

# Client FTP3

---

This chapter describes Client FTP3, the File Transfer Protocol (FTP) that allows file transfers among unlike hosts in diverse internetworking environments. It contains these sections:

- **Introducing Client FTP3**  
Provides a brief overview of the Cisco IOS for S/390 File Transfer Protocol implementation.
- **Client FTP3**  
Describes the FTP3 data flow and includes an illustration showing how Client FTP3 works.
- **Invoking Client FTP3**  
Describes how to use Client FTP3 as both a TSO command and as a regular batch program.
- **General Client FTP3 Operation**  
Describes the general operation of the Client FTP3 program.
- **Client FTP3 Commands**  
Includes a table listing all of the Client FTP3 commands and provides a brief description of each.  
Commands are:

? Command	ACCOUNT	APPEND	ASCII	BINARY
CD	CDUP	CLOSE	DEBUG	DELETE
DELIMIT	DIR	EBCDIC	GET	HELP
LCD	LMKDIR	LOCSITE	LOCSTAT	LPWD
LS	MDELETE	MGET	MKDIR	MODE
MPUT	NOOP	OPEN	PASS	PUT
PWD	QUIT	QUOTE	RENAME	RESTART
RMDIR	SENDSITE	SITE	STATUS	STRUCT
SYSTEM	TRACE	TSO	TYPE	USER

## Introducing Client FTP3

Like Client FTP and Client FTP2, Client FTP3 is a three-party model FTP client. Two control connections are established and maintained by the client. Like Client FTP2, Client FTP3 connects automatically to the local Cisco IOS for S/390 FTP server and signs on the user.

Despite these basic similarities, Client FTP3 differs from both Client FTP and Client FTP2 in several significant ways. The differences are described in this chapter.

## PL/I Runtime Libraries

Client FTP3 is written in C, and so does not require PL/I runtime libraries, as do the other clients. It does require the SAS/C runtime libraries.

## Connections to the FTP Servers

Client FTP3 is a true client application. Both Client FTP and Client FTP2 communicate with a client *agent* through a VTAM LU0 connection in the Cisco IOS for S/390 server address space. Client FTP3, on the other hand, uses the Cisco IOS for S/390 C/Sockets library to establish direct socket connections with both the local Cisco IOS for S/390 FTP server (local server) and the remote FTP server. This eliminates the need for VTAM LU resources, and improves response time.

## Throughput and CPU Utilization

Although primarily a three-party client, Client FTP3 performs some operations in two-party mode to take advantage of the high throughput and low CPU utilization of the Cisco IOS for S/390 FTP server. The result is improved response to the user and quick response time of a direct connection to the remote server.

For file-transfer operations, such as the client commands **GET**, **PUT**, and **APPEND**, Client FTP3 works in three-party mode. For directory commands, such as the client commands **LS** and **DIR**, as well as the implied directory commands in the **MGET**, **MPUT**, and **MDELETE** commands, Client FTP3 operates in two-party mode. Unlike Client FTP2, Client FTP3 does not need to be run under the TMP to support **LS** and **DIR** in batch jobs.

## Client FTP3 Data Transfer

Client FTP3 appears to the user as a two-party model by suppressing almost all local server replies, simplifying the client responses considerably. Also, Client FTP3 operates in *blocked* mode. When a data transfer is initiated, the terminal remains blocked until the transfer completes.

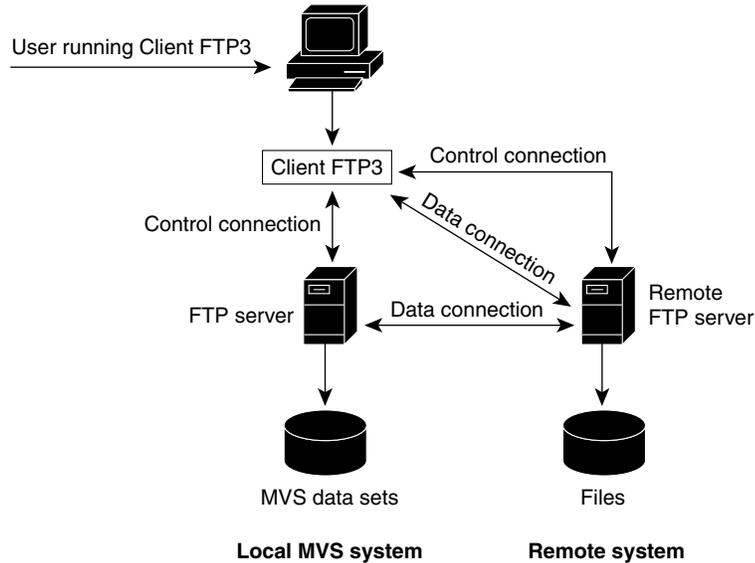
By comparison, Client FTP2 returns a command prompt and allows the user to enter commands as soon as a data transfer is initiated between the local and remote servers. This allows you to enter the **STAT** command to check on progress, or the **ABORT** command to abort the transfer, but it also means that user running interactively must press **ENTER** to be notified of the end of the transfer. Client FTP3, on the other hand, does not allow terminal input during a data transfer. For long-running transfers, a statistics message is written to the terminal every 10 seconds. You can abort the transfer by pressing the attention key on the terminal keyboard. When the transfer ends, Client FTP3 displays the reply, and returns to the command prompt.

---

## Client FTP3

Client FTP3 Program in the Three-Party Model shows the relationship between the Client FTP3 program and the two Server FTP programs in the three-party model.

