

Client FTP

This chapter describes Client FTP, the File Transfer Protocol (FTP) that supports file transfers among unlike hosts in diverse internetworking environments. It contains these sections:

- **Introducing Client FTP**
Provides a brief overview of the File Transfer Protocol.
- **Client FTP**
Illustrates how FTP works.
- **Invoking Client FTP**
Describes how to use Client FTP as both a TSO command processor and as a regular batch program.
- **Invoking FTP as a Batch Program**
Describes how to run the Client FTP program in batch as either a regular program or as a TSO command processor.
- **Client FTP Invocation Options**
Describes both the general and debug options available with the FTP command.
- **General Client FTP Operation**
Describes the general operation of the Client FTP program.
- **Client FTP Commands**
Describes each of the Client FTP commands and includes a table listing all the commands with a brief description of each.

Commands are:

? Command	ABOR	ADD	ALLO
BYE	CDUP	CONN	CWD
DELE	DO	END	EXPE
GET	HELP	LIST	LOG
MKD	MODE	NLST	PUT
PWD	QUIT	QUOT	REN
REST	RMD	SEND	SITE
SNDS	STAT	STRU	TYPE
A=B	A?B		

- Restart Support
Describes how to use the restart marker.
- Client FTP File Transfer Examples
Provides examples of some of the Client FTP commands.

Introducing Client FTP

The File Transfer Protocol (FTP) is an application protocol in the internet protocol suite. It supports file transfers among unlike hosts in diverse internetworking environments. Using FTP, you can move a file from one computer to another, even if each computer runs a different operating system and uses a different file storage format. Files can contain data, programs, text, or anything that can be stored online.

The objectives of the FTP protocol are to

- provide sharing of files (computer programs and/or data)
- encourage indirect or implicit (via programs) use of remote computers
- shield users from variations in file storage systems among hosts
- transfer data reliably and efficiently

The FTP is based on a model of files having a few attributes and a mechanism of commands and replies. The command - reply mechanism establishes the parameters for a file transfer and then performs the transfer. Like TELNET, FTP runs over TCP and assumes the service level provided by TCP.

These documents define FTP:

- *RFC 959, File Transfer Protocol (FTP)*
- *MIL-STD-1780, Military Standard File Transfer Protocol*

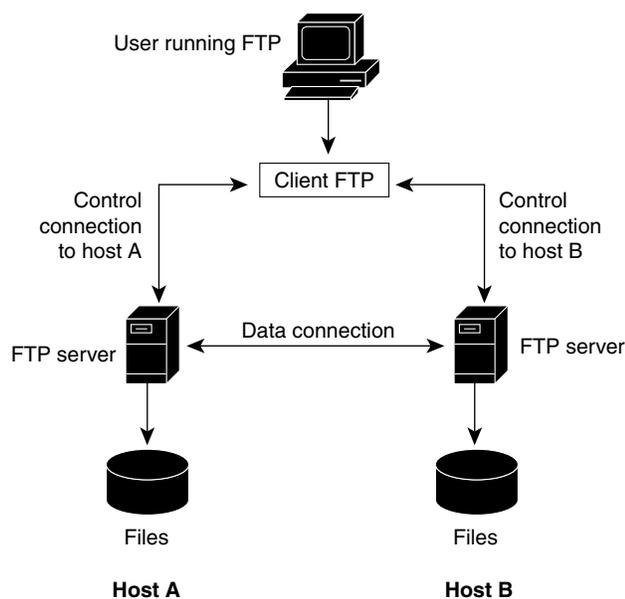
Two versions of Client FTP are provided with Cisco IOS for S/390: Client FTP and Client FTP2. This chapter describes Client FTP.

The Cisco IOS for S/390 Client FTP program uses a three-party model, or the server-to-server model. Client FTP requires the MVS TSO or batch user to sign on to two hosts that have Server FTP programs. After signing on to the two hosts, a user can transfer files between host **A** and host **B**.

Client FTP

Figure 4-1 shows the relationship between the Client FTP program and the two Server FTP programs in the three-party model.

Figure 4-1 Client FTP Program in the Three-Party Model



Through FTP, the user opens connections to two FTP servers. Both servers may be remote, or one may be a local server and the other a remote server. FTP maintains two control connections: One for communication with the Cisco IOS for S/390 FTP server and the second with the remote FTP server. The user issues command sequences to direct both servers.

Client FTP gives the user access to files on both systems. Data transfer occurs directly between the local and remote servers. Data does not pass through the Client FTP application.

Invoking Client FTP

The Client FTP program runs as both a TSO command processor and as a regular batch program. This means that Client FTP can be used as a TSO command and can be called as a regular program with MVS JCL.

TSO Invocation

In a TSO environment, Client FTP can be accessed as a TSO command processor or can be called as a program with the TSO **CALL** command.

Since Client FTP does not use full-screen facilities, it can be used from any type of terminal that is supported by TSO including 3270 systems, 3767 systems, and asynchronous ASCII terminals supported through TCAM, NTO, or NPSI.

Note You must have **PROMPT** set in your TSO profile for FTP to work properly in interactive mode.

FTP Command Processor

You can invoke Client FTP with the FTP TSO command as shown in the following example:

```
FTP [/ option1 option2 ...]
```

Client FTP responds with this message:

```
Cisco IOS for S/390 Rn Client FTP - Enter command or '?'
```

Read Client FTP Invocation Options for a description of the options for the Client FTP commands.

Note All FTP options must be preceded by a slash (/). If this character is omitted, the options are not recognized by Client FTP.

TSO CALL Command

Use the TSO **CALL** command in a TSO environment to invoke Client FTP.

```
CALL `T01TCP.LINK(FTP)` [/option1 option2 ..]
```

Note The data set name, **T01TCP.LINK**, might need to be replaced by the appropriate data set name at your installation. Check with your Cisco IOS for S/390 site administrator.

Usage Guidelines

Options, when specified, must be enclosed in single quotes.

When invoked by **CALL**, Client FTP runs as a program and not as a TSO command processor. No data sets need be allocated before invoking FTP in this fashion.

Under TSO, any data set needed by the Client FTP program is dynamically allocated and freed.

Invoking FTP as a Batch Program

The Client FTP program can be run in batch as either a regular program or as a TSO command processor by running it under a batch Terminal Monitor Program (TMP).

Note When running in batch mode, it is important to specify the Client FTP commands carefully, because the slightest error can cause all subsequent commands to fail and force you to rerun the batch job. For this reason, you should execute only one file transfer per batch job step. For PL/I V2.2.1 users, FMID PLIX150 must be installed. The Cisco IOS for S/390 **LINK** library must be placed before PL/I runtime libraries in your **STEPLIB** concatenation for client batch jobs (including **FTP**, **FTP2**, **ACPEEP**, or **TELNET**). Failure to do this may cause IBM002I, IBM004I, or IBM014I messages.

When run in batch, Client FTP sets a program condition code depending on the severity of Client FTP errors encountered; a return code of 0 indicates all commands entered processed successfully.

Batch Program

You can invoke Client FTP in batch in a manner similar to any other batch utility program.

Note The **FIOS** option is required in the batch environment.

Example

The following example downloads the host names table from the Network Information Center (NIC) to an existing sequential data set on your host.

```
//jobname    JOB job_stmt_parms
//FTP EXEC  PGM=FTP,PARM='/FIOS option1 option2 ..'
//STEPLIB   DD DSN=T01TCP.LINK, DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSPUT    DD SYSOUT=*,DCB=BLKSIZE=133
//SYSGET    DD *,DCB=BLKSIZE=80
CONN NIC
CONN your_hostname
A:LOG ANONYMOUS GUEST
B:LOG your_userid your_password
B:SITE BLKSIZE(4000) LRECL(260) RECFM(VB) SPACE(xx,y)
PUT <NETINFO>HOSTS.TXT 'your_hoststxt_dsname'
A:QUIT
B:QUIT
END
/*
```

Batch TMP

You can invoke Client FTP in batch as a TSO command processor by running it under a batch TMP.

Example

In the following example, the batch TMP program is **IKJEFT01**, which is the normal TSO TMP. The following example downloads a Request For Comments (RFC) from the NIC into a member of an RFC partitioned data set on your host:

```
//jobname JOB job_stmt_parms
//TMP EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*,DCB=BLKSIZE=133
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
FTP / FIOS
/*
//SYSPUT DD SYSOUT=*,DCB=BLKSIZE=133
//SYSGET DD *,DCB=BLKSIZE=80
CONN NIC
CONN your_hostname
A:LOG ANONYMOUS GUEST
B:LOG your_userid your_password
TYPE AT
PUT <RFC>RFCxxx.TXT 'your_rfctxt.dsname(RFCxxx) '
A:QUIT
B:QUIT
END
/*
```

Client FTP Invocation Options

This section describes the Client FTP invocation options and their requirements. These general notes apply to the Client FTP invocation options:

- No invocation options are required for the Client FTP program. If any are specified, they must be preceded by a slash (/) to meet the conventions of the PL/I runtime support package. In the previous examples a blank follows the (/), but it is not required.
- An option name can immediately follow a slash, as in **/FIOS**.
- Invocation options are not case sensitive.
- Specify option names exactly as shown. Abbreviations are not permitted.

