CREATE_HOST_CONNECTION subcommand.

AUTO_LOGIN_MODE (ALM)

Specifies whether an attempt should be made to initiate automatic login when a connection to a remote host has been established.

Setting AUTO_LOGIN_MODE to ON, TRUE, or YES enables AUTO_LOGIN_MODE; setting mode to OFF, FALSE, or NO disables AUTO_LOGIN_MODE.

This parameter allows automatic login to a remote host by specifying the host name only on the CYBER FTP command if all required login information for the remote host is available.

Login information (username, password, family, account and project) for a remote host is established by the DEFINE_AUTO_LOGIN subcommand. To use the automatic login feature, a CYBER FTP prolog file can be specified which includes a DEFINE_AUTO_LOGIN subcommand for each host with which communication is required in a later session. CYBER FTP uses the defined information when a connection to the remote host is established and AUTO_LOGIN_MODE is ON.

If AUTO_LOGIN_MODE is enabled and no login information for the specified host has been made available by the DEFINE_AUTO_LOGIN subcommand when the connection to the remote host is established, the user is prompted for username, password, and, if required, account.

For FTP/NOS, if the login is to a non-default family on a remote CYBER host, the family name must follow the password and be prefixed with a separator comma. If a non-default project is required on a remote CYBER host, the project number must follow the account and be prefixed with the separator comma.

If this parameter is omitted, the default is ON.

VERBOSE_MODE (VM)

Indicates whether FTP transaction messages and transfer rate information are to be written to the output file.

If this parameter is omitted, the default is OFF.

BELL_MODE (BM)

Indicates whether an ASCII BEL character is sent to interactive terminals with the input prompt.

This parameter can be used during long file transfers. The audible prompt allows you to perform concurrent tasks.

If this parameter is omitted, the default is OFF.

DEBUG_MODE (DM)

Indicates whether debug messages are to be written to the output file for each FTP command/reply transaction between the client and server FTPs. Debug messages are also logged in the job log by FTP/VE and the job dayfile by FTP/NOS.

If this parameter is omitted, the default is OFF.

NOTE

This feature is provided to aid experienced FTP network system analysts in resolving FTP interoperability problems with remote systems.

PORT_COMMAND_MODE (PCM)

Specifies whether CYBER FTP sends an FTP PORT command to the remote host prior to establishing a data connection for each data transfer. (The remote host then initiates a TCP active connect to the specified data port for the data transfer.) This is used to eliminate delays between subsequent file transfers from the same remote host by the same instance of CYBER FTP.

If this parameter is not specified as OFF, file transfers take place using the default port. A delay less than or equal to the TCP disconnect quiet time may occur in establishing a data connection while performing successive file transfers.

If this parameter is omitted, the default is ON.

EXPRESSION_EVALUATION (EE)

Specifies whether remote host file name parameters are required to be specified as SCL type STRING enclosed in apostrophes. This parameter must be set to ON to allow file name substitution in CYBER FTP subcommands.

If CYBER FTP is used interactively, set this parameter to OFF. For more information, refer to File Name Processing.

The EXPRESSION_EVALUATION parameter is not supported by FTP/NOS.

If this parameter is omitted, the default is OFF.

OUTPUT (O)

Allows specification of the output file. Responses to some CYBER FTP subcommands and all verbose and debug information are written to it.

If this parameter is omitted, the default is $OUTPUT for FTP/VE and OUTPUT for FTP/NOS.

INPUT (I)

Allows specification of the input file.

If FTP/VE is initiated from within a NOS/VE SCL procedure, CYBER FTP subcommands are not read from the specified file; they are obtained from the file

$COMMAND. For FTP/NOS, subcommands and input prompted for by subcommand execution are always read from the current input file.

If this parameter is omitted, the default is $INPUT for FTP/VE and INPUT for FTP/NOS.

PROLOG (P)

Allows the user to specify a prolog file.

Values specified in the prolog file take precedence over those on the CYBER FTP command for FTP/NOS.

Before being read, the file is rewound and cannot be assigned to a terminal. If the file is not local to the job for FTP/NOS, it attempts to acquire a permanent file of the same name.

If this parameter is omitted, the default is $USER.FTP_PROLOG for FTP/VE and FTPPRLG for FTP/NOS.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

The STATUS parameter is not supported by FTP/NOS.

Remarks

- CYBER FTP prompts for user input with ftp/.
- When CYBER FTP prompts for login information, it reads the current input file.
- FTP/NOS responds to a user_break_1 by terminating execution of the current subcommand.
- FTP/NOS responds to a user_break_2 by terminating execution of the current subcommand and aborting the FTP job step.
- FTP/VE turns off echoplex mode for interactive jobs when prompting for passwords. However, if the terminal supports local echoplex, it must also be disabled in order to inhibit the password being echoed. FTP/NOS cannot provide this level of security because of restrictions imposed by the Interactive Facility (IAF).

- If the keyword HELP is specified on the HOST parameter, CYBER FTP executes the DISPLAY_LOCAL_HELP subcommand. For more information, refer to the DISPLAY_LOCAL_HELP subcommand.

- The CYBER FTP Client and Server each have a timeout set to 5 minutes. Other implementations of FTP have their own timeout values.

  If a message is sent to a peer FTP and a response is not received within the timeout interval, CYBER FTP considers the condition abnormal and responds to the user with an abnormal status.

FTP Subcommands

For all of the following subcommand descriptions:

- The default parameter value listed is the value that is in effect if the value has not been changed with a parameter on the CYBER FTP command, a subcommand in the prolog file, or a previous use of the subcommand.
- The STATUS parameter is not supported by FTP/NOS. If the parameter is present, the parameter is ignored.
- Lists are not supported by FTP/NOS.
- Positional parameter value specification on subcommands is supported by FTP/NOS.

APPEND_FILE

Purpose Appends data from one or more local host files to one or more corresponding remote host files. If the remote host files do not exist, they are created by the remote host's server.

NOTE

A local host file refers to a file resident on the local host and not to a file in the NOS/VE $LOCAL catalog.

Format **FTP/VE Format:**

APPEND_FILE or
APPF or
APPEND or
APPE
LOCAL_FILE = list of local_name
REMOTE_FILE = list of foreign_name
STATUS = var of status

Format **FTP/NOS Format:**

APPEND_FILE or
APPF or
APPEND or
APPE
LOCAL_FILE = local_name *or*
LOCAL_FILE = ' local name *CS=cset FSC=n readeoi* **'**
REMOTE_FILE = foreign_name
STATUS = var of status

Parameters **LOCAL_FILE (LF)**

Name of one or more local host files. The LOCAL_FILE parameter is implemented as a local_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

REMOTE_FILE (RF)

Name of one or more remote host files. The REMOTE_FILE parameter is implemented as a foreign_name parameter whose actual type depends on the value of EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

CS (FTP/NOS only)

Specifies the codeset, data character size and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command is ASCII (A).

The following NOS FCOPY command compatible values are supported:

Value	Description
ASCII (A)	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 (8)	ASCII 8/12 file with TELNET end_of_line terminators.
ASCII88 (88)	ASCII 8/8 file with TELNET end_of_line terminators.
DIS (D)	Display Coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

FSC (FTP/NOS only)

Specifies the number of files to be skipped before starting data transfer to the remote host. If this parameter is specified, n NOS end-of-file (EOF) marks are skipped before data transfer begins. Data transfer begins at file n+1 of a multi-file set.

If this parameter is omitted, the default is zero. (Data transfer starts from the beginning-of-information (BOI) of the file.)

READEOI (FTP/NOS only)

Specifies read to end-of-information (EOI). If this parameter is specified, all files of a multi-file set from the current file position to the EOI of the file are transferred with NOS EOF marks ignored.

If this parameter is omitted, the default is to transfer only one file.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Remarks

- If the REMOTE_FILE parameter is not specified, the APPEND_FILE subcommand uses the value(s) supplied in the LOCAL_FILE parameter as the remote host file name(s). If the REMOTE_FILE parameter is specified, the list must contain the same number of elements as the LOCAL_FILE parameter.
- No CYBER FTP command performs the opposite of an APPEND_FILE subcommand; remote files cannot be appended to local files.

Examples

The following subcommand appends a file resident on the local host to a file of the same name on the remote host.

```
ftp/append_file local_file=test
ftp/
```

CHANGE_AUTO_LOGIN_MODE

Purpose Controls automatic login operations when a connection to a remote host has been established. For more information, refer to the CYBER FTP command parameter AUTO_LOGIN_MODE.

Format **CHANGE_AUTO_LOGIN_MODE** or
CHAALM
MODE = boolean
STATUS = var of status

Parameters **MODE (M)**

Setting the MODE parameter to ON, TRUE, or YES enables auto login mode; setting the MODE parameter to OFF, FALSE, or NO disables auto login mode.

The default is ON.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand attempts automatic login when a connection to a host has been established.

```
ftp/change_auto_login_mode mode=on
ftp/
```

CHANGE_BELL_MODE

Purpose Controls the sending by CYBER FTP of an ASCII BEL character to interactive terminals with the input prompt. For more information, refer to the CYBER FTP command parameter BELL_MODE.

Format

CHANGE_BELL_MODE or
CHABM or
BELL
MODE = boolean
STATUS = var of status

Parameters **MODE (M)**

Setting the MODE parameter to ON, TRUE, or YES enables bell mode; setting the MODE parameter to OFF, FALSE, or NO disables bell mode.

The default is OFF.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand causes the bell to sound when the ftp/ prompt is displayed on a terminal.

```
ftp/change_bell_mode mode=on
ftp/
```

CHANGE_DEBUG_MODE

Purpose Controls the display of debug mode output. For more information, refer to the CYBER FTP command parameter DEBUG_MODE.

Format

CHANGE_DEBUG_MODE or
CHADM or
DEBUG
MODE = boolean
STATUS = var of status

Parameters

MODE (M)

Setting the MODE parameter to ON, TRUE, or YES enables debug mode; setting the MODE parameter to OFF, FALSE, or NO disables debug mode.