**Figure 3-1 Client FTP3 Program in the Three-Party Model**



Note that two Data Connections are shown. The data connection between the Remote Server FTP and the Local Server FTP is used for the **GET**, **PUT**, and **APPEND** client commands (**RETR**, **STOR**, **APPE** and **STOU** server operations). The data connection between the Client FTP3 and the Remote Server FTP is used for the client **LS** and **DIR** commands and the implied **LS** in the **MGET**, **MPUT**, and **MDELETE** commands (**LIST** and **NLST** server operations).

## Invoking Client FTP3

The Client FTP3 program runs as a TSO command and can be called as a regular batch program with MVS JCL.

## Invoking Client FTP3 through TSO

In a TSO environment, Client FTP3 can be accessed as a TSO command or it can be called as a program with the TSO **CALL** command. Because Client FTP3 does not use full-screen facilities, it can be used from any type of terminal supported by TSO, including 3270 systems, 3767 systems, and asynchronous ASCII terminals supported by NTO or NPSI.

---

**Note** You must have **PROMPT** set in your TSO profile for Client FTP3 to work properly in interactive mode.

---

## FTP3 TSO Command

Some keywords used in the FTP3 command can be used to override values specified in the TCPIP.DATA data set. Read Understanding the Configuration Data Sets for more information.

Invoke Client FTP3 by entering the FTP3 TSO command in this format.

**FTP3** *remote\_host* [*port\_number*] [*options*]

### Syntax Description

<b>FTP3</b>	Invokes the Client FTP3 program.
<i>remote_host</i>	Name of the remote host. Client FTP3 will automatically connect to this host at initialization. This parameter is required. If you do not supply this parameter, you will be prompted for it.
<i>port_number</i>	Port number of the FTP server on the remote host. Default: 21
<b>BLOCK   NOBLOCK</b>	Specifies whether the ftp client will permit user input while a file transfer is in progress. Default: BLOCK
<b>DEBUG</b>	Toggle used to activate or deactivate the debugging option. Use the DEBUG or TRACE options interchangeably. You may include either DEBUG or TRACE on the command line, but not both; the second option cancels the first.
<b>EXIT   EXIT=<i>nn</i></b>	Specifies that the client is to terminate in case of an error if the exit_if_error flag is true. Exit= <i>nn</i> provides a return code for error conditions.
<b>LOCALHOST</b> <i>host_name</i>	Specifies the host name of the local FTP server host. <b>Override note:</b> This option will override the HOSTNAME statement in the TCPIP.DATA data set.
<b>SSID</b> <i>subsystem_id</i>	Specifies the subsystem id for the Cisco IOS for S/390 API. Default: ACSS
<b>TCP</b> <i>tcPIP</i>	Specifies the job name of the Cisco IOS for S/390 sockets API address space. If no job name is specified, FTP3 uses the subsystem ID specified in the SSID parameter of the FTP statement. If SSID does not indicate a subsystem ID, ACSS is used. <b>Override note:</b> This option will override the TCPIPJOBNAME statement in the TCPIP.DATA data set.
<b>TIMEOUT</b> <i>nm</i>	Sets the following timeout parameters MyopenTime DconnTime CconnTime InactTime DataCtTime  See the discussion Understanding the Configuration Data Sets for the meaning of these timers.

---

<b>TRACE</b>	Toggle used to activate or deactivate the debugging option. Use the DEBUG or TRACE options interchangeably. You may include either DEBUG or TRACE on the command line, but not both; the second option cancels the first.
<b>TRANSLATE</b> <i>data_set_name</i>	Specifies the name of a nonstandard translate table. If this parameter is not supplied, FTP will use the translate table in <i>hlq.STANDARD.TCPXLBIN</i> (read TCPIP.DATA for an explanation of the <i>hlq</i> ). If present, this parameter will be used to construct a data set name in the form <i>user_id.data_set_name.TCPXLBIN</i> . If this data set does not exist, FTP will attempt to allocate <i>hlq.data_set_name.TCPXLBIN</i> .
<b>VERBOSE</b>	Specifies VERBOSE mode. All commands to and replies from the local host will be echoed to the user.  <b>Note</b> A left parenthesis “(“ separates the <i>remote_host</i> and <i>port_number</i> from the other options.  <b>Example</b> <code>ftp unix.company.com (translate standard</code>

## TSO CALL Command

Use the TSO **CALL** command in a TSO environment to call and execute the FTP3 program out of a specific library. This is especially useful at sites that run multiple releases of the product or have test and production versions of the product at different maintenance levels.

**CALL 'T01TCP.FTPLOAD(FTP3)' [*remote\_host port options*']**

### Syntax Description

**'T01TCP.FTPLOAD(FTP3)** Library from which FTP3 will be called .

*'options'* Any number of FTP3 invocation options can be included in the **CALL** command. Read Invoking Client FTP3 through TSO for a complete list.

### Usage Guidelines

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

When options are specified in the command statement, they must be enclosed in single quotes.

When invoked by the **CALL** command, Client FTP3 runs as a program and not as a TSO command.

You can use the NETRC file with the TSO FTP3 call. Read The NETRC File for more information.

## 'Batch Invocation

The Client FTP3 program can be run in batch as either a program like any other, or as a TSO command by running it under a batch Terminal Monitor Program (TMP).

You can specify a NETRC file in batch mode. Specify a NETRC DD file with the name of your NETRC file.