Client FTP uses general options and debugging options. They are described in the following sections.

Client FTP General Options

This section describes the Client FTP general options.

APP

The **APP** option (in the form **APP=vtam_application_name**) identifies the exact VTAM application name where the client will connect. *vtam_application_name* is a 1- to 8-character name.

If the **APP** option is used, the **SYS** invocation option is ignored.

FIOS

Normally the FTP program interacts with the user through a terminal. The **FIOS** option lets the program read and execute a file containing commands and sends the results to a different file. Commands are read from a sequential file allocated to the **SYSGET DD** statement; execution results are written to a sequential file allocated to the **SYSPUT DD** statement.

Example

Under TSO, the files can be allocated as shown here:

```
ALLOCATE FILE(SYSGET) DATASET(input_dataset)
ALLOCATE FILE(SYSPUT) DATASET(output_dataset)
```

The input file should have all the necessary commands (such as, **CONN**, **LOG**) and should be unnumbered. The output file should have a blocksize of 133.

Note You must use the **FIOS** option when running FTP in batch.

LOGT

The **LOGT** option displays the current time before each line is sent to the terminal. This option is automatically set if either the **TEST** or **FIOS** option is specified.

SYS

The **SYS** option, in the form **SYS=*x***, where *x* is an arbitrary character, identifies an alternate Cisco IOS for S/390 VTAM application to handle the TELNET connections established by the Client FTP program. This option is useful in cases where multiple copies of Cisco IOS for S/390 are running concurrently. You can specify **SYS=*x*** to access a network server called **ACCES*x*** instead of the usual **ACCES**.

WAIT

The **WAIT** option causes the terminal keyboard to remain locked while a data transfer is in progress. Normally the keyboard remains unlocked, allowing additional commands to be entered during a data transfer.

This option is automatically set if the **FIOS** option is specified.

Client FTP Debugging Options

The Client FTP debugging options obtain debugging information on the internal operation of the Client FTP program and the interactions between the Client FTP program and the Server FTPs. Normally, the debugging information displays on the terminal, but if the **FIOS** option is in effect, the information is written to the data set represented by the **SYSPUT DD** statement.

Note Use these options only under the direction of Cisco IOS for S/390 Customer Support.

DISP

Use the **DISP** option to display all requests sent and all responses received on the control connections to each Server FTP.

Examples

Requests sent from the Client FTP program to a Server FTP use the following format:

DISP - OUTPUT - *text of User-to-Server request*

Responses sent from a Server FTP to the Client FTP program have the following format:

DISP - *text of Server-to-User response*

TEST

Use the **TEST** option to obtain detailed status information from the various internal calls in the Client FTP program as well as the information provided by the **DISP** option. This option can be turned on or off while the program is running by issuing the **DEBUG** command.

Example

Information logged by the **TEST** option has the following format:

TEST - *TEST debugging information text*

TESTI

Use the **TESTI** option to obtain local terminal input and output information, as well as to provide the **TEST** and **DISP** information.

Example

Information logged by the **TESTI** option has the following format:

TESTI - *TESTI debugging information text*

VLT

The **VLT** option turns on tracing of the virtual line terminal sessions associated with the FTP session. This option is useful for debugging VTAM problems between the FTP session and the Cisco IOS for S/390 address space. The **VLT** option generates an enormous amount of output. When used interactively, this output comes to the terminal. When used in batch, the output is written to the **SYSVLT DD**. Thus, when running FTP in batch, you must add this **DD** statement to the JCL:

Example

```
//SYSVLT DD SYSOUT=*,DCB=BLKSIZE=133
```

General Client FTP Operation

These steps outline the general procedure for using this Client FTP program:

- Connect to the two host Server FTPs using the **CONN** *host* command. The first host connected is the A: host, and the second is the B: host.

- Use these commands to log in to each host Server FTP:

A:LOG *userid_a password_a*

B:LOG *userid_b password_b*

- Set the appropriate file transfer parameters (such as **MODE**, **STRU**, **TYPE**).
- Perform the desired transfer operation (such as **GET**, **PUT**).

Path Name

A path name is a string that is provided to a file system to identify a file. A path name usually contains a device and/or directory name and a file name. The FTP specification does not specify a standard path name convention. You must follow the file naming conventions of the file systems involved in the transfer. Consult personnel at the host sites involved in the transfer for file naming conventions.

Many of the Client FTP commands take one or more path name arguments.

For information about the syntax for MVS path names supported by the Cisco IOS for S/390 Server FTP, read Data Set Names in Server FTP.

Client FTP Command Conventions

These general notes apply to the Client FTP commands:

- To indicate successful completion of most commands, the Client FTP program gives a new prompt. However, when a data transfer command is issued, a prompt appears when the operation begins successfully. You can then enter other commands (such as, status requests with the **STAT** command) while the operation proceeds. Final completion of the data transfer command is indicated with a message.
- You can use the **DISP** invocation option to see the specific FTP commands and responses (according to *RFC 959*) sent and received over the control connections as a result of Client FTP commands.
- The Client FTP commands are not case sensitive.
- The commands must be specified exactly as shown. Abbreviations are not permitted.
- In the examples, parameters enclosed by brackets are optional for the command line. In many cases, if the optional parameters are omitted from the command line, you are prompted for them.
- In all examples of Client FTP input and output in this manual, user entries are shown in boldface type.
- The Client FTP examples in this manual assume that you have issued **CONN** and **LOG** commands similar to the following example to connect an IBM MVS system to another system:

Example

```

Cisco IOS for S/390 Rn Client FTP - Enter command or '?'
conn mvs
A:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter command or HELP
conn unix
B:220 unix FTP server (SunOS 4.1) ready.
a:log myid
331 Enter PASS command
Password:
b:log myid
331 Enter PASS command
Password:
    
```

The Host Prefix

Because the Client FTP program implements the three-party model in which the Client FTP program establishes control connections to two Server FTPs, you need a way to identify which connection is to be used for certain commands. For example, you might want to provide one set of site parameters with the **SITE** command to one Server FTP and provide a different set of parameters to the other Server FTP. This Client FTP program uses host prefixes to differentiate between the two Server FTPs.

A host prefix is one of the A: or B: character strings. When specified with a Client FTP command, the host prefix precedes the command, as in **A:SITE**. A host Server FTP initially is assigned to one prefix or the other when the connection is established with the **CONN** command. The **A=B** command can switch the host prefixes from A: to B: and B: to A: when necessary. The **A?B** command shows the host name associated with each host prefix in case you become confused.

Client FTP command host prefixes can be required, not allowed, ignored, or optional.

- Commands that require a host prefix must be preceded by an A: or B:.
- Commands for which a host prefix is not allowed must not be preceded by an A: or a B:.
- Commands for which a host prefix is ignored can either have one or not; in any case, the prefix is ignored.
- Commands for which a host prefix is optional can either have a prefix, or not. If you give a prefix, the command applies only to that specific host. If you omit the prefix, the command applies to both hosts, provided the connection to the host is open. The exception is the **CONN** command. If a prefix is omitted, the first available, unused host prefix is assigned to the connection.

Client FTP Commands

This section describes the Client FTP commands. Table 4-1 gives a brief description of each command and its function. The remainder of this chapter provides detailed descriptions of the commands.

Table 4-1 Client FTP Commands

Command	Function	Host Prefix Requirement
?	Get help on all commands or one command	Ignored
ABOR	Abort transfer data	Optional
ADD	Append a file from host A to a file at host B	Not allowed
ALLO	Set file size	Required

Table 4-1 Client FTP Commands (Continued)

Command	Function	Host Prefix Requirement
BYE	Terminate program (same as QUIT)	Required
CDUP	Change to parent of current working directory	Required
CONN	Control connection between Client FTP program and a Server FTP	Optional
CWD	Change current directory	Required
DELE	Delete a file on the remote host	Required
DO	Execute TSO command	Ignored
END	Terminate program	Not allowed
EXPE	Toggle experimental mode for directory commands	Ignored
GET	Copy file from remote host to local host (same as RECV)	Not allowed
HELP	Ask local host for command information	Optional
LIST	Provide current information for files in a specified path name	Not allowed
LOG	Login user	Required
MKD	Make a directory	Required
MODE	Set transmission mode	Optional
NLST	List filenames in specified directory	Not allowed
PUT	Copy file from local host to remote host (same as SEND)	Not allowed
PWD	Show name of current working directory (on the remote host)	Required
QUIT	Terminate program (same as BYE)	Optional
QUOT	Send an FTP command to the remote Server	Required
REN	Rename file from old name to new name (on the remote host)	Required
REST	Restart	Required
RMD	Send an FTP command to the remote Server	Required
SEND	Copy file from local host to remote host (same as PUT)	Not allowed
SITE	Send local host-dependent information	Required
SNDS	Re-send last SITE command to local host	Required
STAT	Ask local host for status	Optional
STRU	Set file structure	Optional
TYPE	Set data type	Optional
A=B	Switch host prefixes associated with each Server FTP connection	Ignored
A?B	Show current connections associated with the A and B host prefixes from the Client FTP program	Ignored

? Command

The ? command obtains information about using the Client FTP program.