The default is OFF.

STATUS

Specifies a status variable to receive the termination condition for the subcommand.

Examples The following subcommand causes debug messages to be written to the output file.

```
ftp/change_debug_mode mode=on
ftp/
```

CHANGE_EXPRESSION_EVALUATION

Purpose Controls the way SCL evaluates file name parameters for FTP/VE. For more information, refer to the CYBER FTP command parameter EXPRESSION_EVALUATION. For more information, refer to File Name Processing.

Format

CHANGE_EXPRESSION_EVALUATION or
CHAEE
MODE = boolean
STATUS = var of status

Parameters

MODE (M)

Setting the MODE parameter to ON, TRUE, or YES enables expression evaluation; setting the MODE parameter to OFF, FALSE, or NO disables expression evaluation.

The default is OFF.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand causes file name parameters to be evaluated as SCL string values.

```
ftp/change_expression_evaluation mode=on
ftp/
```

CHANGE_FILE_NAME

Purpose Renames files on the remote host. This subcommand changes the name of multiple files in one command. There must be the same number of list elements for both the FILE and NEW_FILE_NAME parameters.

NOTE

To change the name of a permanent file on the local host use the NOS/VE command CHANGE_CATALOG_ENTRY (CHACE) or the NOS command CHANGE.

Format

CHANGE_FILE_NAME or
CHAFN or
RENAME
FILE = list of foreign_name
NEW_FILE_NAME = list of foreign_name
STATUS = var of status

Parameters

FILE (F)

Name of one or more remote host files. The FILE parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

NEW_FILE_NAME (NFN)

Specifies one or more new names for the remote host files.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand renames a file on the remote host.

```
ftp/change_file_name file=test new_file_name=test1
ftp/
```

CHANGE_INPUT_FILE

Purpose Allows specification of a new default input file. For more information, refer to the CYBER FTP command parameter INPUT.

Format

CHANGE_INPUT_FILE or
CHAIF
INPUT = file
STATUS = var of status

Parameters **INPUT (I)**

For FTP/VE, the input file is used only for reading the username, password, and account information when logging in to a remote host. If FTP/VE is initiated from within an SCL procedure, CYBER FTP subcommands are read from the $COMMAND file.

For FTP/NOS, subcommands and input prompted for by subcommand execution are always read from the current input file.

The default is $INPUT for FTP/VE and INPUT for FTP/NOS.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand specifies a new default input file.

```
ftp/change_input_file input=test
ftp/
```

CHANGE_OUTPUT_FILE

Purpose Allows the specification of a new output file. For more information, refer the CYBER FTP command parameter OUTPUT.

Format

CHANGE_OUTPUT_FILE or
CHAOF
OUTPUT = **file**
STATUS = var of status

Parameters

OUTPUT (O)

Specifies a new output file.

The default is $OUTPUT for FTP/VE and OUTPUT for FTP/NOS. CRECFC commands are written to the default output file $OUTPUT.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Remarks For FTP/NOS, the current output file is flushed by writing an EOR to it.

Examples The following subcommand specifies a new output file.

```
ftp/change_output_file output=testout
ftp/
```

CHANGE_PORT_COMMAND_MODE

Purpose Specifies if the CYBER FTP PORT command should be sent to the remote host prior to a data transfer.

Format **CHANGE_PORT_COMMAND_MODE** or **CHAPCM**

MODE = boolean
STATUS = var of status

Parameters **MODE (M)**

Setting the MODE parameter to ON, TRUE, or YES enables port command mode; setting the MODE parameter to OFF, FALSE, or NO disables port command mode.

The default is ON.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand disables sending of the FTP PORT command to the remote host prior to each data transfer.

```
ftp/change_port_command_mode mode=off
ftp/
```

CHANGE_TRANSFER_TYPE

Purpose Defines to CYBER FTP the data type of the file being transferred. When an FTP session is started, the default transfer type is ASCII.

Format

CHANGE_TRANSFER_TYPE or
CHATT or
TYPE
TYPE = keyword
STATUS = var of status

Parameters

TYPE

Specifies the CYBER data type of the file to be transferred.

Value	Description
ASCII (A)	ASCII text file. (FTP/VE)
ASCII (A)	ASCII 6/12 file with zero-byte-terminated records. (FTP/NOS).
BINARY (B) or IMAGE (I)	A mass storage file (segment access for NOS/VE) containing binary data, represented in 8-bit bytes.

The default is ASCII.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand changes the desired data type to IMAGE.

```
ftp/change_transfer_type type=image
ftp/
```

CHANGE_VERBOSE_MODE

Purpose Controls the display of VERBOSE MODE output. For more information, refer to the CYBER FTP command parameter VERBOSE_MODE.

Format

CHANGE_VERBOSE_MODE or
CHAVM or
VERBOSE
MODE = boolean
STATUS = var of status

Parameters **MODE (M)**

Setting mode to ON, TRUE, or YES enables VERBOSE_MODE; setting mode to OFF, FALSE, or NO disables VERBOSE_MODE.

The default is OFF.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand writes FTP transaction messages and transfer rate information to the output file.

```
ftp/change_verbose_mode mode=on
ftp/
```

CHANGE_WORKING_DIRECTORY

Purpose Changes the working directory on the remote host. This subcommand reduces the number of path elements that must be supplied for names in a hierarchical file system.

NOTE

To change the local working directory, use the NOS/VE command SET_WORKING_CATALOG or SETWC.

Format

CHANGE_WORKING_DIRECTORY or
CHAWD or
CD or
CWD
NAME = foreign_name
STATUS = var of status

Parameters **NAME (N)**

The name of a directory on the foreign host. For FTP/VE, the NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand changes the name of the working directory on the remote host.

```
ftp/change_working_directory name=/work
ftp/
```

CREATE_DIRECTORY

Purpose Creates one or more directories on the remote host.

NOTE

This command is a recent addition to the FTP protocol and may not be supported in some existing FTP Servers.

Format

CREATE_DIRECTORY or
CRED or
MKDIR or
MKD
NAME = list of foreign_name
STATUS = var of status

Parameters

NAME (N)

One or more foreign names. The NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand creates a directory on the remote host.

```
ftp/create_directory name=/a/eg/test
ftp/
```

CREATE_HOST_CONNECTION

Purpose Establishes a connection with a remote host. An existing host connection must be deleted with the DELETE_HOST_CONNECTION subcommand before a new connection to another host can be made.

Format

CREATE_HOST_CONNECTION or
CREHC or
OPEN
HOST = name
AUTO_LOGIN_MODE = boolean
STATUS = var of status

Parameters **HOST (H)**

The name of the remote host where a connection is to be established.

AUTO_LOGIN_MODE (ALM)

Specifies whether an automatic login attempt should be made when a connection to a host has been established.

Setting AUTO_LOGIN_MODE to ON, TRUE, or YES enables AUTO_LOGIN_MODE; setting AUTO_LOGIN_MODE to OFF, FALSE, or NO disables AUTO_LOGIN_MODE.

If automatic login is enabled and no login information for the specified host has been made available by the DEFINE_AUTO_LOGIN subcommand, then a prompt is issued for username, password, and if required, account. Automatic login information is established with the DEFINE_AUTO_LOGIN subcommand. For more information, refer to the DEFINE_AUTO_LOGIN subcommand.

If this parameter is omitted, the default is ON.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand establishes a connection with the remote host.

```
ftp/create_host_connection host=newhost
User      (Remote Host: NEWHOST):
? username
Password  (Remote Host: NEWHOST):
? password
ftp/
```

DEFINE_AUTO_LOGIN

Purpose Associates login information with a host name, so when a connection to that host is established, the user can be automatically logged in.

Format

DEFINE_AUTO_LOGIN or
DEFAL
HOST = name
USER = username
PASSWORD = password
FAMILY = family
ACCOUNT = account
PROJECT = project
STATUS = var of status

Parameters

HOST (H)

Name of the remote host associated with the login information.

USER (U)

The login user name.

PASSWORD (PW)

Password associated with the login user name.

FAMILY (F)

Family name associated with the login user name if different from the default family name used on the remote host. This parameter is only applicable if the remote host is NOS/VE or NOS.

The FAMILY parameter is not supported by FTP/VE for the R1.3 release.

ACCOUNT (A)

Account associated with the login. This parameter specifies the non-default account to which the resource usage on the remote host is charged.

PROJECT (P)

Project associated with the login. This parameter specifies the non-default project to which the resource usage on the remote host is charged.

If this parameter is omitted, the default established by the server's host administrator is used. The default is only valid if the remote host is a NOS/VE or a NOS host.

The PROJECT parameter is not supported by FTP/VE for the R1.3 release.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Remarks If one of the optional parameters has not been specified and the remote host requires that value for access validation, you are prompted for the value.

Examples The following subcommand defines the login information to be associated with a particular host name.

```
ftp/define_auto_login host=newhost user=name
ftp/
```

DELETE_DIRECTORY

Purpose Deletes one or more directories on the remote host.

NOTE

This command is a recent addition to the FTP protocol and may not be supported in some existing FTP Servers.

Format

DELETE_DIRECTORY or
DELD or
RMDIR or
RMD
NAME = list of foreign_name
STATUS = var of status

Parameters

NAME (N)

Name of one or more directories on the remote host to be deleted. The NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand deletes a directory on the remote host.

```
ftp/delete_directory name=/a/eg/test
ftp/
```

DELETE_FILE

Purpose Deletes one or more files on the remote host.

NOTE

To delete local files while using FTP, use the NOS/VE command $SYSTEM.DELETE_FILE or the NOS command PURGE.

Format

DELETE_FILE or
DELF or
DELETE or
DELE
NAME = list of foreign_name
STATUS = var of status

Parameters **NAME (N)**

Name of one or more remote host files to be deleted. The NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand deletes a file on the remote host.

```
ftp/delete_file name=test1
ftp/
```

DELETE_HOST_CONNECTION

Purpose Terminates an existing connection with a remote host without terminating CYBER FTP.