### Batch Program

You can invoke Client FTP3 in batch in a manner similar to any other batch utility program. A sample JCL file is contained in the SAMP data set as T00PFBJB

```
//<jobname> JOB job_stmt_parms
//FTPSTEP EXEC PGM=FTP3,REGION=1024K,
//          PARM='remote_host port options'
//STEPLIB DD DSN=T01TCP.FTPLOAD,DISP=SHR
//SYSTCPD DD DISP=SHR,DSN=userid.TCPIP.DATA
//SYSFTPD DD DISP=SHR,DSN=userid.FTP.DATA
//SYSTEM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
unix
user pass
stat
quit
//
```

Optionally, you may include a NETRC file, as shown here:

```
//NETRC DD DISP=SHR,DSN=userid.NETRC
```

### Batch TMP

An example of this JCL is located in the SAMP data set as member T00PFTMP.

```
//<jobname> JOB job_stmt_parms
//*
//* RUN FTP3 UNDER BATCH TSO
//*
//FTP3 EXEC PGM=IKJEFT01,REGION=4096K,
//STEPLIB DD DISP=SHR,DSN=T01TCP.FTPLOAD
// DD DISP=SHR,DSN=T01TCP.LOAD
//SYSTSPRT DD SYSOUT=*
//OUTPUT DD SYSOUT=*,DCB=BLKSIZE=133
//SYSTCPD DD DISP=SHR,DSN=userid.TCPIP.DATA
//SYSFTPD DD DISP=SHR,DSN=userid.FTP.DATA
//SYSTEM DD SYSOUT=*
//SYSTEMSIN DD *
FTP3 remote_host port options
//INPUT DD *
unix
user pass
stat
quit
//
```

## Understanding the Configuration Data Sets

If you have previously installed the IBM TCP/IP for MVS, the configuration files described here might already exist. Cisco IOS for S/390 provides support for these data sets to allow former IBM customers to run their applications using Cisco IOS for S/390.

---

Client FTP3 automatically searches for and dynamically allocates the configuration data sets when you start FTP3. You can define user-specific environment settings for FTP3 by setting parameters in these configuration data sets:

- *hlq.TCPIP.DATA*

TCPIP.DATA defines the Cisco IOS for S/390 system environment on the local host. You can set your own values by creating a *userid.TCPIP.DATA* data set.

- *hlq.FTP.DATA*

FTP.DATA defines the local **SITE** parameters that are sent to the local host in the form of **SITE** commands. You can set your own values by creating a *userid.FTP.DATA* data set.

The high-level qualifier is specified in T00PFUM. See Changing the High-Level Qualifier for instructions on changing the default HLQ.

Samples of these files are located in the SAMP data set. The member names are T00FTPDS (FTP.DATA) and T00TCPDS (TCPIP.DATA).



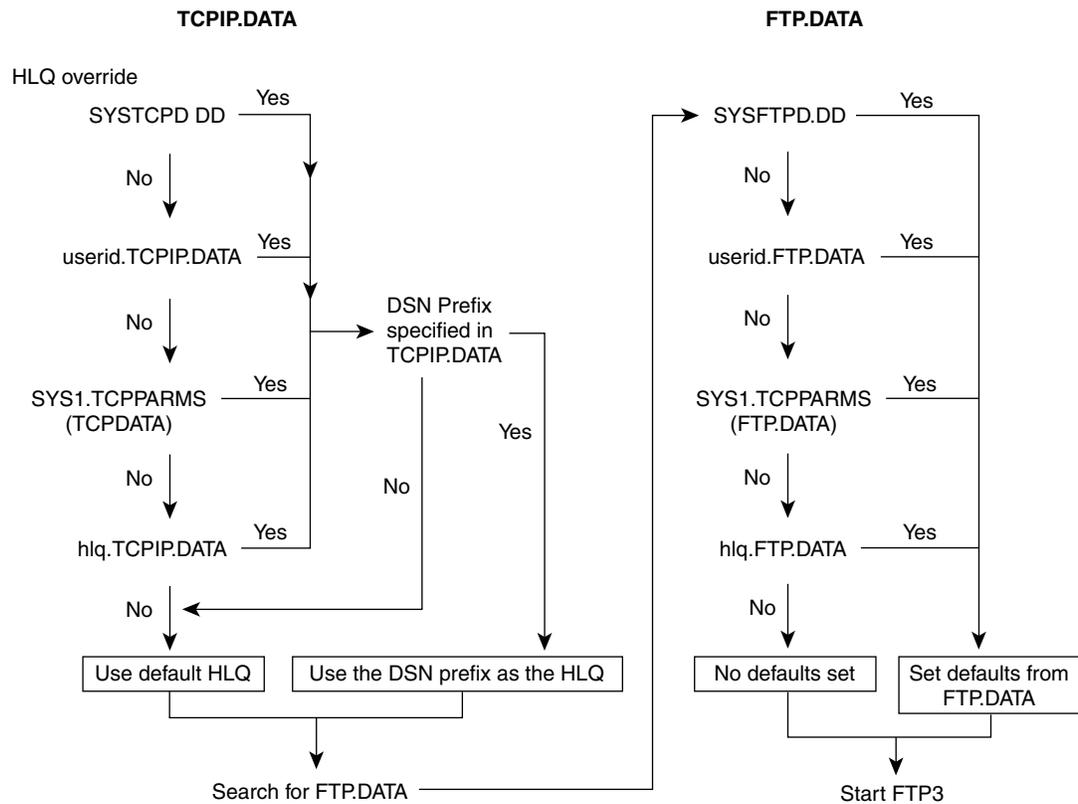
**Caution** These configuration data sets must be preconfigured before you execute FTP3.

If you want more details about how the configuration data sets are allocated, see Dynamic Data Set Allocation.

## Dynamic Data Set Allocation

Client FTP3 searches first for the TCPIP.DATA file. If the TCPIP.DATA file is found and contains a DATASETPREFIX statement, the prefix will be used to search for the FTP.DATA file. You can override the search sequence if you want to use high-level qualifiers other than the defaults for the configuration data sets. If the data sets are not found, the search continues as described for each file. Figure 3-2 illustrates the allocation sequence.