? [*command_name*]

Syntax Description

command_name Command for which information is requested.

Default

If no arguments are specified, a list of Client FTP commands is displayed.

Usage Guidelines

Each command name is preceded by a special character that indicates the requirement for host prefixing for that particular command. A key is provided in the display to explain the prefix requirement indicated by each special character.

If a command name is specified as an argument on the ? command, a line of information appears, showing

the command syntax

a short description of the command function

host prefix requirements for that command

The ? command ignores a host prefix.

Example

The following example shows a request for information about using the **log** command.

```
? log
LOG <USERID> <CURRENT_PASSWORD> </NEW_PASSWORD> - LOGIN USER
(PREFIX REQUIRED).
```

Related Command

HELP Requests help from a server FTP program. (By contrast, the ? command requests help from the Client FTP)

ABOR

The **ABOR** command instructs the Server FTP to abort the previous command and any associated transfer of data. No action is taken by the Server FTP if the previous command has been completed (including data transfer). The control connection to the Server FTP is not closed, but the data connection is closed.

ABOR

Syntax Description

This command has no arguments or keywords.

Default

If no prefix is given, the **ABOR** request is directed to both Server FTPs.

Usage Guidelines

A host prefix is optional with the **ABOR** command. If a host prefix is specified, the **ABOR** request is directed only to the specified Server FTP.

Example

```
PUT 'scm.p015235.dump' tempdump
A:150-Data set open with attributes:
A:Type A N   Tabs 8   Stru F   Mode S   Path SCM.P015235.DUMP
A:Volser SYSPK1   Unit SYSALLDA   Dsorg PS   Recfm VB   Lrecl 138
A:Blksize 23476   Rlse
A:150
B:150 ASCII data connection for tempdump (138.42.224.15,4127).
ABOR
B:552 tempdump: Connection reset by peer.
A:426-Data transfer aborted
A: 57344 bytes sent in 13.3 seconds (4295 bytes/s)   Path SCM.P015235.DUMP
A:User MYID   Data bytes sent 66334
A:Disk tracks read 1
A:426
B:225 ABOR command successful.
A:226 Abort command completed.
```

ADD

The **ADD** command requests that a file from host **A** be appended to a file at host **B**. The file retrieved is always at host **A** and the target file is always at host **B**. You might need to use the **A=B** command to switch host prefixes so that the correct host is associated with the correct host prefix. See the section on **A=B** for details about the **A=B** command. See also the **PUT Example** and **GET Example** sections for examples of how the **A=B** command is used.

```
ADD [path_name_a] [path_name_b]
```

Syntax Description

path_name_a File name of the file to be retrieved from host **A**.

path_name_b File name on host **B** to which that file from host **A** is to be appended.

Default

If either file name is omitted, you are prompted for the file name.

Usage Guidelines

The syntax for each path name depends on the associated Server **FTP**.

A host prefix is not allowed with the **ADD** command.

Example

```
ADD n.d tempdump
A:150-Data set open with attributes:
A:Type A N  Tabs 8   Stru F   Mode S   Path MYID.N.D
A:Volser COLPK1  Unit SYSALLDA  Dsorg PS   Recfm FB   Lrecl  80
A:Blksize 3120   Rlse
A:150
B:150 ASCII data connection for tempdump (138.42.224.15,4128).
A:226-Transfer Complete
A: 3439 bytes sent in 2.49 seconds (1381 bytes/s)   Path MYID.N.D
A>User MYID   Data bytes sent 6480
A:Disk tracks read 1
A:226
B:226 ASCII Transfer complete.
```

ALLO

The **ALLO** (allocate) command provides a file size to those Server FTPs that require it. Refer to the documentation for your particular Server FTP to see if you need this command. The Cisco IOS for S/390 Server FTP does not require use of the **ALLO** command; it can be used optionally to cause records to be truncated or to allocate space.

ALLO integer1 [*R integer2*]

Syntax Description

integer1 Number of logical bytes of storage to be reserved for the file.

R integer2 Optional; specifies the maximum record or page size.

Usage Guidelines

For files sent with record or page structure, a maximum record or page size (in logical bytes) can be required. This is indicated by the second integer argument.

The *integer2* argument is optional, but if it is specified, it must be separated from the first by the three characters, “ **R** ” (space **R** space).

A host prefix is required with the **ALLO** command.

Example

In the following example, the **ALLO** command truncates records to a length of 60 bytes during a file transfer between two Cisco IOS for S/390 systems.

```
STRU R
b:site lrecl(60) blk(6000)
b:allo 12000 r 60
PUT n.d allor.data
A:150-Data set open with attributes:
A:Type A N   Tabs 8   Stru R   Mode S   Path MYID.N.D
A: Volser COLPK1   Unit SYSALLDA   Dsorg PS   Recfm FB
A: Lrecl 80   Blksize 3120   Rlse
A:150
B:150-Data set open with attributes:
B:Type A N   Tabs 8   Stru R   Mode S   Recall 5
B:Path MYID.ALLOR.DATA
B:Volser HAGCAT   Unit SYSALLDA   Dsorg PS   Recfm FB
B:Lrecl 60   Blksize 6000   Space 2 1 Tracks Rlse   Maxr 60
B:150
A:226-Transfer Complete
A: 3441 bytes sent in 0.32 seconds (10753 bytes/s)   Path MYID.N.D
A:User MYID   Data bytes sent 6480
A:Disk tracks read 1
A:226
B:226-Transfer Complete
B: 3441 bytes received in 0.44 seconds (7820 bytes/s)
B:Path MYID.ALLOR.DATA   User MYID   Data bytes received 3277
B:Disk tracks written 1   Records truncated 21
B:Records padded 80   Records folded 20
B:226
```

BYE

The **BYE** command is the same as the **QUIT** command. Read **QUIT** for details on the **BYE** and **QUIT** commands.

BYE

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

A host prefix is required with the **BYE** command.

Example

```
A: BYE  
A:221 Goodbye.
```

Related Commands

END	Can be used instead of BYE ; END does not require a host prefix.
QUIT	Can be used instead of BYE ; QUIT requires a host prefix and takes no arguments.

CDUP

The **CDUP** command directs a Server FTP to change the current directory to the parent directory of the old current directory. The **CDUP** command is most useful when the Server FTP manipulates a hierarchical file system such as UNIX.

CDUP

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

A host prefix is required with the **CDUP** command.

Example

In the following example, a UNIX Server FTP in session with the Client FTP program has `/u/user1/work` as the current directory. If a **CDUP** command is issued by the Client FTP to that UNIX Server FTP, the resulting current directory is the parent of the old current directory (`/u/user1`).

```
B: CDUP  
B:250 CWD command successful.
```

CONN

The **CONN** command sets up one control connection between the Client FTP program and a Server FTP. Since the Cisco IOS for S/390 Client FTP uses the third-party model, two such connections must be established, one to each Server FTP that participates in the file transfer. If the same Server FTP is used as both Server FTPs, two **CONN** commands are required. Since the **CONN** command sets up the control connections to the Server FTPs, issue the **CONN** commands first when using the Client FTP program.

CONN [*hostname*]

Syntax Description

CONN Command used to establish connections between client programs and Server FTP.

hostname Provides the host name to which one FTP control connection is established. If you omit the host name, you are prompted to provide one.

Default

If the host prefix is omitted, the first available unused host prefix is assigned to that connection.

Usage Guidelines

Host name strings must correspond to the syntax specified in Using Host Name Strings in Introduction to Cisco IOS for S/390.

A host prefix is optional with the **CONN** command. If a host prefix is specified, the connection is assigned to that host prefix.

Example

```
CONN 10.1.1.99  
A:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter command or HELP  
CONN unix  
B:220 unix FTP server (SunOS 4.1) ready.
```

CWD

The **CWD** command requests that a Server FTP change the current directory being maintained for you by the Server to a new directory.

CWD [*path_name*]

Syntax Description

path_name Indicates to the Server FTP the name of the directory to be made the current directory.

Default

If you omit *path_name*, the Client FTP program prompts for one.

Usage Guidelines

The syntax for *path_name* depends on the associated Server FTP.

A host prefix is required with the **CWD** command.