Format

DELETE_HOST_CONNECTION or
DELHC or
CLOSE
STATUS = var of status

Parameters *STATUS*

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand deletes a connection with the remote host.

```
ftp/delete_host_connection
ftp/
```

DISPLAY_COMMAND_INFORMATION

Purpose Displays information regarding a specific CYBER FTP subcommand.

Format **DISPLAY_COMMAND_INFORMATION** or **DISCI**

COMMAND = CYBER FTP command
STATUS = var of status

Parameters **COMMAND**

Name of a CYBER FTP subcommand for which information is required.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays information regarding the DELETE_FILE subcommand.

```
ftp/display_command_information command=delete_file
    name,n      : string = $required
    status      : var of status = $optional
ftp/
```

DISPLAY_DIRECTORY

Purpose

Displays a detailed list of files in the specified remote directories. The default is to display the current directory. If the name specified is a file name on the remote server host, current information on the file is displayed. The display format varies with different FTP Server implementations.

NOTE

To display the local directory, use the NOS/VE command DISPLAY_CATALOG or the NOS command CATLIST.

Format

DISPLAY_DIRECTORY or
DISD or
DIR or LIST
NAME = list of foreign_name
OUTPUT = local_name
STATUS = var of status

Parameters

NAME (N)

One or more remote host file names. The NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

OUTPUT (O)

Allows the specification of an output file other than the default.

The default is $OUTPUT for FTP/VE and OUTPUT for FTP/NOS.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays the current directory on the remote host.

```
ftp/display_directory
total 1119
-rw-rw-r--  1 eg      wheel      62 Mar 13 16:07 .login
-rw-rw-r--  1 eg      wheel    3752 Jul 9  13:45 .ftp_command_library
-rw-r--r--  1 eg      wheel   74227 Jul 22 14:36 .ftp_test
-rw-r--r--  1 eg      wheel      92 Jul 22 08:52 .prolog
-rw-rw-r--  1 eg      wheel     660 Jul 21 18:27 .test1
ftp/
```

DISPLAY_FILE_NAMES

Purpose Displays a list of file names in one or more remote directories. The default is to display the current directory. The display format varies with different FTP Server implementations.

Format

DISPLAY_FILE_NAMES or
DISFN or
LS or
NLST
NAME = list of foreign_name
OUTPUT = local_name
STATUS = var of status

Parameters *NAME (N)*

One or more remote host file names. The NAME parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

OUTPUT (O)

Allows the specification of an output file other than the default.

The default is $OUTPUT for FTP/VE and OUTPUT for FTP/NOS.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays a list of file names on the current directory in the remote host.

```
ftp/display_file_names
.login
.ftp_command_library
.ftp_test
.prolog
.test1
ftp/
```

DISPLAY_FTP_OPTIONS

Purpose Allows the user to obtain current information about the existing FTP connection.

Format

DISPLAY_FTP_OPTIONS or
DISFO or
STATUS
STATUS = var of status

Parameters *STATUS (S)*

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays an existing FTP connection.

```
ftp/display_ftp_options
Auto_Login_Mode          : ON
Debug_Mode               : OFF
Bell_Mode                : OFF
Expression_Evaluation    : OFF
Local_Byte_Size          : 8
Input_File               : $INPUT
Output_File              : $OUTPUT
Port_Command_Mode        : ON
Transfer_Format          : Non_Print
Transfer_Mode            : Stream
Transfer_Structure       : File
Transfer_Type            : ASCII
Verbose_Mode             : ON
Remote_Host_Name         : SUN
ftp/
```

DISPLAY_LOCAL_HELP

Purpose Provides information about each command. The output contains the CYBER FTP command, a brief description of the command and the aliases of the command.

NOTE

To request help from the NOS/VE system, use the NOS/VE command $SYSTEM.HELP or the NOS command HELP.

Format

DISPLAY_LOCAL_HELP or
DISLH or
HELP
STATUS = var of status

Parameters *STATUS*

Specifies a status variable to receive the termination condition of the subcommand.

Remarks

- For NOS/VE users, the command DISPLAY_COMMAND_INFORMATION (DISCI) displays command parameter information about any given command on the local host.
- FTP/NOS directly supports the DISPLAY_COMMAND_INFORMATION subcommand to display parameter information for a specific subcommand.
- The subcommand DISPLAY_COMMAND_LIST_ENTRY (DISCLE) displays the available CYBER FTP subcommands for both NOS/VE and NOS users.

Examples The following subcommand displays help information for all valid FTP subcommands.

```
ftp/display_local_help
 Command Name                  Command Aliases
 ------------                  ---------------

 APPEND_FILE                   APPE, APPEND, APPF
   Append to one or more remote files

 CHANGE_AUTO_LOGIN_MODE        CHAALM
   Turn Auto_login mode on or off

 CHANGE_BELL_MODE              BELL, CHABM
   Turn bell prompt on or off

 CHANGE_DEBUG_MODE             CHADM, DEBUG
   Turn debug messages on or off

 CHANGE_EXPRESSION_EVALUATION CHAEE
   Turn SCL evaluation on or off

 CHANGE_FILE_NAME              CHAFN, RENAME
   Rename one or more remote files

 CHANGE_INPUT_FILE             CHAIF
   Change default input file

 CHANGE_OUTPUT_FILE            CHAOF
   Change default output file

    .
    .
    .

Use DISPLAY_COMMAND_INFORMATION (DISCI) to get parameter info on a command.
Note: *'ed commands may not be available on other systems.
```

DISPLAY_REMOTE_HELP

Purpose Provides help information from the remote FTP Server.

Format

DISPLAY_REMOTE_HELP or
DISRH or
REMOTEHELP
COMMAND = ftp_command
STATUS = var of status

Parameters *COMMAND (C)*

Identifies a specific FTP command for which a user wants help.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays help information for the remote FTP Server.

```
ftp/display_remote_help
 214- The following commands are recognized (* =>'s unimplemented).
    USER      PORT      RETR      MSND*     ALLO      DELE      SITE*     XRMD
    PASS      PASV*     STOR      MSOM*     REST*     CWD       STAT*     XPWD
    ACCT*     TYPE      APPE      MSAM*     RNFR      XCWD      HELP      XCUP
    REIN*     STRU      MLFL*     MRSQ*     RNTO      LIST      NOOP
    QUIT      MODE      MAIL*     MRCP*     ABOR*     NLST      XMKD
 214  Direct comments to ftp-bugs@Svlvax.
ftp/
```

DISPLAY_WORKING_DIRECTORY

Purpose Displays the name of the current working directory on the remote host.

NOTE

This command is a recent addition to the FTP protocol and may not be supported in some existing FTP Servers.

Format

DISPLAY_WORKING_DIRECTORY or
DISWD or
PWD
STATUS = var of status

Parameters

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand displays the current working directory on the remote host.

```
ftp/display_working_directory
"/a/eg" is current directory
ftp/
```

ELEVATE_WORKING_DIRECTORY

Purpose Allows the user to move up one level in the directory hierarchy on the remote host.

NOTE

This command is a recent addition to the FTP protocol and may not be supported in some existing FTP Servers.

Format

ELEVATE_WORKING_DIRECTORY or
ELEWD or
CDUP
STATUS = var of status

Parameters *STATUS*

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand allows the user to move up one level in the directory hierarchy on the remote host.

```
ftp/elevate_working_directory
ftp/
```

GET_FILE

Purpose Retrieves one or more files from the remote host.

NOTE

To get a file from NOS to NOS/VE on a dual-state local host, use the NOS/VE command $SYSTEM.GET_FILE.

Format **FTP/VE Format:**

GET_FILE or
GETF or
GET or
RECV or
RETR
REMOTE_FILE = list of foreign_name
LOCAL_FILE = list of local_name
STATUS = var of status

Format **FTP/NOS Format:**

GET_FILE or
GETF or
GET or
RECV or
RETR
REMOTE_FILE = foreign_name
LOCAL_FILE = local_name or
LOCAL_FILE = ' local name CS=cset trunc '
STATUS = var of status

Parameters **REMOTE_FILE (RF)**

Name of one or more remote host files. The REMOTE_FILE parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

LOCAL_FILE (LF)

Name of one or more local host files. The LOCAL_FILE parameter is implemented as a local_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

CS (FTP/NOS only)

Specifies the codeset, data character size and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command is ASCII (A).

The following NOS FCOPY command compatible values are supported:

Value	Description
ASCII (A)	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 (8)	ASCII 8/12 file with TELNET end_of_line terminators.
ASCII88 (88)	ASCII 8/8 file with TELNET end_of_line terminators.
DIS (D)	Display Coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

TRUNC (FTP/NOS only)

Specifies the deletion of pad bits from the file being received by FTP/NOS. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command is IMAGE (I).

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Remarks

- If the LOCAL_FILE parameter is not specified, the GET_FILE subcommand attempts to use the value(s) supplied in the REMOTE_FILE parameter as the local host file name(s). If this is not possible, the GET_FILE subcommand aborts with an error. If the LOCAL_FILE parameter is specified, the list must contain the same number of elements as the REMOTE_FILE parameter.

- The REMOTE_FILE parameter can be prefixed with a server-host-dependent special character that causes the parameter string to be executed as local commands at the remote FTP Server host. The output resulting from command execution on the remote host is stored in the file indicated by the LOCAL_FILE parameter. For FTP/VE, the EXPRESSION_EVALUATION parameter must be set to ON.

- IF an FTP/NOS control record is not present, four pad bits are added to a NOS file that is an odd number of CYBER CM words (not an integral number of 8 bit bytes) during an image mode transmission to another host. These bits must be stripped from the file upon its return to a NOS system.