**Figure 3-2 Dynamic Data Set Allocation**



12898

**TCPIP.DATA Allocation**

FTP3 follows the search path described below to locate the TCPIP.DATA file.

- 1 FTP3 looks first for a SYSTCPD DD statement, which is used to define an override for the high-level qualifier.
- 2 If no SYSTCPD DD statement is located, FTP3 searches for a *userid.TCPIP.DATA* file.  
The user ID is the TSO user ID of the TSO user issuing the FTP3 command or the user ID of the FTP3 batch job if the command is issued from a batch job.
- 3 If no *userid.TCPIP.DATA* file is located, FTP3 searches for a TCPDATA member in the SYS1.TCPPARMS partitioned data set (PDS).  
SYS1.TCPPARMS will be used during initialization; it may have been created previously if the IBM product was installed.
- 4 If the TCPDATA member is not located, FTP3 looks for the *hlq.TCPIP.DATA* data set.
- 5 Once the TCPIP.DATA file is located, FTP3 uses the high-level qualifier defined by the DATASETPREFIX statement in the TCPIP.DATA file.

- 
- 6 Overriding: If no TCPIP.DATA file is located, no default values are imposed, with the exception that FTP3 will connect to either
    - the resolved subsystem ID of the started procedure name or the job name specified in the TCP option of the FTP3 command

*or*

    - the subsystem ID specified in the SSID option in the FTP3 command.
  - 7 If the FTP3 command includes neither the SSID nor TCP option, FTP3 connects to the SSID default subsystem ID, ACSS.

### FTP.DATA Allocation

After the TCPIP.DATA search is complete, FTP3 starts searching for the FTP.DATA file. Once the FTP.DATA file is located, the search is complete.

- 1 It searches first for a SYSFTPD DD statement.

The data set name specified in this statement will be used by FTP3.
- 2 If no SYSFTPD DD statement is found, FTP3 searches for a *userid.FTP.DATA* file.

The user ID prefix in *userid.FTP.DATA* is the TSO user ID of the TSO user issuing the FTP3 command or the user ID of the FTP3 batch job.
- 3 If no *userid.FTP.DATA* file is located, FTP3 searches for an FTPDATA member in the SYS1.TCPPARMS PDS.

SYS1.TCPPARMS will be used during initialization; it may have been created previously if the IBM product was installed.
- 4 If a *userid.FTP.DATA* is not found, FTP3 looks for *hlq.FTP.DATA*.
- 5 If no FTP.DATA file is located, or if configuration parameters are missing from FTP.DATA, FTP3 sets no defaults except for these timeout parameters:
  - CCONNTIME
  - DATACTIME
  - DCONNTIME
  - INACTTIME
  - MYOPENTIME

The default for each of these parameters is 300 seconds.

Normal settings for local SITE parameters are determined by the Cisco IOS for S/390 administrator in the Server FTP configuration statements in APPCFGxx.

### Related References on the FTP.DATA File

For additional detailed information about the FTP.DATA file, refer to *SC31-7134-01 IBM TCP/IP V3R1 for MVS: Customization and Administration Guide*.

### Changing the High-Level Qualifier

Client FTP3 uses both implicit and explicit data set allocation. You can use the pre-assigned data set names, *hlq*.TCPIP.DATA and *hlq*.FTP.DATA, or you can override the pre-assigned data set names with your JCL.

Client FTP3 is distributed with the high-level qualifier TCPIP. The name is defined in the T00PFUM module. You may change the default high-level qualifier by doing one of the following:

- Apply USERMOD member T00PFUM1 in the CNTL data set to change the 26-character field in T00PFUM to a value suitable for your site. The APAR contains instructions for changing the high-level qualifier.
- Specify the DATASETPREFIX statement in the TCPIP.DATA data set at execution time.

### TCPIP.DATA

The data set TCPIP.DATA defines TCP/IP parameters. An example of this file is located in the SAMP data set member T00TCPDS.

FTP3 does not support multiple host definitions in a single TCPIP.DATA file. If you have a TCPIP.DATA file which contains more than one host definition, you need to modify your file and create a separate TCPIP.DATA file for each host.

Table 3-1 describes the parameters are supported in the TCPIP.DATA data set.

**Table 3-1 TCPIP.DATA Parameters**

Parameter	Description
<b>DATASETPREFIX</b> <i>dsn_prefix</i>	<i>dsn_prefix</i> identifies the high-level qualifier to be used to determine the <i>hlq</i> .FTP.DATA data set name and the translate table data set name. Default: None
<b>HOSTNAME</b> <i>host_name</i>	<i>host_name</i> specifies the host name of the local FTP server. Default: None
<b>DOMAINORIGIN</b> <i>origin</i>	The <i>origin</i> is to be appended to the <i>host_name</i> to form the fully-qualified host name. Default: None
<b>MESSAGECASE MIXED   UPPER</b>	Specifies whether messages from the FTP client will be displayed in mixed case or translated to upper case. Default: <b>MIXED</b>
<b>TCPIPJOBNAME</b> <i>tcPIP_proc</i>	Identifies the started procedure name or job name of the Cisco IOS for S/390 stack. This is resolved to a subsystem ID by the client. Default: None

---

## FTP.DATA

The FTP.DATA data set defines local SITE parameters that are sent to the local host in the form of **SITE** commands. You can set your own values by creating a *userid*.FTP.DATA data set. An example of this file is contained in the SAMP data set member T00FTPDS.

The parameters listed in Table 3-2 are set in the FTP.DATA file. Many of these parameters correspond to **SITE** commands available in Cisco IOS for S/390 Server FTP. You can use the **LOCSITE** command to change any of these parameters during the FTP session.