Example

In the following example, a UNIX Server FTP in session with the Client FTP program has `/u/user1/work` as the current directory. If a **CWD junk** command is issued by the Client FTP to that UNIX Server FTP, the resulting current directory is the junk subdirectory of the old current directory (`/u/user1/work/junk`). The same result is achieved by specifying **CWD /u/user1/work/junk**.

```
B: CWD /u/myid/acces
B:250 CWD command successful.
```

DELE

The **DELE** (delete) command directs a Server FTP to delete the specified file.

DELE [*path_name*]

Syntax Description

path_name Specifies the specific file to delete.

Default

If you omit *path_name*, you are prompted to provide one.

Usage Guidelines

The syntax for *path_name* depends on the associated Server FTP.

A host prefix is required with the **DELE** command.

Example

```
A:DELE t1.data  
A:250 Deleted OK.
```

DO

The **DO** (do TSO) command requests the Client FTP program to execute a TSO command for you.

DO *tso_command parameters*

Syntax Description

tso_command TSO command followed by any parameters to be passed to the TSO command.

Usage Guidelines

The **DO** command is handled by the Client FTP program.

A host prefix is not necessary and is ignored if included with the command.

Note In batch mode, the TSO environment is required for a **DO** command to work.

Example

```
A:DO listc l(myid)
IN CATALOG:CATALOG.MVSICF1.VMVSTSO
MYID.ACCE.S.ASM
MYID.LIB.LOAD
MYID.T.D
MYID.VBIG.D
MYID.VB.D.
```

END

The **END** command terminates the Client FTP program. This is typically the last command you enter. Any open control connections are closed before the program terminates.

END

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

Any host prefix is ignored.

Related Commands

You can use the **BYE** and **QUIT** commands interchangeably with **END**. Both **BYE** and **QUIT** require a host prefix; **END** does not.

BYE Can be used instead of **END**; **BYE** requires a host prefix.

QUIT Can be used instead of **BYE**; **QUIT** requires a host prefix and takes no arguments.

EXPE

The **EXPE** command toggles the use of experimental or regular directory commands. Since there is no consistent support for this command, it is recommended that you not use this command.

Note The directory commands were added to FTP subsequent to the initial FTP specification and are documented in *RFC 959, File Transfer Protocol (FTP)*, Appendix II, Directory Commands.

EXPE

Syntax Description

No operands are associated with the **EXPE** command, since it is a Client FTP command.

Usage Guidelines

Any host prefix is ignored.

Table 4-2 shows the FTP command that is sent over the control connection for each directory command with an **EXPE** setting:

Table 4-2 Using Directory Commands with an EXPE Setting

Client FTP Command	Regular	Experimental
MKD	MKD	XMKD
RMD	RMD	XRMD
PWD	PWD	XPWD
CDUP	CDUP	XCUP

GET

The **GET** command requests that a file from host B be copied to a file on host A by the appropriate Server FTPs. The file to be retrieved is always at host B, and the file to be copied to is always at host A. You might need to use the **A=B** command to switch host prefixes to get the correct host. An alternative to switching host prefixes is to use the **PUT** command; read the **PUT** section.

```
GET [path_name_b] [path_name_a]
```

Syntax Description

path_name_b File name to be retrieved from host B.

path_name_a File name at host A into which the file from host B is to be copied.

Default

If you omit either path name, you are prompted for one.

Usage Guidelines

The syntax for each path name depends on the associated Server FTP.

A host prefix is not allowed with the **GET** command.

Example

```
GET jclbr14 cntl(newbr14)  
A:150-Data set open with attributes:  
A:Type A N  Tabs 8  Stru F  Mode S  Recall 5  
A:Path MYID.CNTL(NEWBR14)  
A:Volser COLPK1  Unit SYSALLDA  Dsorg PS  Recfm FB  
A:Lrecl 80  Blksize 3120  Space 31 15 Tracks Rlse  Dir 5  
A:150  
B:150 ASCII data connection for jclbr14 (138.42.128.13,4106) (810 bytes).  
A:226-Transfer Complete  
A: 820 bytes received in 3.24 seconds (253 bytes/s)  
A:Path MYID.CNTL(NEWBR14)  User MYID  Data bytes received 800  
A:Disk tracks written 1  
A:226  
B:226 ASCII Transfer complete.
```

HELP

The **HELP** command requests help information from one or both of the Server FTPs.

HELP [*text*]

Syntax Description

text Any command for which more information or usage guidelines are needed.

Default

If no host prefix is given, help is requested from each host with an open control connection.

Usage Guidelines

HELP has no required arguments.

Any operands specified on the **HELP** command are passed to Server FTP unchanged and are interpreted by Server FTP. Different Server FTPs may interpret these operands differently.

A host prefix is optional with the **HELP** command. If a host prefix is specified, help information is requested only from that host.

Example

```
A:HELP REST
--- HELP ---
*** HELP REST ***
FTPREST (Restart) Command:
Function: Specifies that the data transfer command which follows
(immediately) is to restart at a specified intermediate
point in the file.
Syntax: REST <marker>
Notes:
(1) After a REST command, STOR and APPE have identical meanings.
(2) Data transfer must be MODE B (block).
(3) A file RETRIEved will normally include restart
markers approximately every 32767 bytes. The REST
parameter on the SITE command allows you to change
this interval or even entirely suppress restart markers.
See HELP SITE. The actual decision to send a marker
depends on a count of data bytes read from the disk
not including OS count/control bytes). When this count
reaches the limit, the marker is sent at the next end
of a complete logical record, segment of a spanned record
(if RECFM includes VS), or a physical disk block (if
RECFM is U, V, or F).
(4) FTP can accept (and send) restart markers in either
```

STRU F or STRU R.
(5) FTP restart markers consist of 10 characters, which are the hex representation of five 8-bit bytes: TTRBB. Here "TTR" forms a standard OS disk block address, and BB is a byte offset within the block.
214 <end of HELP>

Related Command

? command Requests help from the Client FTP program.

LIST

The **LIST** command requests a Server FTP to provide current information for files corresponding to a specified path name. This contrasts to the **NLST** command that provides only a list of file names without any other file information.

LIST [*path_name_a*] [*path_name_b*]

Syntax Description

path_name_a Path name to be listed from host A.

path_name_b File name at host B into which the list from host is copied.

Default

If you omit either path name, you are prompted to provide one.

Usage Guidelines

If the path name specifies a directory or other group of files, the Server FTP transfers current information for a list of files.

If the path name specifies a single file, the Server FTP transfers current information on that file.

The syntax for each path name depends on the associated Server FTP.

A host prefix is not allowed with the **LIST** command.

Example

```
LIST acces ibmcatent
A:125 Transfer started
B:150 ASCII data connection for ibmcatent (138.42.128.13,4107).

A:226-Transfer Complete
A: 432 bytes sent in 1.03 seconds (419 bytes/s) Path MYID.ACCES
A>User MYID Data bytes sent 420
A:226
B:226 ASCII Transfer complete.
```

LOG

The **LOG** (login) command identifies the user by sending a user ID and password to a Server FTP.

You may change your password when logging into the Cisco IOS for S/390 server. You typically issue the **LOG** command immediately following the **CONN** command.

```
LOG [userid] [current_password] / [new_password]
```

Syntax Description

userid User name.

current_password User's current password.

new_password New password to be applied to the account.

Default

If you omit either the *userid* or *current_password*, the Client FTP program prompts you for them.

Usage Guidelines

The */new_password* parameter is a 1 to 8 character string password. The new password replaces the current password after the user ID and current password are validated. The new password option is valid only when in a session with a Cisco IOS for S/390 server. The slash (/) must follow the current password without any intervening blanks. The new password must follow the slash without any intervening blanks.

If the Server FTP requires additional accounting information during the user identification process, the Client FTP program prompts you to enter the accounting data.

A host prefix is required with the **LOG** command.

Example

```
B:LOG myid  
331 Enter PASS command  
Password:
```

In the following example, user USER01 changes his current password from CJAY to MACDUFF:

```
A:LOG user01 cjay/macduff  
230 User USER01 logged in.
```

MKD

The **MKD** (make directory) command directs a Server FTP to create the specified directory. The exact FTP command sent to the Server FTP program depends on the setting of the **EXPE** variable. For details regarding the **EXPE** variable, see the EXPE.section

MKD [*path_name*]

Syntax Description

path_name Directory to be created.

Default

If you omit *path_name*, you are prompted for it.

Usage Guidelines

If the path name is relative, the specified subdirectory is created in the current working directory.

If the path name is absolute, the specified directory is created.

The syntax for *path_name* depends on the associated Server FTP.

A host prefix is required with the **MKD** command.

Example

For an example of **MKD** operation, consider this case:

The current directory for a session between a UNIX Server FTP and a Client FTP program is */u/user1/work*. If a **MKD junk** command is issued by the Client FTP to that UNIX Server FTP, the subdirectory **junk** is created as a subdirectory in the current directory (*/u/user1/work/junk*).