- If the remote host is NOS/VE and the REMOTE_FILE string begins with the ! character, the rest of the string is interpreted as a command line that may include multiple NOS/VE commands. The FTP/NOS Server does not currently support the execution of NOS commands from a remote host client.

Examples The following subcommand retrieves a file from the remote host.

```
ftp/get_file remote_file=prolog local_file=test1
ftp/
```

The example on the following page shows how NOS/VE commands can be executed from a VAX workstation using FTP.

After logging into the VAX workstation, you can connect to the CYBER host using FTP. The GET command allows you to display and change the file attributes of a NOS/VE file.

```
Login to VAX....

%ftp cyber
Connected to s0.
220 LOCALHOST Server FTP (Version 1.0.0) ready.
Name (v06:usn): username

331 User name received, need password.
Password:password

230 User username logged in.
ftp>ls

200 PORT command successful.
150 Opening data connection for Name List (192.12.251.32,1141) ( 837 bytes).
TESTFILE
TESTFILE1
TESTFILE2
226 Transfer complete.
196 bytes received in 0.13 seconds (1.5 Kbytes/s)
ftp> get  "!display_file_attributes testfile"-

200 PORT command successful.
150 Opening data connection for !display_file_attribute testfile (192.12.251.32,
1142) ( 519 bytes).
1display_file_attributes                          NOS/VE R1  16503  09P9
                    1988-02-18  10.43.30,70     PAGE 1
 FILE :mercury.username.testfile
0File_Contents              : legible
 File_Processor             : unknown
 File_Structure             : data
 Global_Access_Mode         : (read, shorten, append, modify, execute)
 Permanent                  : yes
 Size                       : 69
226 Transfer complete.
remote: !display_file_attribute testfile
423 bytes received in 0.1 seconds (4 Kbytes/s)
ftp> get "!change_file_attributes testfile fc=list fs=data;  display_file_
_attributes testfile"-

200 PORT command successful.
150 Opening data connection for !change_file_attribute testfile fc=list fs=data;
 display_file_attribute testfile (192.12.251.32,1143) ( 516 bytes).
1display_file_attributes                          NOS/VE R1  16503  09P9
                    1988-02-18  10.44.41,89     PAGE 1
 FILE :mercury.username.testfile
0File_Contents              : list
 File_Processor             : unknown
 File_Structure             : data
 Global_Access_Mode         : (read, shorten, append, modify, execute)
 Permanent                  : yes
 Size                       : 69
226 Transfer complete.
remote: !change_file_attribute testfile fc=list fs=data; display_file_attribute
testfile
420 bytes received in 0.15 seconds (2.8 Kbytes/s)
ftp> quit

221 Goodbye.
```

LOGIN_USER

Purpose Sends login information to the remote host after a connection has been established.

NOTE

You must supply all parameters that the remote host requires for login. The LOGIN_USER subcommand aborts with an error if the remote host requires a value that was not supplied by the user.

Format

LOGIN_USER or
LOGU or
USER
USER = username
PASSWORD = password
FAMILY = family
ACCOUNT = account
PROJECT = project
STATUS = var of status

Parameters

USER (U)

Login user name.

PASSWORD (PW)

Password associated with the login user name.

FAMILY (F)

Family name associated with the login user name if it is other than the default family for the remote host. The FAMILY parameter value is sent to the remote host in the PASS FTP protocol element (FTP command) by prefixing it with a comma acting as a delimiter and suffixing the result string to the PASSWORD parameter value. The FAMILY parameter is only applicable if the remote host is a CYBER host.

The FAMILY parameter is not supported by FTP/VE for the R1.3 release.

ACCOUNT (A)

Account associated with the login. This parameter specifies the non-default account to which the resource usage on the remote is charged.

PROJECT (P)

Project associated with the login. This parameter specifies the non-default to which the resource usage on the remote host is charged.

If this parameter is omitted, the default established by the server's host administrator is used. This parameter is only valid if the remote host is a CYBER host.

The PROJECT parameter is not supported by FTP/VE for the R1.3 release.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples

The following subcommand logs the CYBER FTP user into the remote host after a connection has been established.

```
ftp/login_user user=usrname
ftp/
```

QUIT

Purpose Terminates execution of CYBER FTP.

Format

QUIT or
QUI or
BYE or Q
ABORT = boolean
STATUS = var of status

Parameters *ABORT*

Specifies whether to abort the current connection before terminating the execution of CYBER FTP.

Setting the ABORT parameter to ON, TRUE, or YES causes CYBER FTP to abort the current connection before terminating; setting the ABORT parameter to OFF, FALSE, or NO causes CYBER FTP to close any existing connection before terminating.

If this parameter is omitted, by default CYBER FTP tries to close any existing connection before terminating.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand terminates the execution of CYBER FTP.

```
ftp/quit
/
```

REPLACE_FILE

Purpose Sends data from a file on the local host to a file on the remote host. If the file does not exist on the remote host, it is created.

NOTE

To replace a file from NOS/VE to NOS on a dual-state local host, use the NOS/VE command $SYSTEM.REPLACE_FILE.

Format **FTP/VE Format:**

REPLACE_FILE or
REPF or
PUT or
SEND or
STOR
LOCAL_FILE = list of local_name
REMOTE_FILE = list of foreign_name
STATUS = var of status

Format **FTP/NOS Format:**

REPLACE_FILE or
REPF or
PUT or
SEND or
STOR
LOCAL_FILE = local_name *or*
LOCAL_FILE = ' local name *CS=cset FSC=n readeoi* **'**
REMOTE_FILE = foreign_name
STATUS = var of status

Parameters **LOCAL_FILE**

Name of one or more files on the local host. The LOCAL_FILE parameter is implemented as a LOCAL_NAME parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

REMOTE_FILE (RF)

Name of one or more files on the remote host. The REMOTE_FILE parameter is implemented as a foreign_name parameter whose actual type depends on the value of the EXPRESSION_EVALUATION parameter. For more information, refer to File Name Processing.

CS (FTP/NOS only)

Specifies the codeset, data character size and format of the file on the NOS host. Files transmitted by FTP/NOS are mapped to network ASCII from the specified representation. This parameter is ignored unless the transfer type, as selected or defaulted by the CHANGE_TRANSFER_TYPE command is IMAGE (I).

The following NOS FCOPY command compatible values are supported:

Value	Description
ASCII (A)	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 (8)	ASCII 8/12 file with TELNET end_of_line terminators.
ASCII88 (88)	ASCII 8/8 file with TELNET end_of_line terminators.
DIS (D)	Display Coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

FSC (FTP/NOS only)

Specifies the number of files to be skiped before starting data transfer to the remote host. If this parameter is specified, n NOS EOF marks are skipped before data transfer begins. Data transfer begins at file n+1 of a multi-file set. tion (BOI) of the file.

If this parameter is omitted, the default is zero. Data transfer starts from the BOI of the file.

READEOI (FTP/NOS only)

Specifies read to EOI. If this parameter is specified, all files of a multi-file set from the current file position to the EOI of the file are transferred with NOS EOF marks ignored.

The default is to transfer only one file.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Remarks If the REMOTE_FILE parameter is not specified, the REPLACE_FILE subcommand uses the local file name(s) for the remote file name(s). If the REMOTE_FILE parameter is specified, the list must contain the same number of elements as the LOCAL_FILE parameter.

Examples The following subcommand sends a local host file to a remote host file.

```
ftp/replace_file local_file=test remote_file=test
ftp/
```

SEND_FTP_COMMAND

Purpose Allows the sophisticated CYBER FTP user to directly send an FTP command to a remote host. It receives a single reply from the remote server. This command is used for debugging purposes by users who are familiar with the FTP protocol. This command does not work for FTP commands that initiate a data transfer (LIST, NLST, STOR, RETR, and APPE).

Format

SEND_FTP_COMMAND or
SENFC or
QUOTE
COMMAND = ftp_command
PARAMETERS = command_parameters
STATUS = var of status

Parameters

COMMAND

Sends 3- or 4-character FTP commands. This parameter follows the same rules as the foreign_name file parameter.

PARAMETERS

Sends additional parameters required for the given FTP command. This parameter is parsed similarly to foreign_name file parameter, following the same rules.

STATUS

Specifies a status variable to receive the termination condition of the subcommand.

Examples The following subcommand sends an FTP command to the remote host.

```
ftp/send_ftp_command command=help
ftp/
```

CYBER FTP Support of FTP Commands

This section lists the RFC 959 FTP commands that are supported by the CYBER FTP Client and Server. TELNET protocol support on the control connection and deviations from the recommended minimum implementation are also discussed.

TELNET Support on the Control Connection

CYBER FTP directly supports the TELNET strings on the control connection. Support of TELNET option negotiation commands is limited to responding to a WILL command with a DON'T and a DO command with a WON'T. All other TELNET commands are not supported.

RFC 959 FTP Minimum Implementation Statement

The FTP Protocol Specification document is RFC 959. Section 5.1 is the declarative specification for a minimum implementation. It states the minimum implementation required for all servers.

Duplicate of RFC 959, Section 5.1

```
Minimum Implementation

In order to make FTP workable without needless error messages,
the following minimum implementation is required for servers:

    TYPE - ASCII Non-print
    MODE - Stream
    STRUCTURE - File, Record
    COMMANDS - USER, QUIT, PORT,
               TYPE, MODE, STRU for the default values
               RETR, STOR,
               NOOP

The default values for transfer parameters are:

           TYPE - ASCII Non-print
           MODE - Stream
           STRU - File

All hosts must accept the above as the standard defaults.
```

FTP/VE Support of RFC 959 Commands

Table 3-1 lists all of the RFC 959 FTP commands and defines the FTP commands supported by the FTP/VE Client/Server.

Command descriptions that have no corresponding comment are fully supported by both the CYBER FTP Client and Server.