**Table 3-2 FTP.DATA Parameters**

Parameter	Description
<b>AUTOMOUNT</b> TRUE   FALSE	If TRUE, the client will send <b>SITE AUTOMount</b> to local host; if FALSE, the client will send <b>SITE NOAUTOMount</b> . Default: None.
<b>AUTORECALL</b> TRUE   FALSE	If TRUE, the client will send <b>SITE AUTOREcall</b> to local host; if FALSE, the client will send <b>SITE NOAUTOREcall</b> . Default: None.
<b>BLKSIZE</b> <i>blk_size</i>	The client will send a <b>SITE BLKSize=blk_size</b> command to the local host.
<b>BLOCKSIZE</b> <i>blk_size</i>	Default: None.
<b>CHKPTint</b> <i>checkpoint_interval</i>	The client will send a <b>SITE CHKPTint=checkpoint_interval</b> command to the local host. Default: None.
<b>CCONNTIME</b> <i>close_connection_time</i>	Set internal timer. Defines the timeout value (in seconds) for closing a control connection. Default: 300 seconds.
<b>DATACLASS</b> <i>SMS_data_class</i>	Sends a <b>SITE DATAClass=SMS_data_class</b> command to the local host. Default: None.
<b>DATACTTIME</b> <i>data_connection_timeout_time</i>	Sets internal timer; sends <b>SITE DIDle=data_connection_timeout_time</b> to the local host. Defines the timeout value when sending or receiving data. Default: 300 seconds.
<b>DCBDSN</b> <i>DCB_dataset_name</i>	Sends a <b>SITE DCBdsn=DCB_dataset_name</b> command to the local host. Default: None.
<b>DCONNTIME</b> <i>data_connection_close_time</i>	Sets internal timer; sends a <b>SITE DClose=data_connection_close_time</b> command to the local host. Defines the timeout value when closing a data connection. Default: 300 seconds.
<b>DIRECTORY</b> <i>directory_blocks</i>	Sends a <b>SITE Directory=directory_blocks</b> command to the local host. Default: None.
<b>DIRECTORYMODE</b> TRUE   FALSE	If TRUE, the client will send <b>SITE DIRECTORYmode</b> to local host; if FALSE, the client will send <b>SITE DATASETmode</b> . Default: None.
<b>FILETYPE</b> SEQ   JES	Sends a <b>SITE FILEType=SEQ   JES</b> command to the local host. Default: None.
<b>INACTTIME</b> <i>control_connection_inactive_time</i>	Sets internal timer. Defines the timeout for a reply on the control connection. Default: 300 seconds.
<b>LRECL</b> <i>logical_record_length</i>	Sends a <b>SITE LRecl=logical_record_length</b> command to the local host. Default: None.
<b>MGMTCLASS</b> <i>SMS_management_class</i>	Sends a <b>SITE MGmtclass=SMS_management_class</b> command to the local host. Default: None.

**Table 3-2 FTP.DATA Parameters (Continued)**

<b>MIGRATEVOL</b> <i>migrate_volser</i>	Default: None.
<b>MYOPENTIME</b> <i>data_connection_open_time</i>	Sets internal timer; sends a <b>SITE DOpen=</b> <i>data_connection_open_time</i> command to the local host. Defines the timeout when opening a new control or data connection.  Default: 300 seconds
<b>NCP</b> <i>number_of_channel_programs</i>	Sends a <b>SITE NCP=</b> <i>number_of_channel_programs</i> command to the local host.  Default: None
<b>PRIMARY</b> <i>primary_space</i>	Sends a <b>SITE PRImary=</b> <i>primary_space</i> command to the local host.  Default: None.
<b>RDW</b> TRUE   FALSE	If TRUE, the client will send <b>SITE RDW</b> to local host; if FALSE, the client will send <b>SITE NORDW</b> .  Default: None.
<b>RECFM</b> <i>record_format</i>	Sends a <b>SITE RECFm=</b> <i>record_format</i> command to the local host.  Default: None.
<b>RETPD</b> <i>retention_period</i>	Sends a <b>SITE RETpd=</b> <i>retention_period</i> command to the local host.  Default: None.
<b>SECONDARY</b> <i>secondary_space</i>	Sends a <b>SITE SECONDAry=</b> <i>secondary_space</i> command to the local host.  Default: None.
<b>SPACETYPE</b> BLOCKS   CYLINDERS   TRACKS	Sends a <b>SITE BLocks/CYLinders/TRACKs</b> command to the local host.  Default: None.
<b>STORCLASS</b> <i>SMS_storage_class</i>	Send <b>SITE STORclass=</b> <i>SMS_storage_class</i> command to the local host.  Default: 300 seconds.
<b>UNITNAME</b> <i>unit_name</i>	Sends a <b>SITE Unit=</b> <i>unit_name</i> command to the local host.  Default: None.
<b>VOLUME</b> <i>volume_serial</i>	Sends a <b>SITE VOLume=</b> <i>volume_serial</i> command to the local host.  Default: None.
<b>WRAPRECORD</b> TRUE   FALSE	If TRUE, the client will send <b>SITE WRAPRecord</b> to local host; if FALSE, the client will send <b>SITE NOWRAPRecord</b> .  Default: None.

## General Client FTP3 Operation

These steps outline the general procedure for using Client FTP3 program:

- Issue the FTP3 command (with any optional parameters) to log on to the remote host. You are automatically logged on to the local MVS Cisco IOS for S/390 host.
- If you do not specify a remote host on your FTP3 command line, you will be prompted for the remote host.
- If the remote host name is in the NETRC file, the user ID and password are taken from the NETRC file. If the host name is not in the NETRC file, you will be prompted for a user ID and password.