The same result can be achieved by specifying **MKD /u/user1/work/junk**.

```
B:MKD /u/myid/sampdir
B:257 MKD command successful.
```

MODE

The **MODE** command sets one of three transmission modes:

- Block Mode

Block mode formats the data and allows for restart procedures.

- Compressed Mode

Compressed mode compresses the data for efficient transfer.

- Stream Mode

Stream mode passes the data with little or no processing. It interacts with the structure attribute to determine the type of processing. Stream mode is the default if no **MODE** command was used.

The sending host translates its internal end-of-line or end-of-record representation into the representation required by the transfer mode and file structure; the receiving host performs the inverse translation to its internal representation. Because these transformations make extra work for some systems, identical systems transferring non-record structured text files might use binary representation and stream mode to simplify transfer.

MODE S | B | C

Syntax Description

MODE	Directory to be created.
S	Stream mode.
B	Block mode.
C	Compress mode.

Default

If the prefix is omitted, the **MODE** command is directed to each Server FTP with an open connection.

Usage Guidelines

One of the three codes is required as an argument on the **MODE** command.

A host prefix is optional with the **MODE** command. If a host prefix is specified, the **MODE** command is directed only to that Server FTP

Example

Each of the possible transmission modes is discussed in the following sections. For a detailed description of the effect of various transmission modes, read the “Transmission Modes” section in *RFC 959, File Transfer Protocol*.

Not all Server FTPs support all transmission modes; review the documentation for the target Server FTP if you have questions concerning transmission mode support.

Note Cisco IOS for S/390 Server FTP does not support compressed mode. Client FTP uses the third-party model (server-to-server) and an environment can exist where Cisco IOS for S/390's Server FTP is not implemented. In this instance, compress mode is valid if both remote servers support compress mode.

Block Mode

Set block mode with the character B. In block mode, the file is transmitted as a series of data blocks preceded by one or more header bytes. Record structures are allowed in this mode, and any representation type can be used. Restart markers are embedded in the data stream.

Compressed Mode

Set compressed mode with the character C. In compressed mode, filler bytes (space characters in ASCII or EBCDIC) and replicated data bytes are compressed when transmitted over the data connection. Compressed mode can increase bandwidth on very large network transmissions at little extra CPU cost. Additionally, compressed mode reduces the size of printer files.

Set the structure with the **STRU** command. In compressed mode, the representation type determines the filler byte. Set this with the **TYPE** command.

Stream Mode

Set stream mode with the character S. This is the default if no **MODE** command has been used. In stream mode, the data is transmitted as a stream of bytes. There are no restrictions on the representation type used, and record structures are allowed. In a record structured file, End of Record (EOR) and End of File (EOF) are each indicated by a two-byte control code included with the data sent over the data connection. If the structure is a file structure, the EOF is indicated by the sending host closing the data connection, and all bytes sent over the data connection are data bytes.

NLST

The **NLST** command directs a Server FTP to provide a list of file names in a specified directory or file group. That is, the Server FTP returns a list of file names with no additional information. This contrasts to the **LIST** command, which provides the file names and other current information about the files.

```
NLST [path_name_a] [path_name_b]
```

Syntax Description

path_name_a Path name to be listed from host A.

path_name_b File name at host B into which the name list from host A is copied.

Default

If you omit either path name, you are prompted for it.

Usage Guidelines

If the path name specifies a directory or other group of files, the Server FTP transfers a corresponding list of file names.

If the path name specifies a single file, the Server FTP transfers the file name of that file.

The syntax of each path name depends on the associated Server FTP.

A host prefix is not allowed with the **NLST** command.

Example

```
NLST acces ibmcatent2  
A:125 Transfer Started  
B:150 ASCII data connection for ibmcatent2 (138.42.224.15,4137).  
B:226 Transfer complete.  
A:226-Transfer Complete  
A: 70 bytes sent in 0.54 seconds (129 bytes/s)    Path MYID.ACCESS  
A>User MYID  
A:Data bytes sent 58  
A:226
```

PUT

The **PUT** command requests that the appropriate Server FTP copy a file from host A to a file on host B. The file to be copied is always at host A and the file destination is always at host B. You might need to use the **A=B** command to switch host prefixes and associate the correct host with the correct host prefix.

```
PUT [path_name_a] [path_name_b]
```

Syntax Description

path_name_a Name of the file to be copied from host A.

path_name_b File name at host B into which the file from host A is copied.

Default

If you omit either path name, you are prompted for the file name.

Usage Guidelines

The syntax for each path name depends on the associated Server FTP.

A host prefix is not allowed with the **PUT** or **SEND** commands.

Example

```
PUT n.d put_example  
A:150-Data set open with attributes:  
A:Type A N  Tabs 8  STRU F  Mode S  Path MYID.N.D  
A:Volser COLPK1  Unit SYSALLDA  Dsorg PS  Recfm FB  
A:Lrecl 80  Blksize 3120  Rlse  
A:150  
B:150 ASCII data connection for put_example (138.42.224.15,4138).  
  
A:226-Transfer Complete  
A: 3439 bytes sent in 2.12 seconds (1622 bytes/s)  Path MYID.N.D  
A:User MYID  Data bytes sent 6480  
A:Disk tracks read 1  
A:226  
B:226 ASCII Transfer complete.
```

Related Commands

GET An alternative to switching host prefixes; read GET for details

SEND Can be used in place of **PUT**. There are no differences between **PUT** and **SEND**.

PWD

The **PWD** command directs a Server FTP to return the path name of the current working directory.

PWD

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

A host prefix is required.

Example

```
B: PWD  
B:257 "/u/myid" is current directory.
```

QUIT

The **QUIT** command disconnects you from a host. It logs you out and terminates the connection between you and the Server FTPs. The **BYE** command is a synonym for the **QUIT** command. The **QUIT** command is the opposite of the **CONN** command.

QUIT

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

The **QUIT** command requires a prefix.

Example

```
A: QUIT
A:221 Session Terminated.
```

Related Commands

BYE Can be used instead of **QUIT**. **BYE** works exactly as the **QUIT** command.

END Can be used instead of **QUIT**; **END** does not require a host prefix.

QUOT

The **QUOT** (quote) command sends an uninterpreted, unaltered character string to the Server FTP over the control connection. This mechanism sends FTP commands to the Server that the Client FTP program might not be able to send.

QUOT [*text*]

Syntax Description

text Sent to the Server over the control connection exactly as you enter it.

Default

If the text is omitted, you are prompted to enter it.

Usage Guidelines

A host prefix is required with the **QUOT** command.

Example

```
A:QUOT site norlse
```

REN

The **REN** (rename) command directs a Server FTP to rename a file.

```
REN [old_path_name] [new_path_name]
```

Syntax Description

old_path_name File name to be renamed.

new_path_name New name to be assigned to that file.

Default

If you omit either argument, you are prompted to enter it.

Usage Guidelines

The syntax of the path names depends on the associated Server FTP.

A host prefix is required with the **REN** command.

Example

```
A:REN t2.data rename.data  
A:350 Requested file action pending further information  
A:250 Renamed OK
```

REST

The **REST** (restart) command shows the Server FTP the restart marker where a file transfer is to be restarted.

This command does not cause a file transfer but instead causes the Server FTP to skip over the file to the specified data checkpoint. This command should be followed immediately by the Client FTP command that causes the file transfer to resume.

Note The restart facility requires that you run in **MODE B**. Many UNIX implementations do not support **MODE B** and cannot use the restart facility.

REST *marker*

Syntax Description

marker Server FTP marker point where the file transfer is to be restarted.

Usage Guidelines

The marker is required in the restart command.

The format of the restart marker is determined by the sending Server FTP and should be entered exactly as displayed during the interrupted file transfer.

A host prefix is required with the **REST** command.

Example

The following is a restart marker message received by a user during a file transfer:

```
B:110 MARK 0100220040D82 = 0100220211C3
```

To restart the file transfer at the restart markers, issue the following commands to receive the following output:

```
TYPE i
MODE b
a:rest 010020040d82
b:rest 0100220211c3
a:site rest(100000)
put 'scm.p016572.t01tcp' psr16572.job
A:350 Requested file action pending further information
B:350 Requested file action pending further information
B:150-Data set open with attributes:
B:Type I N   Stru F   Mode B   Recall 5   Path MYID.PSR16572.JOB
B:Volser HAGCAT   Unit SYSALLDA   Dsorg PS   Recfm FB   Lrecl 133
B:Blksize 6650   Space 3 1 Cyl Rlse   Restart at 0100220211C3
B:150
A:Type I N   Stru F   Mode B   Path SCM.P016572.T01TCP
A:Volser HAGCAT   Unit SYSALLDA   Dsorg PS   Recfm FB   Lrecl 133
A:Blksize 6650   Rlse Bytes/Restart 100000   Restart at 010020040D82
A:150

A:226-Transfer Complete
A: 326284 bytes sent in 3.39 seconds (96248 bytes/s)
A:Path SCM.P016572.T01TCP   User MYID   Data bytes sent 318934
A:Disk tracks read 9       Restart markers sent 3
A:226
B:110 MARK 010023010E8C = 010024060D3F
B:110 MARK 010025040F96 = 0100270408BB
B:110 MARK 0100280110A0 = 01002A02043B
B:226-Transfer complete
B: 326284 bytes received in 3.40 seconds (95965 bytes/s)
B:Path MYID.PSR16572.JOB   User MYID   Data bytes received 318934
B:Disk tracks written 9   Records folded 2472
B:Restart markers received 3
B:226
```

RMD

The **RMD** (remove directory) command directs a Server FTP to remove the specified directory.