Table 3-1. FTP/VE Implementation of RFC 959 Commands

Command	Description	Supported by Client	Supported by Server	Comments
ACCT	Account	YES	NO	
ALLO	Allocate	NO	NO	
APPE	Append	YES	YES	
CDUP	Change To Parent Working Directory	YES	YES	
CWD	Change Working Directory	YES	YES	
DELE	Delete	YES	YES	
HELP	Help	YES	YES	
LIST	List	YES	YES	See Note 1.
MKD	Make Directory	YES	YES	
MODE	Transfer Mode	NO	YES	STREAM mode only.
NLST	Name List	YES	YES	See Note 1.
NOOP	No Operation	NO	YES	
PASS	Password	YES	YES	See Note 2.
PASV	Passive	NO	NO	
PORT	Data Port	YES	YES	Sent by Client only if PORT_COMMAND_MODE is ON.
PWD	Print Working Directory	YES	YES	
QUIT	Logout	YES	YES	

(Continued)

Table 3-1. FTP/VE Implementation of RFC 959 Commands *(Continued)*

Command	Description	Supported by Client	Supported by Server	Comments
REIN	Reinitialize	NO	NO	
REST	Restart	NO	NO	
RETR	Retrieve	YES	YES	See Note 3.
RMD	Remove Directory	YES	YES	
RNFR	Rename From	YES	YES	RNFR must be followed by the RNTO.
RNTO	Rename To	YES	YES	
SITE	Site Parameters	NO	NO	
SMNT	Structure Mount	NO	NO	
STAT	Status	NO	NO	
STOR	Store	YES	YES	
STOU	Store Unique	NO	NO	
STRU	File Structure	NO	YES	FILE Structure only.
SYST	System	NO	NO	
TYPE	Representation Type	YES	YES	ASCII NONPRINT and IMAGE only.
USER	User Name	YES	YES	
XCUP	Change to Parent Working Directory	NO	YES	Non-standard command.
XCWD	Change Working Directory	NO	YES	Non-standard command.
XMKD	Make Directory	NO	YES	Non-standard command.

(Continued)

Table 3-1. FTP/VE Implementation of RFC 959 Commands *(Continued)*

Command	Description	Supported by Client	Supported by Server	Comments
XPWD	Print Working Directory	NO	YES	Non-standard command.
XRMD	Remove Directory	NO	YES	Non-standard command.

NOTE

Notes on FTP/VE Server support of FTP commands:

Note 1: If the transfer type is IMAGE, ASCII text strings terminated by the <LF> character are sent to the client. This is implemented in order to enhance the interoperability with the UNIX FTP Clients.

Note 2: FTP/VE Server does not require ACCOUNT/PROJECT for user access validation. If the remote user is required to provide the ACCOUNT/PROJECT for logging in to the FTP/VE Server host and no default ACCOUNT/PROJECT exists in the validation, the required information can be included in the password parameter of the PASS command.

The string ',account=aaa' where aaa is a valid account number, can be appended to the password to specify a non-default account number for the user login.

The string ',project=ppp' where ppp is a valid project number, can be appended to the password to specify a non-default project number for the user login.

A non-default family for the user login can also be specified by appending the string ',family_name=fff' where fff is a valid NOS/VE family name, to the password string.

Note 3: If the RETR command specifies a pathname that begins with the '!' character, the FTP/VE Server executes the rest of the pathname text as a NOS/VE command line which can include multiple SCL commands. The text that follows the '!' character must conform to SCL command syntax. The resulting output from the SCL command(s) is returned to the FTP Client over the data connection when the CHANGE_EXPRESSION_EVALUATION mode is ON.

FTP/NOS Support of RFC 959 Commands

Table 3-2 lists all of the RFC 959 FTP commands and defines the FTP commands supported by the FTP/NOS Client/Server.

Command descriptions that have no corresponding comment are fully supported by both the CYBER FTP Client and Server.

Table 3-2. FTP/NOS Implementation of RFC 959 Commands

Command	Description	Supported by Client	Supported by Server	Comments
ABOR	Abort	NO	NO	
ACCT	Account	YES	YES	
ALLO	Allocate	NO	NO	
APPE	Append	YES	YES	
CDUP	Change To Parent Working Directory	YES	NO	
CWD	Change Working Directory	YES	NO	
DELE	Delete	YES	YES	
HELP	Help	YES	YES	
LIST	List	YES	YES	
MKD	Make Directory	YES	NO	
MODE	Transfer Mode	NO	YES	STREAM mode only.
NLST	Name List	YES	YES	
NOOP	No Operation	NO	YES	
PASS	Password	YES	YES	
PASV	Passive	NO	YES	
PORT	Data Port	YES	YES	Sent by client only if PORT_COMMAND_MODE is ON.

(Continued)

Table 3-2. FTP/NOS Implementation of RFC 959 Commands *(Continued)*

Command	Description	Supported by Client	Supported by Server	Comments
PWD	Print Working Directory	YES	NO	
QUIT	Logout	YES	YES	
REIN	Reinitialize	NO	NO	
REST	Restart	NO	NO	
RETR	Retrieve	YES	YES	
RMD	Remove Directory	YES	NO	
RNFR	Rename From	YES	YES	
RNTO	Rename To	YES	YES	
SITE	Site Parameters	NO	NO	
SMNT	Structure Mount	NO	NO	
STAT	Status	NO	NO	
STOR	Store	YES	YES	
STOU	Store Unique	NO	NO	
STRU	File Structure	NO	YES	FILE STRUCTURE only.
SYST	System	NO	NO	
TYPE	Representation Type	YES	YES	ASCII NONPRINT and IMAGE only.
USER	User Name	YES	YES	
XCUP	Change to Parent Working Directory	NO	NO	Non-standard command.
XCWD	Change Working Directory	NO	NO	Non-standard command.

(Continued)

Table 3-2. FTP/NOS Implementation of RFC 959 Commands *(Continued)*

Command	Description	Supported by Client	Supported by Server	Comments
XMKD	Make Directory	NO	NO	Non-standard command.
XPWD	Print Working Directory	NO	NO	Non-standard command.
XRMD	Remove Directory	NO	NO	Non-standard command.

FTP/NOS Server Interface

The FTP/NOS Server provides remote TCP/IP users and applications access to the NOS file system.

FTP/NOS Syntax Requirements for Protocol Elements

The FTP protocol elements are TELNET ASCII character strings terminated by the TELNET end-of-line code and are partitioned as those specifying access-control identifiers, data transfer parameters, or FTP service requests. The syntax common to all FTP protocol elements is as follows:

- FTP protocol elements begin with a command code consisting of one to four alphabetic ASCII characters.
- Upper and lowercase alphabetic characters are recognized as equal in the command code.
- Parameters following the command code are separated from the command code by one or more ASCII spaces.
- Parameters consist of a variable length ASCII character string.
- Upper and lowercase alphabetic characters are recognized as equal in the parameters.
- The FTP protocol element is terminated with the TELNET end-of-line code consisting of the ASCII character sequence of Carriage Return and Line Feed [CRLF].
- No action is taken by the FTP Server until the TELNET end-of-line code [CRLF] is received.

Access Control Protocol Element Support

Access control protocol elements allow the user of a remote host's FTP Client to provide validation information to the FTP/NOS Server. The Server uses validation information provided to determine the remote user's access privileges to the system and to the files contained in the system. The protocol elements described are as follows:

- User Protocol Element
- Password Protocol Element
- Account Protocol Element
- Logout Protocol Element

User Protocol Element

Purpose Identifies the user to the Server FTP for access to the NOS file system.

Format **USER username,***password,family*

Parameters

username

Specifies a NOS user name of one to seven alphanumeric characters.

password

Specifies the NOS batch password for the user name specified. It can be from one to seven alphanumeric characters and is preceded by a comma (,).

If this parameter is omitted, two commas are necessary.

family

Specifies a NOS family name of zero to seven alphanumeric characters and is preceded by one comma (,).

If this parameter is omitted, the default family is used.

Password Protocol Element

Purpose Identifies the user password to the Server FTP for access to the NOS file system.

Format **PASS password,***family*

Parameters **password**

Specifies the NOS batch password for the user name previously specified. It can be from one to seven alphanumeric characters.

family

Specifies a NOS family name of zero to seven alphanumeric characters preceded by a comma (,) and takes precedence over any FAMILY selection on the USER command.

Remarks The PASS protocol element can be processed only immediately after a USER command.

Account Protocol Element

Purpose Identifies the account number associated with the previously supplied FAMILY/USER combination.

Format **ACCT chargenum,projectnum**

Parameters

chargenum

Specifies a NOS charge number of 1 to 10 alphanumeric characters in upper and/or lowercase. The comma following the charge number is required.

projectnum

Specifies a NOS project number of 1 to 20 alphanumeric characters in upper and/or lowercase.

Logout Protocol Element

Purpose Causes FTP to complete any file transfer in progress, terminate the user connection, and close the control connection.

Format **QUIT**

Remarks This command causes FTP to complete any file transfer in progress and close the user's data and control connections.

Transfer Parameter Protocol Element Support

The transfer parameter protocol elements allow the user of a remote host's FTP Client to change default parameter values from the last value specified by a transfer parameter command or, if no parameters are specified, to the default value listed in the FTP standard. Transfer parameter commands are order-independent except that they must all precede the FTP service commands. The protocol elements described are as follows:

- Data Port Protocol Element
- Passive Protocol Element
- Representation Type Protocol Element
- File Structure Protocol Element
- Transfer Mode Protocol Element

Data Port Protocol Element

Purpose Specifies the data port to be used for the data connection.

Format **PORT host_port**

Parameters **host_port**

Specifies 6 groups of ASCII decimal digits, each in the range of 0 to 255, delimited by commas (,). Each group of digits represents the value of subsequent 8-bit fields in a 32-bit internet host address and a 16-bit TCP port address.

Passive Protocol Element

Purpose Causes the Server FTP to listen on a data port (not the default data port) and wait for a connection, rather than initiate a connection upon receipt of a transfer service command.

Format **PASV**

Representation Type Protocol Element

Purpose Specifies the data type for the file transfer.