If your user ID and password fail the logon because one or both were entered incorrectly, you must enter the **LOG** command to sign on to the remote host.

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

## Path Name

A path name is a string that identifies a file to a file system. A path name must contain a device and/or directory name and a file name. The FTP specification does not specify a standard path name convention. Each user must follow the file naming conventions of the file systems involved in the transfer. Consult the System Administrators at the host sites involved in the transfer for file naming conventions.

Many of the Client FTP3 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 FTP3 Command Conventions

Table 3-3 provides general notes that apply to the Client FTP3 commands:

**Table 3-3 Client FTP3 Command Conventions**

Convention	Description
Program Prompt	To indicate successful completion of most commands, the Client FTP3 program gives a new prompt.
Completion of Data Transfer	Final completion of the data transfer command is indicated with a message.
Testing the Control Connections	You can use the <b>VERBOSE</b> command to see the specific FTP commands and responses sent and received over the control connections as a result of Client FTP3 commands. If you want more information about this test, refer to <b>RFC 959</b> .
Case Sensitivity	The Client FTP3 commands are not case sensitive.
Abbreviations in Commands	Abbreviations are permitted if they are not ambiguous. For example, you can type <b>AB</b> for <b>ABORT</b> but you cannot type only <b>A</b> because several commands begin with this letter.
Brackets in Commands	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.
Syntax Conventions	Command words are shown in uppercase and parameters are shown in lowercase. When the actual values for a parameter are given (such as <b>LSOUTPUT</b> ), they are shown in uppercase.
Example Conventions	In all examples of Client FTP3 input and output in this manual, user entries are shown in boldface type.  The Client FTP3 examples in this manual assume that you have issued <b>OPEN</b> and implied <b>LOG</b> commands similar to this example to connect an IBM MVS TSO user to another system:

## The NETRC File

The Client FTP3 program uses information in the NETRC file for automatic login to the remote host. The NETRC filename is assumed to be *userid*.NETRC. To use a different naming convention for this file, you must pre-allocate it with an NETRC DD statement.

You can specify any number of MACHINE/LOGIN/PASSWORD sets to define remote hosts. If you define a machine with no login, password, and/or account, you are prompted for this information when needed. When a user connects to a remote host, if a remote MACHINE/LOGIN/PASSWORD set exists for the host, this set logs the user on to the remote host.

If you have not created a NETRC file, you are prompted to supply a user ID and password for the remote host whenever you open a connection to a remote host.

This is the format of the NETRC file; edit this file to specify user IDs, passwords, and accounts, or to create a macro.

```
MACHINE remhost LOGIN userid (for remhost) PASSWORD password (for remhost)
[ACCOUNT account (for remhost)
```

If an installation does not require a password and/or an account, these can be omitted. Before using the Client FTP3 commands, you must log on to a remote host with an **OPEN** command.



**Caution** You should use your local access control facility to protect NETRC files since these files can contain valid user ID password combinations for remote hosts. Only the TSO user ID using the NETRC file should be able to read their own NETRC file.

The NETRC file can also be used in batch jobs by using the NETRC invocation along with a NETRC DD pointing to your NETRC data set.

---

**Note** The batch invocation for NETRC does not assume a default FTP.NETRC data set name.

---

## Client FTP3 Commands

Table 3-4 gives a brief description of each command and its function; detailed descriptions of each command follow the table.

**Table 3-4 Client FTP Commands**

Command	Function
?	Get help on all commands or one command
ACCOUNT	Send account information
APPEND	Append data to file on remote host
ASCII	Set the transfer type to ASCII
BINARY	Set the file transfer mode to binary
CD	Change working directory on the remote
CDUP	Change to parent directory on remote
CLOSE	Disconnect from remote host
CWD	Change working directory (same as CD)
DEBUG	Toggle to activate or deactivate the <b>DEBUG</b> option
DELETE	Delete file on remote host
DELIMIT	Display file delimiter on local host
DIR	List directory on remote host
EBCDIC	Set file transfer type to EBCDIC
GET	Copy file from remote host

---

**Table 3-4 Client FTP Commands (Continued)**

<b>Command</b>	<b>Function</b>
<b>HELP</b>	Display help information
<b>LCD</b>	Change working directory on local host
<b>LMKDIR</b>	Create a PDS on local host
<b>LOCSITE</b>	Send <b>SITE</b> to local host
<b>LOCSTAT</b>	Display status for local host
<b>LPWD</b>	Print the local directory name
<b>LS</b>	List file names on remote host
<b>MDELETE</b>	Delete multiple files on remote host
<b>MGET</b>	Retrieve multiple files from remote host
<b>MKDIR</b>	Create a new directory on remote host
<b>MODE</b>	Set transmission mode
<b>MPUT</b>	Store multiple files on remote
<b>NOOP</b>	Send <b>NOOP</b> to remote
<b>OPEN</b>	Open connection to remote host
<b>PASS</b>	Send password to remote host
<b>PROMPT</b>	Toggle prompting for <b>MGET</b> , <b>MPUT</b> and <b>MDELETE</b>
<b>PUT</b>	Store file on remote host
<b>PWD</b>	Show name of working directory on remote host
<b>QUIT</b>	Exit FTP
<b>QUOTE</b>	Send uninterpreted string to remote host
<b>RENAME</b>	Rename file on remote host
<b>RESTART</b>	Restart a transfer
<b>RMDIR</b>	Remove directory on remote host
<b>RUNIQUE</b>	Toggle <b>STORAGE</b> method on local host
<b>SENDSITE</b>	Toggle sending of <b>SITE</b> command
<b>SITE</b>	Send <b>SITE</b> parameters to remote host
<b>STATUS</b>	Send <b>STATUS</b> to remote
<b>STRUCT</b>	Set file structure
<b>SUNIQUE</b>	Store unique file names on remote host
<b>SYSTEM</b>	Display remote host operating system
<b>TSO</b>	Execute <b>TSO</b> command
<b>TYPE</b>	Set transfer type
<b>USER</b>	Send userid to remote host
<b>VERBOSE</b>	Display system replies with additional information

## ? Command

Use the ? command to display information about using the Client FTP3 program. This command is very similar to the **HELP** command.