RMD [*path_name*]

Syntax Description

path_name Directory to be removed.

Default

If you omit *path_name*, you are prompted for it.

Usage Guidelines

If the path name is relative, the specified subdirectory is removed from the current working directory. If the path name is absolute, the directory is removed.

The syntax of *path_name* depends on the associated Server FTP.

A host prefix is required with the **RMD** command. Many systems require the directory to be empty before it can be removed.

Example

A UNIX Server FTP in session with the Client FTP program has /u/user1/work as the current directory. If an **RMD junk** command is issued by the Client FTP to that UNIX Server FTP, the junk subdirectory (/u/user1/work/junk) of the current directory is removed. The same result is achieved by specifying **RMD /u/user1/work/junk**.

```
B:RMD /u/myid/sampdir
B:250 RMD command successful.
```

SEND

The **SEND** command is the same as the **PUT** command.

SEND [*path_name_a*] [*path_name_b*]

Syntax Description

path_name_a File name of the file to be copied from host A.

path_name_b File name at host B into which the file from host A is copied.

Default

If you omit either path name, you are prompted for the file name.

Usage Guidelines

The syntax for each path name depends on the associated Server FTP.

A host prefix is not allowed with the **PUT** or **SEND** commands.

Example

```
SEND n.d send_example
A:150-Data set open with attributes:
A:Type A N  Tabs 8   Stru F   Mode S   Path MYID.N.D
A:Volser COLPK1  Unit SYSALLDA  Dsorg PS   Recfm FB   Lrecl  80
A:BLKSIZE 3120   Rlse
A:150
B:150 ASCII data connection for send_example (138.42.224.15,4148).
B:226 ASCII Transfer complete.
A:226-Transfer Complete
A: 3439 bytes sent in 4.66 seconds (737 bytes/s)   Path MYID.N.D
A>User MYID   Data bytes sent 6480
A:Disk tracks read 1
A:226
```

Related Commands

GET An alternative to switching host prefixes; read **GET**

PUT Can be used in place of **SEND**. There are no differences between **SEND** and **PUT**.

SITE

The **SITE** (site parameters) command provides the Server FTP with specific information it requires. This information is essential to file transfers involving that Server FTP, but is not sufficiently universal to have been included specifically in the FTP. Typically, you use a **HELP SITE** Client FTP command to find the **SITE** requirements for a specific Server FTP. Otherwise, review the Server FTP documentation for the **SITE** requirements.

SITE *text*

Syntax Description

text Passed through unchanged to the specified server.

Usage Guidelines

Text is required in the **SITE** command syntax.

A host prefix is required with the **SITE** command.

Detailed documentation of this command is in the SITE section in Server FTP.

Example

```
A:SITE vol(mvsts2)
```

SNDS

The **SNDS** (re-send **SITE** parameters) command directs the Client FTP program to re-send the last **SITE** command to the specified Server FTP. Your **SITE** command is reissued without your having to retype it. Since most Server FTPs require that new site parameters be provided before each data transfer, the **SNDS** saves time if identical site parameters are to be used repeatedly.

SNDS

Syntax Description

This command has no arguments or keywords.

Usage Guidelines

A host prefix is required with the **SNDS** command.

Example

```
A: SNDS  
site vol(mvsts2) <SENT
```

STAT

The **STAT** (status) command requests a status response from a specified server.

STAT [*path_name*]

Syntax Description

path_name Optional; specifies path to server.

Default

If no path name is given, the indicated Server FTP sends status information relative to parameters and connection status.

Usage Guidelines

When **STAT** is issued between data transfer operations, the path name argument can be given. When it is given, the command works the same as the **LIST** command: It displays current information about the referenced files.

When using the **STAT** command, the information is transferred over the control connection instead of the data connection.

The syntax of *path_name* depends on the associated Server FTP(s).

A host prefix is optional with the **STAT** command

If a host prefix is specified, the **STAT** command is sent only to the specified host.

If the prefix is omitted, the **STAT** command is sent to both hosts that have open connections.

The Cisco IOS for S/390 Server FTP program implements some additional parameters on the **STAT** command. Use a **HELP STAT** Client FTP command to find additional parameters.

Example

In the following example, the status for the A host is requested.

```
A:STAT *
A:211--- STATUS ---
A:  -- FTP Parameters --
A:Remote DT Host, Port 138.42.32.160, 0
A:Local DT Host, Port 138.42.224.15, 0
A:Type A N  Tabs 8  Stru F  Mode S  Recall 5  Server is passive
A:  -- END --
A:  -- Control --
A:User MYID  Acct Accs E0000200  Unit SYSALLDA  Host 138.42.224.15
A:  -- End Control --
A:  -- Path Data --
A:Rlse
A:  -- End Path Data --
A:  -- Transfer Information --
A:Data Transfer not in progress  Data bytes sent 6480
A:Disk tracks read 1  Network bytes sent 3439  Elapsed time 00.00.04
A:Bytes/Second 737
A:  -- END --
A:211 <End of Status>.
```

STRU

The **STRU** (file structure) command provides information on file structure to a Server FTP.

STRU F | R

Syntax Description

- | | |
|----------|---|
| F | Specifies the file structure. File structure is used for files with no internal structure, and the file is considered to be a contiguous sequence of data bytes. |
| R | Specifies the record structure. This is for files made up of sequential records. Record structure is accepted for text files (such as, files with type ASCII or EBCDIC) by all FTP implementations. |

Defaults

If no **STRU** command has been used, the default file structure is **F**.

If no host prefix is specified, the command goes to both Server FTPs.

Usage Guidelines

One argument is required on the **STRU** command to set the file structure.

A host prefix is optional with the **STRU** command. When a host prefix is given, the **STRU** command goes only to the specified Server FTP.

TYPE

The **TYPE** command tells a Server FTP the data type to use.

```
TYPE I | L byte_size | {A | E [N | T | C]}
```

Syntax Description

- I** Image type. The data is sent as a contiguous bit stream that, for transfer, is packed into 8-bit transfer bytes. The receiving site stores the data as contiguous bits.
- The receiving storage system might need to pad the file (or each record, in record-structured files) to some convenient boundary. Review the documentation for a Server FTP to find out about padding.
- Image type is for the efficient storage and retrieval of files and for transfer of binary data. All FTP implementations are required to support the image type.
- L *byte_size*** Local file type and the logical byte size of the file. The byte size value (*byte_size*), representing the logical byte size, is required with the local type. With this type, the data is transferred in logical bytes of the specified size. The logical byte size might differ from the transfer byte size. If the logical and transfer byte sizes differ, the logical bytes are packed contiguously disregarding transfer byte boundaries and are padded at the end if necessary.
- When the data reaches the receiving host, it is transformed in a manner dependent on the logical byte size and the particular host. The transformation is invertible; an identical file can be retrieved if the same parameters are used.
- A** Sets the file type to ASCII. This type is accepted by all FTP implementations and is good for transferring text files, except when both hosts find the EBCDIC type more convenient. In accordance with the *NVT* standard, the **CRLF** sequence is used at the end of a line of text.
- The sender converts the data from an internal character representation to the standard 8-bit NVT ASCII representation (see the TELNET specification in the list of reference documents). The receiver converts the data from this standard form to the receiver's own internal form.
- E** Sets the file type to EBCDIC, which performs efficient transfer between hosts that use EBCDIC. Cisco IOS for S/390 Client FTP2 users usually use this type when copying files to their MVS host.
- Data is transmitted as 8-bit EBCDIC characters. The character code is the only difference between EBCDIC and ASCII types.
- End-of-line is rarely used with EBCDIC type to denote structure, but where it is necessary, the **NL** character is used.
- The types ASCII and EBCDIC optionally take a second parameter that indicates what kind of vertical format control, if any, is associated with a file. If a file is to be sent to a host for printing, the vertical format control must be defined in the format expected by the target. The ASCII and EBCDIC types have a second parameter specifying non-print, TELNET, or carriage control (ASA).
- N** **N** sets non-print format control. This is used when the file does not contain vertical format information. Normally, this format is used with files destined for processing or for storage. Non-print format is accepted by all FTP implementations.