Format **TYPE type_code_n**

Parameters **type_code_n**

Specifies the data type for the file transfer using one of the following keywords:

- **type_code_1** - Specifies the data type for the file transfer parameter which specifies that the file contains no vertical format information, the file contains TELNET vertical format control characters, or the file contains FORTRAN vertical format control characters. The type_code_1 parameter is recognized in upper and/or lowercase in one of the following formats:

Value	Description
A	Specifies an ASCII file transfer.
N	Specifies that the file contains no vertical format control information.

  The default is N.

- **type_code_2** - Specifies the data type for transfer as IMAGE with data sent as contiguous bits. The type_code_2 parameter is recognized in upper and/or lowercase using the following format:

  I Specifies an IMAGE file transfer.

File Structure Protocol Element

Purpose Specifies the file structure for the file transfer.

Format **STRU s_code**

Parameters **s_code**

Specifies the file structure as a single ASCII character in upper or lowercase. Valid value is:

F File structure (no record structure)

Transfer Mode Protocol Element

Purpose Specifies the mode for the file transfer.

Format **MODE m_code**

Parameters **m_code**

Specifies the transfer mode as a single ASCII character in upper or lowercase. Valid value is:

S Stream

FTP Service Commands

The FTP service commands define the file transfer or the file system function requested by the user. The only order dependency required for FTP service commands is that RENAME FROM must be followed by RENAME TO. The protocol elements described are as follows:

- Append Protocol Element
- Retrieve Protocol Element
- Store Protocol Element
- Rename From Protocol Element
- Rename To Protocol Element
- Delete Protocol Element
- LIST Protocol Element
- NLST Protocol Element
- HELP Protocol Element
- NOOP Protocol Element

Append Protocol Element

Purpose Causes the Server FTP to accept the data from the data connection, add it to the end of an existing file with the same name, or create a new file to contain the data.

Format **APPE pf_parameters**,*data_handling*

Parameters **pf_parameters**

Any parameter combination valid for a single *pfn* for the NOS ATTACH and GET or DEFINE and SAVE commands, except for the parameters *lfn*, NA, RT, and WB. The NOS ATTACH, GET, DEFINE, and SAVE commands are defined in the NOS Reference Manual, Volume 3.

The addition of indirect access (IA) or direct access (DA) may be necessary to select the file type if the file must be created.

data_handling

The data format and handling requirements of the file are discussed in the section File Data and Position Handling.

Remarks

- Only the PF_PARAMETERS appropriate to the particular operation being performed are used, the remainder are ignored.
- If the file does not exist, it is created. If no type was specified, the default is direct access unless the user is not validated for direct access files, in which case an indirect access file is created. Existing indirect access files are replaced after the data has been added at the end of the file. Existing direct access files are attached in write mode and the data is added at the end of the file.
- Refer to the File Support section for a detailed discussion on the control record suffixed to files created by FTP/NOS during image mode transfers, and how it is used to determine where to add data at the end of a file.

Retrieve Protocol Element

Purpose Causes FTP to transfer a copy of the named file to a server or user FTP at the other end of the data connection.

Format **RETR pf_parameters**,*data_handling*

Parameters **pf_parameters**

Any parameter combination valid for a single *pfn* for the NOS ATTACH and GET commands, except parameters *lfn*, NA, RT and WB. The NOS ATTACH and GET commands are defined in the NOS Reference Manual, Volume 3.

data_handling

The data format and handling requirements of the file are discussed in the section File Data and Position Handling.

Remarks

- Only the PF_PARAMETERS appropriate to the particular operation being performed are used; the rest are ignored.
- Refer to the File Support section for a detailed discussion on the control record suffixed to files created by FTP/NOS during image mode transfers, and how it is used to determine the end of data when transmitting a file.

Store Protocol Element

Purpose Causes FTP to accept data from the data connection and to store the data on a NOS file.

Format **STOR pf_parameters**,*data_handling*

Parameters **pf_parameters**

Any parameter combination valid for a single *pfn* for the NOS DEFINE and SAVE commands, except parameters *lfn*, NA, and WB. The NOS DEFINE and SAVE commands are defined in the NOS Reference Manual, Volume 3.

The addition of IA and DA may be necessary to select the file type if the file must be created.

data_handling

The data format and handling requirements of the file are discussed in the section File Data and Position Handling.

Remarks

- Only the PF_PARAMETERS appropriate to the particular operation being performed are used; the rest are ignored.
- If the file does not exist, it is created. If no type was specified, the default is direct access unless the user is not validated for direct access files, in which case an indirect access file is created. Existing indirect access files are replaced. Existing direct access files are attached in write mode and overwritten.
- Refer to the File Support section for a detailed discussion on the control record suffixed to files created by FTP/NOS during image mode transfers.

Rename From Protocol Element

Purpose Specifies the file to be renamed by the Server FTP upon receipt of the RENAME TO command.

Format **RNFR change_parameters**

Parameters **change_parameters**

The *pfn* parameter on the NOS CHANGE command. The CHANGE command is defined in the NOS Reference Manual, Volume 3.

Rename To Protocol Element

Purpose Specifies the new file name for the file named in the immediately preceding RENAME FROM command.

Format **RNTO change_parameters**

Parameters **change_parameters**

Any parameter combination valid for the NOS CHANGE command, except parameters *lfn*, NA and WB. The NOS CHANGE command is defined in the NOS Reference Manual, Volume 3.

Delete Protocol Element

Purpose Causes the Server FTP to delete the named file at the Server site. User FTP provides a caution message to NOS interactive terminal users prior to file deletion.

Format **DELE purge_parameters**

Parameters **purge_parameters**

Any parameter combination valid for the NOS PURGE command, except parameters NA and WB. The NOS PURGE command is defined in the NOS Reference Manual, Volume 3.

LIST Protocol Element

Purpose Causes the Server FTP to send a list of files to the user FTP.

Format **LIST catlist_parameters**

Parameters **catlist_parameters**

Any parameter combination valid for the NOS CATLIST command, except parameters NA, WB, and L. The NOS CATLIST command is defined in the NOS Reference Manual, Volume 3.

The LO parameter is defaulted to F.

NLST Protocol Element

Purpose Causes the Server FTP to send a list of files to the user FTP.

Format **NLST catlist_parameters**

Parameters **catlist_parameters**

Any parameter combination valid for the NOS CATLIST command, except parameters NA, WB, and L. The NOS CATLIST command is defined in the NOS Reference Manual, Volume 3.

The LO parameter is defaulted to 0 (zero).

HELP Protocol Element

Purpose Causes help text to be returned to the user.

Format **HELP** *key_word*

Parameters *KEY_WORD*

Specifies the type of help needed as a character string in upper and/or lowercase ASCII characters.

NOOP Protocol Element

Purpose Specifies that no action be taken by the Server FTP, except to send an OK reply.

Format **NOOP**

File Data and Position Handling

The DATA_HANDLING parameter is used to define to FTP/NOS the attributes of and handling requirements for the file to be manipulated.

The following parameters are available:

Parameter	Description
CS=cs	Character set
FSC=n	Multi-file file positioning
READEOI	Multi-file file processing
TRUNC	Pad bit processing

CS Parameter

The character set parameter (CS) allows definition of the codeset, data character size, and format of the file on the NOS host. FTP/NOS maps the files it transmits or receives between network ASCII and the selected representation.

The following NOS FCOPY command compatible values are supported:

Value	Description
ASCII (A)	ASCII 6/12 file with zero-byte-terminated records.
ASCII8 (8)	ASCII 8/12 file with TELNET end_of_line terminators.
ASCII88 (88)	ASCII 8/8 file with TELNET end_of_line terminators.
DIS (D)	Display Coded file with zero-byte-terminated records.

If this parameter is omitted, the default is ASCII (A).

FSC and READEOI Parameters

The FSC and READEOI parameters allow FTP/NOS to manipulate NOS multi-file files for the user.

NOS defines three types of file marks:

- EOR (end-of-record)
- EOF (end-of-file)
- EOI (end-of-information)

FTP only defines two:

- EOR (end-of-record)
- EOF (end-of-file)

The NOS and FTP EOR marks correspond, while the NOS EOI corresponds to the FTP EOF. There is no FTP equivalent to the NOS EOF.

By default, FTP/NOS treats NOS EOF as an FTP EOF. Since FTP stops transferring a file when it receives an EOF mark, by default only one NOS file in a multi-file file can be transferred. The following parameters, which are only supported for the RETR command allow multi-file files to be directly manipulated:

FSC=n	File skip count. If this parameter is specified, n NOS EOF marks are skipped before data transfer begins. Data transfer begins at file n+1 of a multi-file set. The default is zero. Data transfer starts from the BOI of the file.
READEOI	Read EOI. If specified, all files of a multi-file set from the current file position to the EOI of the file are transferred with NOS EOF marks ignored.

TRUNC Parameter

The TRUNC parameter is used to delete pad bits from a file being received by FTP/NOS. It is applicable only to image mode transfers to FTP/NOS of image files that originated on a NOS system and did not have an FTP/NOS control record. The four bits of pad that are added to a NOS file that is an odd number of CYBER CM words (not an integral multiple of 8-bit bytes) during an image mode transmission to another host are stripped from the file returned to the NOS host when this parameter is specified.

File Support

The following section describes file support for FTP/VE and FTP/NOS.

FTP/VE Support of NOS/VE Files

FTP/VE Client and FTP/VE Server support identical FTP file transfer parameters and NOS/VE file system interface.

FTP supports only ASCII (a network ASCII text file) and IMAGE (a binary file) types of file transfers; otherwise, only the default values defined for the remaining FTP file transfer parameters are supported as follows:

- STREAM Transmission Mode. Data is transmitted as a stream of bytes.
- NONPRINT Format Control. An ASCII file contains no vertical format control information. If FTP/VE receives a file with its file attributes indicating a print file (file_contents is 'LIST'), the vertical spacing character ' ' is prefixed at the beginning of each text line.
- FILE Structure. The file contains a continuous sequence of data bytes with no internal structure.