? [ *command\_name* ]

### Syntax Description

*command\_name* Command for which you are requesting information.

### Default

If the ? command is specified with no arguments, it displays a list of Client FTP3 commands.

### Usage Guidelines

The ? command can be used either with or without arguments.

If *command\_name* is specified as an argument to the ? command, a line of information displays similar to the information in the table in Client FTP3 Commands. The line shows the command syntax and a short description of the command function.

### Example

```
? DIR
DIR pathr pathl - Directory list of remote host
```

---

## ACCOUNT

The **ACCOUNT** command displays information that is needed by the remote host. Refer to documentation for the remote FTP server for information that it needs.

**ACCOUNT** [*account\_info*]

### Syntax Description

*account\_info*      Information to be sent to the remote host.

# APPEND

The **APPEND** command requests that a file from the local host be appended to a file at the remote host.

```
APPEND [ local_path ] [ remote_path ]
```

### Syntax Description

*local\_path*            Name of the file to be retrieved from the local host.

*remote\_path*          Remote file to which the local file 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 depends on the associated Server FTP.

Data set attributes are maintained for the transferred data set.

If the data set already exists, and the LRECL is less than that of the transferred data set, the transmitted data set will be truncated.

---

## **ASCII**

The **ASCII** command sets the data type to ASCII for the data transfer. This command is equivalent to a **TYPE** command with the ASCII (A) parameter specified.

**ASCII**

## **BINARY**

The **BINARY** command sets the data type to binary for the data transfer. This command is equivalent to a **type** command with the image (I) parameter specified.

### **BINARY**

This command has no arguments or keywords.

---

## CD

The **CD** command requests that the remote Server FTP change the current directory to a new directory.

```
CD [ path_name ]
```

### Syntax Description

*path\_name* Indicates to the remote Server FTP the name of the directory to be made the current directory.

### Default

If you omit *path\_name*, the Client FTP3 program prompts you for the desired value.

### Usage Guidelines

The syntax for *path\_name* depends on the associated Server FTP.

If you want to specify a path name that is not a subdirectory of the current directory, enclose the path name in single quotes. Do not leave a space after the first quote; FTP3 will ignore the quotation mark.

A UNIX Server FTP in session with the Client FTP3 program is using /u/user1/work as the current directory. If a **CD junk** command is issued by the Client FTP3 to that UNIX Server FTP, the resulting current directory is /u/user1/work/junk. The same result is achieved by specifying **CD /u/user1/work/junk**.

### Example

The following example shows a change of directory to a remote UNIX system:

```
CD /u/1pn/d.ddn  
250 CWD command successful.
```

The following example shows a change to a directory on a remote system running Cisco IOS for S/390:

```
CD ACCES  
250 "'USER1.ACCES.'" is current prefix
```

The following example shows a change to a specific directory:

```
CD `TEST.DIR`  
250 "'TEST.DIR'" is current prefix
```

## CDUP

The **CDUP** command directs the remote 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**

A UNIX Server FTP in session with the Client FTP3 program has /u/user1/work as the current directory. If a **CDUP** command is issued by the Client FTP3 to that UNIX Server FTP, the resulting current directory is the parent of the old current directory (/u/user1).

### Example

The following example shows a change to the parent directory of the old current directory on a remote UNIX system:

```
CDUP  
250 CWD command successful.
```

---

## CLOSE

The **CLOSE** command logs you out and terminates the connection between you and the remote Server FTP.

**CLOSE**

### Example

```
CLOSE  
221 Goodbye.
```

## DEBUG

The **DEBUG** command displays all commands and responses going to and from the Server FTPs and the user. This command is equivalent to specifying the TRACE option on the Client FTP3 command line.

### **DEBUG**

#### Usage Guidelines

The **DEBUG** command toggles the previous state. If **DEBUG** or **TRACE** was turned on previously, then a subsequent **DEBUG** or **TRACE** command will turn it off.

#### Example

```
DEBUG
```

---

# DELETE

The **DELETE** command directs the remote Server FTP to delete the specified file.

**DELETE** [ *path\_name* ]

## Syntax Description

*path\_name*            The specific file to delete.

## Default

If you omit *path\_name*, you are prompted to supply one.

## Usage Guidelines

The syntax for *path\_name* depends on the associated Server FTP.

## Example

```
DELETE oldfile  
250 DELE command successful.
```

## DELIMIT

The **DELIMIT** command displays the character delimiter used between the file name and the file type.

### **DELIMIT**

The delimiter cannot be changed by this command.

---

# DIR

The **DIR** command requests that the remote Server FTP provide a directory list for the specified path.

```
DIR [ remote_path ] [ local_path ] [ ( DISK ]
```

## Syntax Description

*remote\_path* Remote directory to be listed. If *remote\_path* is not specified, you receive a list of your current directory.

*local\_path* File to which the directory list is written. If *local\_path* is not specified, the directory list is displayed on your screen.

**DISK** Stores output in a data set named *local\_directory*.FTP.DIROUTP. If *local\_directory* is a PDS, output is stored in member DIROUTP.