T sets TELNET format control. This is used when the file contains ASCII/EBCDIC vertical format controls (in other words, **CR**, **LF**, **NL**, **VT**, **FF**). The characters **CRLF**, in exactly this sequence, also denote end-of-line.

C sets carriage control (ASA) format control. This is used when the file contains ASA (FORTRAN) vertical format control characters.

ASA standard specifies these control characters:

blank Move paper up one line.
0 Move paper up two lines.
- Move paper up three lines.
1 Move paper to top of next page.
+ No movement (that is, type will overprint)

Default

For both ASCII and EBCDIC file types, vertical format control **N** is the default.

ASCII is the default argument for the **TYPE** command.

Non-print format is the default.

Usage Guidelines

One of the four arguments (**I**, **L** *byte_size*, **A**, or **E**) is required.

If local type (**L**) is set, the integer byte size argument must also be set.

If ASCII (**A**) or EBCDIC (**E**) type is set, one of the three vertical format control arguments, **N**, **T**, or **C**, also can be set.

A=B

The **A=B** (switch prefixes) command directs the Client FTP program to switch the host prefixes associated with each Server FTP connection. This command gets the correct host associated with the correct host prefix before issuing a data transfer command such as **ADD**, **NLST**, or **LIST**.

A=B

Syntax Description

This command has no arguments or keywords.

Default

The **A=B** command is directed to the Client FTP program and any host prefix is ignored.

Example

The A host is an MVS system and the B host is a UNIX system. You want to obtain a name list of the current directory on the UNIX system. If you issue the Client FTP **NLST** command to obtain a name list, the **NLST** command obtains the name list from the A host and copies it to a file on the B host. Since the UNIX host is the B host, issue the **A=B** command to make the UNIX host the A host before issuing the **NLST** command.

A?B

The **A?B** (show connections) command displays the current connections associated with the **A** and **B** host prefixes from the Client FTP program.

A?B

Syntax Description

This command has no arguments or keywords. The following output is produced by the **A?B** command:

```
HOST - A:host_name_a
HOST - B:host_name_b
```

Usage Guidelines

The host names displayed are the host names in the **CONN** commands. If a connection is not established for a host prefix, the response is **NOT CONNECTED**.

The **A?B** command goes to the Client FTP program.

A host prefix is not necessary with the **A?B** command and is ignored.

Example

```
A?B
HOST - A:MVS
HOST - B:UNIX
```

Restart Support

If a file transfer is interrupted, it can be restarted. However, restart support requires that **MODE B** be specified.

The restart marker provided by Cisco IOS for S/390 is six bytes long in the format *VTRBB*. *V* is the volume sequence number, **TTR** is the standard IBM OS disk block address, and *BB* is a byte offset within the block.

How to Restart

Use the **SITE REST(XXXXX)** FTP command to tell a Cisco IOS for S/390 FTP Server how often to send a restart marker. Send a restart marker after the sending of the record that exceeds or equals *XXXXX* number of bytes (varying between 1 and 500,000). Send the **SITE** command only to the **RETR** side of a data transfer. Other FTP Servers may initiate sending restart markers in a different way from Cisco IOS for S/390.

A 110 message is written once per output block if a restart marker is sent somewhere in the data written for the block.

For example, suppose during a data transfer this restart mark message is sent:

```
B:110 MARK 0100030212C0 = 010003020FA0
```

You can restart the transfer at this point by sending the first number to the **RETR** side and the second number to the **STOR**. If you receive this restart mark message during an aborted data transfer, you can restart the transfer at these disk locations with these Client FTP **REST** commands:

```
to the RETR side: REST 0100030212C0
to the STOR side: REST 010003020FA0
```

Cisco IOS for S/390 supports restart markers (set at default value of every 32767 data bytes) if these conditions exist:

- **TYPE I**
- **MODE B**

Client FTP File Transfer Examples

This section provides some examples of file transfers. These examples illustrate use of the more frequently used commands and FTP features. They include:

- PUT Example
- GET Example
- Transfer to an MVS Internal Reader
- Restart File Transfer Example
- Managing Directories on UNIX-based Systems

These examples follow the conventions described here.

FTP Invocation and Conventions

- Issue FTP under TSO to enter Client FTP.
- When Client FTP is ready for command input from a user, it places the FTP: prompt on the left side of the screen.

Host Prefixes

- Client FTP is based on the three-party model. The FTP protocols require that a user connect to a host before most commands can be issued. By convention, the first host the user connects to is the A side. Also by convention, the second host connected to is the B side.
- Commands that are specific to one host must be prefixed with the side identification (A or B) followed by a colon. Thus, to send a **SITE VOL** (MVSTSO) command to the A side, enter this command:

```
A:SITE VOL (MVSTSO)
```

- All command output sent from a host is prefixed by the host site ID and a colon. All the A side host output is prefixed by A:.

Successful Completion of a Transfer

- If the only response from the Client FTP command is **USERFTP:**, the command completed successfully.

Entering Text

- Text can be entered in uppercase or lowercase. Some host systems allow a mixture of lowercase and uppercase letters, while other host systems use uppercase for most functions. All commands entered are translated to uppercase before being sent to the servers. The data associated with a command is sent to its appropriate FTP server without case translation. The Cisco IOS for S/390 FTP server translates user IDs, passwords, data set names, and similar items to uppercase before the commands associated with them are executed.

Readability

- To improve readability, blank lines have been inserted between commands in the examples in this section. These lines do not appear in a real FTP terminal session.

PUT Example

In this FTP session, the **PUT** command transfers a file from host MVS host UNIX.

```

Cisco IOS for S/390 R/ Client FTP - Enter command or '?'
conn mvs
A:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter command or HELP
a:log myid
331 Enter PASS command
Password:
conn unix
B:220 unix FTP Server (SunOS 4.1) ready.
b:log myid
331 Enter PASS command
Password:
put cntl(iefbr14) jclbr14
A:150-Data set open with attributes:
A:Type A N   Tabs 8   Stru F   Mode S   Path MYID.CNTL(IEFBR14)
A:Volser COLPK1   Unit SYSALLDA   Dsorg PO   Recfm FB   Lrecl 80
A:Blksize 3120   Rlse
A:150
B:150 ASCII data connection for jclbr14 (138.42.224.15,4151).
A:226-Transfer complete
A: 820 bytes sent in 0.50 seconds (1640 bytes/s) Path MYID.CNTL(IEFBR14)
A>User MYID   Data bytes sent 800
A:Disk tracks read 1
A:226
B:226 ASCII Transfer Complete
end

```

- The **conn mvs** command connects to host MVS. The first connection is to the A side. The response “A:220” on the next line shows that the connect command succeeded.
- The command **a:log myid** logs a user on to the A side. myid is the user ID. FTP prompts for a password, then sends the user ID and password combination to the A side host for validation. No error message is received and the USERFTP: prompt displays on the next line, indicating that the log command worked and user myid is logged on to the A host, MVS.
- The **conn unix** command connects to the B side host, unix.
- The command **b:log myid** logs a user onto the B side. FTP prompts for a password in a nondisplay field. FTP sends the userid/password combination to host unix. These are valid and user myid logs on to the B side.
- The command **put cntl(iefbr14) jclbr14** tells Client FTP to transfer file MYID.CNTL(IEFBR14) from the A side host, MVS. (FTP prefixes the user ID to the beginning of any non-quoted data set name referenced while a user is connected to an MVS host with Cisco IOS for S/390 running.) FTP creates or overwrites file jclbr14 on the side B host, unix.

The A:226 message from the A side host means that all of file MYID.CNTL(IEFBR14) successfully transferred from the A side host, MVS. The B:226 messages mean that FTP successfully created or overwrote file jclbr14 on the B side host, unix.

- The **end** command ends the FTP session.