NOS/VE file attributes are not preserved when the file is transferred to a remote host. The user is responsible for setting the correct file attributes for a file when it is retrieved from a remote host; otherwise, the existing file attributes are used for an old file, and the NOS/VE default file attributes are used for a new file.

Values for file attributes can be set with the NOS/VE commands SET_FILE_ATTRIBUTES for a new file, or CHANGE_FILE_ATTRIBUTES for an existing file. If the file attributes are important in the proper processing of the file by NOS/VE, then they must be verified and changed by the user if necessary. For more information, refer to NOS/VE System Usage manual.

A local NOS/VE user can use SCL to change file attributes on files retrieved from remote hosts. A remote user may change file attributes on NOS/VE server host files by invoking a remote host FTP command that would cause a RETR command to be sent with the pathname beginning with the '!' character and which is followed by the NOS/VE CHANGE_FILE_ATTRIBUTES command. The FTP/VE Server would execute the text following the '!' as a NOS/VE command line.

ASCII Files

FTP/VE accesses ASCII text files as sequential record access files. When transferring such ASCII text files to a remote host, FTP/VE would indicate the end of record with the TELNET end_of_line code <CRLF> and the end_of_file by closing the data connection. When retrieving an ASCII text file, FTP/VE performs the inverse translation.

For example, the TELNET end_of_line code would denote an end of record and cause a record to be written to the file. Closing of the data connection would signal the end_of_file.

IMAGE Files

FTP/VE accesses IMAGE files as segment access files; it performs no translation. It reads or writes file data to the file until the end of file is reached, which is indicated by the closing of the data connection.

FTP/NOS Support of NOS Files

The FTP/NOS Client can perform only data transfer on files local to the job in which it was invoked. Files sent by the client from a NOS host must be made local to the job before FTP/NOS is invoked. They are required to be of NOS type LO or PM (local or direct access permanent file, respectively) as defined by the NOS CCL FILE function. If a file to be sent by FTP/NOS is not one of the above types, FTP/NOS issues a diagnostic message and aborts.

Files retrieved from a remote host are written to the local file name specified on the FTP transfer subcommand. If the named local file does exist but is not of the correct type, FTP/NOS issues a warning message to the user, returns the file and continues as if the file did not exist. If the named local file did not exist before the client was invoked, it is created.

FTP/NOS always rewinds the specified local file before and after each transfer.

FTP/NOS supports the following representation formats:

- DISPLAY and ASCII 6/12 files with zero-byte-terminated records
- ASCII 8/12 and ASCII 8/8 files with TELNET end_of_line terminators
- IMAGE files

Only the default values defined for the remaining FTP file transfer parameters are supported:

- STREAM Transmission Mode. Data is transmitted as a stream of bytes.
- NONPRINT Format Control. An ASCII file contains no vertical format control information. If FTP/NOS receives a file with its file attributes indicating a print file the vertical spacing character ' ' is prefixed to the beginning of each text line.
- FILE Structure. The file contains a continuous sequence of data bytes with no internal structure.

ASCII Files

FTP/NOS accesses ASCII text files as sequential files. When transferring ASCII text files to a remote host, FTP/NOS indicates the end of record (line) with the TELNET end_of_line code <CRLF> and the end_of_file by closing the data connection. When retrieving an ASCII text file, the inverse translation takes place. For example, the TELNET end_of_line code would denote an end of record and cause a zero-byte terminated record to be written to a DISPLAY file or an ASCII 6/12 file. Closing of the data connection would signal EOF.

ASCII 8/12 and 8/8 files are transmitted/received as a stream of bytes with no mapping performed to/from network ASCII. End of data on transmission is defined as the line terminator in the last CM word of the file. The bytes following are not sent to the network. If no line terminator is present in the last CM word of the file, the data bytes in the last word are output completely, which means that the four bits at the end of an odd CM word are discarded to make the transfer an integral number of bytes.

IMAGE Files

FTP/NOS accesses IMAGE files as sequential files. No translation is performed by FTP/NOS. File data is read from or written to the file until the end of file is reached, which is indicated by the closing of the data connection. The actual end of data is determined differently depending upon the presence or absence of an FTP/NOS control record on the file.

FTP/NOS Control Record

FTP/NOS suffixes a control record to every image (binary) file that it creates. It also updates the control record if one already exists when it appends to the file. The control record is required because there is no standard NOS record format that allows a file consisting of an arbitrary number of eight-bit bytes to be stored on a NOS host and recovered.

The control record immediately follows the EOF mark that delimits the end of user data in the file. It is used by FTP/NOS to determine the position of the last byte of actual user data in the last word of the file.

FTP/NOS Control Record Format

The following shows the FTP/NOS control record format.

FTP/NOS Control Record Use

All files created in image mode by FTP/NOS have a control record suffixed. When a file is transmited in image mode by FTP/NOS, it checks for the presence of a control record. If one is found, the end of user data within the file is determined from the record.

When FTP/NOS appends received data to a file, it checks for the presence of a control record. If one is found, the appended data starts in the next free byte and the control record is updated at the end of the transfer. If one is not found, the appended data starts after the last word in the file and no control record is created for the transfer.

Example of VAX Access From FTP/VE

Figure 3-1 shows an example session in which the user logs on to a CYBER 930 and establishes a connection from the NOS/VE Client to the VAX Server.

He then invokes FTP/VE to connect to the FTP/NOS Server on the remote host.

After validation information has been provided to the VAX, commands for the VAX can be processed.

```
You may enter CDCNET commands.
crec cyber
Connection $A created.

Enter validation for service access.
User:  username,password
Welcome to the NOS/VE Software System.
Copyright Control Data 1983, 1987.
CYBER 930 Class SN106.  NOS/VE R1  16471  08R9
1988-02-03.  13.17.38,16.


ftp/ftp vax

Copyright Control Data Corporation 1987, all rights reserved.
User  (Remote Host: VAX):
? username

Password  (Remote Host: VAX):
? password

ftp/change_verbose_mode mode=on

ftp/change_debug_mode mode=on

ftp/display_directory
--> PORT 192,12,251,145,4,71
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1095).
--> LIST
150 Opening data connection for /bin/ls (192.12.251.145,1095) (0 bytes).
Data connection from : VAX (192.12.251.32, 20) is accepted.
total 3365
-rw-r--r--  1 aea      wheel      387348 Jul  1  1987 a
-rw-r--r--  1 aea      wheel      374513 Jun 18  1987 aa
-rw-r--r--  1 aea      wheel      387348 Jun 28  1987 aaa
226 Transfer complete.
Transfer complete: retrieved 1093 bytes in 1.002 sec (1.065 Kbytes/sec).
```

Figure 3-3. VAX Access From FTP/VE

(Continued)

Example of VAX Access From FTP/VE

(Continued)

```
ftp/disfo
 Auto_Login_Mode          :  ON
 Debug_Mode               :  ON
 Bell_Mode                :  OFF
 Expression_Evaluation    :  OFF
 Local_Byte_Size          :  8
 Input_File               :  $INPUT
 Output_File              :  $OUTPUT
 Port Command Mode        :  ON
 Transfer_Format          :  Non_Print
 Transfer_Mode            :  Stream
 Transfer_Structure       :  File
 Transfer_Type            :  ASCII
 Verbose_Mode             :  ON
 Remote Host Name         :  VAX
 Control_Connection_ID    :  1
 Data_Connection_ID       :  -
 Remote_Control_Port      :  21
 Remote_Data_Port         :  -
 Remote_Internet_Number   :  192.12.251.32
 Local_Control_Port       :  1093
 Local_Data_Port          :  -
 Local_Internet_Number    :  192.12.251.145

ftp/help
 Command Name                    Command Aliases
 ------------                    ---------------
 APPEND_FILE                     APPE, APPEND, APPF
   Append to one or more remote files
 CHANGE_AUTO_LOGIN_MODE          CHAALM
   Turn Auto_Login mode on or off
 CHANGE_BELL_MODE                BELL, CHABM
   Turn bell prompt on or off
 CHANGE_DEBUG_MODE               CHADM, DEBUG
   Turn debug messages on or off
        .
        .
        .
Use DISPLAY_COMMAND_INFORMATION (DISCI) to get parameter info on a command.
Note: * indicates commands may not be available on other systems.

ftp/create_directory name=test
--> MKD test
257 MKD command successful.

ftp/change_working_directory name=test
--> CWD test
250 CWD command successful.

ftp/display_directory
--> PORT 192,12,251,145,4,73
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1097).
--> LIST
150 Opening data connection for /bin/ls (192.12.251.145,1097) (0 bytes).
Data connection from : VAX (192.12.251.32, 20) is accepted.
total 0
226 Transfer complete.
Transfer complete: retrieved 9 bytes in .071 sec (.123 Kbytes/sec).
```

Figure 3-3. VAX Access From FTP/VE

(Continued)

(Continued)

```
ftp/send testfile
--> PORT 192,12,251,145,4,74
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1098).
--> STOR testfile
150 Opening data connection for testfile (192.12.251.145,1098).
Data connection from : VAX (192.12.251.32, 20) is accepted.
Transfer complete: sent 45 bytes in .011 sec (4.067 Kbytes/sec).
226 Transfer complete.

ftp/display_directory
--> PORT 192,12,251,145,4,75
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1099).
--> LIST
150 Opening data connection for /bin/ls (192.12.251.145,1099) (0 bytes).
Data connection from : VAX (192.12.251.32, 20) is accepted.
total 1.
-rw-r--r--  1 aea      wheel          43 Feb  3 05:29 testfile
226 Transfer complete.
Transfer complete: retrieved 73 bytes in .241 sec (.296 Kbytes/sec).

ftp/get_file rf=testfile lf=testfile2
--> PORT 192,12,251,145,4,76
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1100).
--> RETR testfile
150 Opening data connection for testfile (192.12.251.145,1100) (43 bytes).
Data connection from : VAX (192.12.251.32, 20) is accepted.
226 Transfer complete.
Transfer complete: retrieved 45 bytes in .076 sec (.577 Kbytes/sec).

ftp/disc
   FILE: SCU_EDITOR_PROLOG
   FILE: TESTFILE
   FILE: TESTFILE2

ftp/delf testfile
--> DELE testfile
250 DELE command successful.

ftp/dir
--> PORT 192,12,251,145,4,78
200 PORT command successful.
Waiting for a data connection from VAX at (192.12.251.145, 1102).
--> LIST
150 Opening data connection for /bin/ls (192.12.251.145,1102) (0 bytes).
Data connection from : VAX (192.12.251.32, 20) is accepted.
total 0
226 Transfer complete.
Transfer complete: retrieved 9 bytes in .090 sec (.097 Kbytes/sec).

ftp/quit
--> QUIT
221 Goodbye.
/
```

Figure 3-3. VAX Access From FTP/VE

Example of FTP/NOS Access from FTP/VE

Figure 3-1 shows an example session logging on to a CYBER 930 and establishing a connection from the NOS/VE Client to the NOS Server.