GET Example

In the following FTP session, the **GET** command transfers a file from host unix to host MVS.

```
Cisco IOS for S/390 Rn Client FTP - Enter command or '?'
conn mvs
A:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter Command or HELP
a:log myid
331 Enter PASS command
Password:
conn unix
B:220 unix FTP server (SunOS 4.1) ready.
b:log myid
331 Enter PASS command
Password:
get jclbr14 cntl(newbr14)
A:150-Data set open with attributes:
A:Type A N   Tabs 8   Stru F   Mode S   Recall 5
A:Path MYID.CNTL(NEWBR14)
A:Volser COLPK1   Unit SYSALLDA   Dsorg PO   Recfm FB   Lrecl 80
A:Blksize 3120   Space 31 15 Tracks Rlse
A:150
B:150 ASCII data connection for jclbr14 (138.42.224.15,4098) (810 bytes).
A:226-Transfer Complete
A: 820 bytes received in 3.36 seconds (244 bytes/s)
A:Path MYID.CNTL(NEWBR14)   User MYID   Data bytes received 800
A:Disk tracks written 1
A:226
B:226 ASCII Transfer complete.
end
```

- The command **conn mvs** connects to host MVS. The first connection is to side A. The response A:220 on the next line shows that the connect command succeeded.
- The command **a:log myid** logs a user onto the A side. myid is the user ID. FTP prompts for a password, then sends the user ID and password combination to the A side host for validation. No error message is received and the USERFTP: prompt displays on the next line, indicating that the log command worked and user myid is logged on to the A side host, MVS.
- The command **conn unix** connects to the B side host, unix.
- The command **b:log myid** logs a user on to the B side. FTP prompts for a password in a nondisplay field. FTP sends the userid/password combination to host unix. These are valid and user myid logs on to the B side.
- The command **get jclbr14 cntl(newbr14)** tells Client FTP to transfer file jclbr14 from the B side host, unix. FTP creates or overwrites file MYID.CNTL(NEWBR14) on the A side host, MVS. (FTP prefixes the user ID to the beginning of any non-quoted data set name referenced when a user is connected to an MVS host with Cisco IOS for S/390 running.)

The B:226 message from the B side host means that all of file jclbr14 successfully transferred from the B side host, unix. The A:226 message means that FTP successfully created or overwrote file MYID.CNTL(NEWBR14) on the A side host, MVS.
- The **end** command ends the FTP session.

Transfer to an MVS Internal Reader

The following is an example of an FTP file transfer from a data set on host **unix** to an MVS internal reader on host **MVS**.

```
Cisco IOS for S/390 R0 Client FTP - Enter command or '?'
conn unix
A:220 unix FTP server (SunOS 4.1) ready.
a:log myid
331 Enter PASS command
Password:
conn mvs
B:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter command or HELP
b:log myid
331 Enter PASS command
Password:
b:site submit
put jclbr14 anyname.data
B:150-Data set open with attributes:
B:Type A N  Tabs 8   Stru F   Mode S   Intrdr   Recfm FB   Lrecl 80
B:Blksize 20000
B:150
A:150 ASCII data connection for jclbr14 (138.42.224.15,20) (810 bytes).
A:226 ASCII Transfer complete.
B:226-Transfer complete
B: 820 bytes received in 0.30 seconds (2733 bytes/s)   User MYID
B>Data bytes received 800
B:226
end
```

- The command **conn unix** connects to host unix. The first connection is to the A side. The A:220 message on the next line shows that the connect succeeded.
- The command **a:log myid** logs a user on to the A side. The myid is the user ID. FTP prompts for a password. FTP sends the user ID and password combination to the A side host for validation. No error message is received and the USERFTP: prompt displays on the next line, indicating the log command succeeded. User myid is logged on to the A side host, unix.
- The command **conn mvs** connects to the B side host, MVS.
- The command **b:log myid** logs a user on to the B side. FTP prompts for a password in a nondisplay field. FTP sends the userid/password combination to host MVS. These are valid and user MYID logs on to the B side.
- The command **b:site submit** is to the B side host, MVS, directing the next data transfer to the MVS internal reader for execution. The **SITE** command is relevant to an MVS host where Cisco IOS for S/390 is running.
- The **put jclbr14 anyname.data** commands FTP to transfer file jclbr14 from the A side host, unix. Due to the previous command in step 5, the file is transferred to an MVS internal reader on side B, host MVS. The file name (*anyname.data*) for the B side host is ignored because no data set is being created or updated.
- The **end** command ends the FTP session.

Restart File Transfer Example

This is an example of a restart file transfer.

```
Cisco IOS for S/390 R/ Client FTP - Enter command or '?'
conn mvs
A:220 MVS.HQ.COMPANY.COM -- FTP Server, Enter Command or HELP
conn mvs2
B:220 MVS2.HQ.COMPANY.COM -- FTP Server, Enter Command or HELP
a:log myid
331 Enter PASS command
Password:
b:log myid
331 Enter PASS command
Password:
type i
mode b
a:site rest(100000)
b:site lrecl(133) recfm(vb) blksize(6650) space(3 1) CYL
put 'scm.p016572.t01tcp' psr16572.t01tcp
A:150-Data set open with attributes:
A:Type I N   STru F   Mode B   Path SCM.P016572.T01TCP
A:Volser HAGCAT   Unit SYSALLDA   Dsorg PS   Recfm FB   Lrecl 133
A:Blksize 6650   Rlse   Bytes/Restart 100000
A:150
B:150-Data set open with attributes:
B:Type I N   Stru F   Mode B   Recall 5   Path MYID.PSR16572.T01TCP
B:Volser ICSFK3   Unit SYSALLDA   Dsorg PS   Recfm VB   Lrecl 133
B:Blksize 6650   Space 3 1 Cyl Rlse
B:150
B:110 MARK 000204010A = 00020414F9
B:110 MARK 0005010214 = 0005021075
B:110 MARK 000704031E = 0007060BF1
B:110 MARK 000A010428 = 000A040771
B:110 MARK 000C040532 = 000D0202ED
B:110 MARK 000F01063C = 000F0517DE
B:110 MARK 0011040746 = 001203135E
B:110 MARK 0014010850 = 0015010EDA
B:110 MARK 001604095A = 0017050A56
B:110 MARK 0019010A64 = 001A0305D6
B:110 MARK 001B040B6E = 001D010152
B:110 MARK 001E010C78 = 001F041643
B:110 MARK 0020040D82 = 00220211C3
B:110 MARK 0023010E8C = 0024060D3F
B:110 MARK 0025040F96 = 00270408BB
B:110 MARK 00280110A0 = 002A02043B
A:226-Transfer Complete
A: 1656466 bytes sent in 52.1 seconds (31781 bytes/s)
A:Path SCM.P016572.T01TCP   User MYID   Data bytes sent 1619142
A:Dist tracks read 41   Restart markers sent 16
A:226
B:226-Transfer complete
B: 1656466 bytes received in 52.1 seconds (31775 bytes/s)
B:Path MYID.PSR16572.T01TCP   User MYID   Data bytes received 1619142
B:Disk tracks written 43   Records folded 12551
B:Restart markers received 16
B:226
end
```

- The two **conn mvs** commands connect to host MVS on the A side, and to host MVS2 on the B side.
- The **a:log myid** command logs user MYID onto the A side, and user MYID onto the B side.
- The **type i** command sets image type to a binary transfer.

- The **mode b** command sets block mode. Restart markers are embedded in the data stream.
- The **a:site rest(100000)** command tells the sending side how often to insert restart markers into the data.
- The **b:site lrecl(133) recfm(vb) blksize(6650) space(3 1) CYL** command changes some of the default allocation parameters on the B side.bu
- The **put** command tells FTP to transfer a file from the A side host, MVS. FTP creates file MYID.PSR16572.T01TCP on the side B host, MVS2.
- The A:150 messages give details of the files being transferred.
- The B:110 MARK messages mark blocks sent with embedded restart markers. The first number of the 110 message is the restart marker for the RETR side (the A side in this example). The second number of the 110 message is a restart marker for the STOR side (the B side in this example).
- The A:226 messages from the A side host indicate that all of file SCM.P016572.T01TCP successfully transferred from the A side host, MVS. The B:226 messages indicate that FTP successfully created file MYID.PSR16572.T01TCP on the B side host, MVS2.
- The **end** command ends the FTP session.

Managing Directories on UNIX-based Systems

These commands show how to manage directories on UNIX systems.

```
Cisco IOS for S/390 RnClient FTP - Enter command or '?'  
conn unix  
A:220 unix FTPserver (SunOS 4.1) ready.  
a:log demo ftptest  
a:pwd  
A:257 "/u/demo" is current directory.  
a:mkd tempdir  
A:257 MKD command successful.  
a:pwd  
A:257 "/u/demo" is current directory.  
a:cwd tempdir  
A:250 CWD command successful.  
a:pwd A:257 "/u/demo/tempdir" is current directory.  
a:cdup  
A:250 CWD command successful.  
a:pwd  
A:257 "/u/demo" is current directory.  
a:rmd tempdir  
A:250 RMD command successful.  
a:pwd  
A:257 "/u/demo" is current directory.  
end
```

- The command **conn unix** connects the A side to host unix.
- The command **a:log demo ftptest** logs User demo on to the A side host unix.
- The command **a:pwd** asks the server to print the A side path name of its current directory.
- The command **a:mkd tempdir** asks the A side server to create a directory called tempdir.
- The command **a:cwd tempdir** changes the A side directory from /u/demo to /u/demo/tempdir.
- The command **a:cdup** changes the A side directory from its current directory /u/demo/tempdir to its parent directory u/demo.

-
- The command **a:rmd tempdir** asks the A side server to remove directory /u/demo/tempdir.
 - The **end** command terminates the FTP session.