After establishing a connection to NOS/VE and logging in, FTP/VE is invoked to establish a connection to the FTP/NOS Server on the remote host.

After validation information has been provided to FTP/NOS, commands for FTP/NOS can be processed.

```
You may enter CDCNET commands.
crec cyber_930
Connection $A created.

Enter validation for service access.
User:  username,password
Welcome to the NOS/VE Software System.
Copyright Control Data 1983, 1987.
CYBER 930 Class SN106.  NOS/VE R1  155335 08R0
1987-10-03.  19:07:38.

/ftp nos_f

Copyright Control Data Corporation 1987, all rights reserved.
User  (Remote Host: NOS_F):
? nosuser

PASSWORD  (REMOTE HOST: NOS_F):
? nospw

ftp/dir
    FILE NAME  FILE TYPE       LENGTH DN CREATION  ACCESS    DATA MOD
     PASSWORD  COUNT        PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
      EXPIRES   LEVEL  PR BR AC RS
       CHARGE NO. PROJECT NUMBER

      BUSYFIL  DIR. PRIVATE         0 40 87/12/07. 87/12/07. 87/12/07.
                    0          READ       10.25.55. 10.25.55. 10.25.55.
                         N  Y  Y  D
       7195       43EV8550

ftp/debug on

ftp/verbose on

ftp/send eg_prolog eprolog
--> PORT 192,5,209,61,4,46
200 COMMAND OK
Waiting for a data connection from NOS_F at (192.5.209.61, 1070).
--> STOR eprolog
150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION
Data connection from : NOS_F (192.5.209.201, 20) is accepted.
Transfer complete: sent 267 bytes in .075 sec (3.472 Kbytes/sec).
226 CLOSING DATA CONNECTION; REQUESTED FILE ACTION SUCCESSFUL
```

Figure 3-4. FTP/NOS Access From FTP/VE

(Continued)

(Continued)

```
ftp/change_expression_evaluation mode=on

ftp/send 'eg_prolog' 'eprolog,cs=88'
--> PORT 192,5,209,61,4,47
200 COMMAND OK
Waiting for a data connection from NOS_F at (192.5.209.61, 1071).
--> STOR eprolg8,cs=88
150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION
Data connection from : NOS_F (192.5.209.201, 20) is accepted.
Transfer complete: sent 267 bytes in .017 sec (15.347 Kbytes/sec).
226 CLOSING DATA CONNECTION; REQUESTED FILE ACTION SUCCESSFUL

ftp/get 'eprolg8,cs=88' 'eg_prolog2'
--> PORT 192,5,209,61,4,48
200 COMMAND OK
Waiting for a data connection from NOS_F at (192.5.209.61, 1072).
--> RETR eprolg8,CS=88
150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION
Data connection from : NOS_F (192.5.209.201, 20) is accepted.
226 CLOSING DATA CONNECTION; REQUESTED FILE ACTION SUCCESSFUL
Transfer complete: retrieved 267 bytes in .034 sec (7.708 Kbytes/sec).
ftp/change_expression_evaluation mode=off

ftp/change_transfer_type mode=binary
--> TYPE I
200 COMMAND OK

ftp/send eg_prolog ebin
--> PORT 192,5,209,61,4,50
200 COMMAND OK
Waiting for a data connection from NOS_F at (192.5.209.61, 1074).
--> STOR ebin
150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION
Data connection from : NOS_F (192.5.209.201, 20) is accepted.
Transfer complete: sent 375 bytes in .057 sec (6.378 Kbytes/sec).
226 CLOSING DATA CONNECTION; REQUESTED FILE ACTION SUCCESSFUL

ftp/delete ebin
--> DELE EBIN
250 REQUESTED FILE ACTION OKAY, COMPLETED.

ftp/remotehelp
--> HELP
214-THE FOLLOWING COMMANDS ARE RECOGNIZED (* =>'S UNIMPLEMENTED).

   ABOR*       DELE        NOOP        REIN*       SITE*       SYST*
   ACCT        HELP        PASS        REST*       SMNT*       TYPE
   ALLO*       LIST        PASV        RETR        STAT*       USER
   APPE        MKD *       PORT        RMD *       STOR
   CDUP*       MODE        PWD *       RNFR        STOU*
   CWD *       NLST        QUIT        RNTO        STRU

214 HELP COMMAND COMPLETED.

ftp/quit
--> QUIT
221 SERVICE CLOSING TELNET CONNECTION

ftp/QUIT
/
```

Figure 3-4. FTP/NOS Access From FTP/VE

Correspondence of User Commands and FTP Protocol Commands

Table 3-3 lists the FTP/VE user commands and correlates them with the RFC 959 FTP protocol commands.

Table 3-3. Correspondence of User Commands and FTP Protocol Commands

FTP User Commands	FTP Protocol Commands
APPEND_FILE	APPE
CHANGE_AUTO_LOGIN_MODE	
CHANGE_BELL_MODE	
CHANGE_DEBUG_MODE	
CHANGE_EXPRESSION_EVALUATION	
CHANGE_FILE_NAME	RNFR, RNTO
CHANGE_INPUT_FILE	
CHANGE_OUTPUT_FILE	
CHANGE_PORT_COMMAND_MODE	
CHANGE_TRANSFER_TYPE	TYPE
CHANGE_VERBOSE_MODE	
CHANGE_WORKING_DIRECTORY	CWD
CREATE_DIRECTORY	MKD
CREATE_HOST_CONNECTION	
DEFINE_AUTO_LOGIN	
DELETE_DIRECTORY	RMD
DELETE_FILE	DELE
DELETE_HOST_CONNECTION	QUIT
DISPLAY_DIRECTORY	LIST
DISPLAY_FILE_NAMES	NLST
DISPLAY_FTP_OPTIONS	
DISPLAY_LOCAL_HELP	
DISPLAY_REMOTE_HELP	HELP
DISPLAY_WORKING_DIRECTORY	PWD
ELEVATE_WORKING_DIRECTORY	CDUP
GET_FILE	RETR
LOGIN_USER	USER, PASS, ACCT
QUIT	QUIT
REPLACE_FILE	STOR
SEND_FTP_COMMAND	<as specified>

Appendixes

Glossary A

A

ARPANET

A Defense Data Network (DDN) developed by the Defense Advanced Research Projects Agency. ARPANET supports research and development projects funded by the Department of Defense.

D

DDN

Refer to the Defense Data Network.

Defense Data Network (DDN)

A packet switching network provided by the Department of Defense (DOD) to meet its current and projected data communication requirements. It is based upon the Defense Advanced Research Projects Agency Network (ARPANET), an existing operational network.

F

File Transfer Protocol (FTP)

TCP/IP protocol that provides the file transfer server and user functions.

FTP

Refer to File Transfer Protocol.

FTP Client

The FTP Client provides CYBER host terminal users and TCP/IP TELNET users access to file systems on remote hosts.

FTP Server

The FTP Server provides remote TCP/IP users and applications access to the CYBER host file system.

G

Gateway

A software interface between systems with different architectures and protocols.

I

Internet Protocol (IP)

A term used in DDN networks that refers to a connectionless, point-to-point protocol corresponding to the CDCNET Internet Layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

IP

Refer to Internet Protocol.

M

MILNET

A Defense Data Network (DDN) evolved from ARPANET that supports operational communication requirements.

S

Server TELNET

Provides a mechanism for an interactive terminal on a foreign host to communicate with the interactive services of CYBER hosts.

T

TCP

Refer to Transmission Control Protocol.

TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is the name given to a suite of protocols that support the ARPANET community. TCP/IP protocol implementation is required within CDCNET for connectability to Defense Data Networks (MILNET or ARPANET) and to workstations that use TCP/IP.

TELNET

A TCP/IP protocol that provides presentation layer services for other application protocols. The TELNET protocol is roughly equivalent to Virtual Terminal Protocol (VTP) in the ISO model. It establishes connections and controls interactive virtual circuits.

Transmission Control Protocol (TCP)

A term used in DDN networks that refers to an end-to-end, connection-oriented protocol corresponding to the CDCNET transport layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

U

User TELNET

Allows a CDCNET terminal to connect to an interactive service of a foreign host.

Index

Index

D

E

F

S

T

U

V

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technology & Publications Division
SVL104
P.O. Box 3492
Sunnyvale, CA 94088-3492

TCP/IP Usage 60000214 B

We value your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other _____

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up

What programming languages do you use? _____

_____

How do you like this manual? Check those questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Do the illustrations help you?

Comments? If applicable, note page and paragraph. Use other side if needed.

Would you like a reply? ☐ Yes ☐ No

From:

Name _____ Company _____

Address _____ Date _____

_____ Phone _____

_____

Please send program listing and output if applicable to your comment.