## Example

The following example lists the contents of a directory on an MVS system.

```
dir /export/home/user1/mvs
150 ASCII data connection for /bin/ls (138.42.224.15,4126) (0 bytes).
-rw-----      1 user1      dvlp      2730 Oct 22      08:36 channel
-rw-----      1 user1      dvlp      2901 Sep  1      09:17 comten
-rw-----      1 user1      dvlp      1276 Oct 21      14:43 dump
-rw-----      1 user1      dvlp        729 Nov 12      05:24 ibmlink
-rw-----      1 user1      dvlp        751 Nov 12      07:41 ibmlink2
226 ASCII Transfer complete.
```

## EBCDIC

The **EBCDIC** command sets the data type to EBCDIC for data transfer. The command is equivalent to the **type** command with the EBCDIC (**E**) parameter specified.

### **EBCDIC**

#### Example

```
EBCDIC  
EBCD ENTERED  
stat
```

---

# GET

The **GET** command requests that a file from the remote host be copied to a file on the local host by the appropriate Server FTPs. The file to be retrieved is always at the remote host, and the file to be copied into is always at the local host.

```
GET [ remote_path ] [ local_path ] [ ( REPLACE ) ]
```

## Syntax Description

<i>remote_path</i>	File name to be retrieved from the remote host.
<i>local_path</i>	File name at the local host into which the file from the remote host is copied
<b>REPLACE</b>	Specifies that the data set on the local host be overwritten.

## Default

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

## Usage Guidelines

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

---

**Note** The **FTP GET GDGBASE** command gets only the most recent data set. The action of the **REPLACE** option is dependent on the configuration of the FTP statement in your APPCFGxx file. If **OVERWRITE** is configured, the file name will be overwritten even without the **REPLACE** option. If **NOOVERWRITE** is configured, the file will not be overwritten even if the **REPLACE** option is specified. If **SITEOVERWRITE** is configured, you must issue a **SITE OVERWRITE** command and specify **REPLACE** to overwrite an existing file.

---

## HELP

The **HELP** command requests help information from the local Server FTP.

**HELP** [**ALL** | **?** | *command* | [**SERVER** [*remote\_command*]]]

### Syntax Description

<b>HELP</b>	(Without any parameters) requests a list of commands available to the FTP3 client.
<b>ALL</b>	Provides a brief description of all the commands available to the FTP3 client.
<b>?</b>	Lists the syntax for the <b>HELP</b> command.
<i>command</i>	Provides a description of a specific command from the local FTP3 client.
<b>SERVER</b> <i>remote_command</i>	Requests information from the remote FTP server. If a specific <i>remote_command</i> is specified, a description of the command is returned from the remote server.

### Default

If no command is specified, a list of supported commands is returned.

---

## LCD

The **LCD** command changes the current working directory on the local host.

**LCD** [*local\_path\_name*]

### Syntax Description

*local\_path\_name* Specifies the name of the directory on the local host. If you want to specify a path name that is not a subdirectory of the current directory, enclose the path name in single quotes. Do not leave a space after the first quote; FTP3 will ignore the quotation mark.

## LMKDIR

The **LMKDIR** command creates a directory (or PDS) on the local host.

**LMKDIR** [*local\_path\_name*]

### Syntax Description

*local\_path\_name* Specifies the name of the directory or PDS that is to be created on the local host. If you want to specify a path name that is not to be appended to the current directory, enclose the path name in single quotes. Do not leave a space after the first quote; FTP3 will ignore the quotation mark.

---

## LOCSITE

The **LOCSITE** command allows you to specify **SITE** information to be used by the local host.

**LOCSITE** [ *options* ]

### Syntax Description

*options*                      Local site information. For a list of the **SITE** commands, read the **SITE** section in **Server FTP**.

## LOCSTAT

The **LOCSTAT** command displays local FTP status information.

**LOCSTAT**

---

## **LPWD**

The **LPWD** command displays the name of the current working directory on the local host.

**LPWD**

## LS

The **LS** command requests a remote Server FTP to provide a list of file names for the specified path.

```
LS [ remote_path ] [ local_path ] [ ( DISK )
```

### Syntax Description

*remote\_path* Path to be listed from the remote host.

*local\_path* Local file into which the list from the remote server is printed.

**DISK** Stores output in a data set named *local\_directory*.FTP.LSOUTPUT. If *local\_directory* is a PDS, output is stored in member LSOUTPUT.

### Default

If a local path is not specified, the list of files appears on your screen.

### Usage Guidelines

If the path specifies a directory or other group of files, the remote Server FTP transfers a list of files.

The syntax for each path depends on the associated FTP server.

If a local path is specified, the list of files is written to the specified file.

### Example

The following example shows of the **LS** command with parameters:

```
ls /export/home/user1 unix.dir.temp
150 ASCII data connection for /bin/ls (138.42.224.15,4134) (0 bytes).
226 ASCII Transfer complete.
 350 bytes received in 1.54 seconds (227 bytes/s)
```

This is an example of the **LS** command without parameters:

```
ls
150 ASCII data connection for /bin/ls (138.42.224.15,4135) (0 bytes).
-Transfer complete
channel
comten
dump
filea
tempfile
226 ASCII Transfer complete.
```

---

## MDELETE

The **MDELETE** command allows you to delete multiple files on the remote host.

```
MDELETE [remote_path_name] | [mask]
```

### Syntax Description

*remote\_path\_name* Name of the file to be deleted on the remote host.

*mask* Mask to use to describe files. This mask complies with mask conventions on the remote host.

### Example

```
MDELETE userid.myfiles.*  
MDELETE file199?
```

## MGET

The **MGET** command copies multiple files from the remote host to the local host. You cannot specify a different name for the files that are retrieved. Filenames created on the local host will be identical to those on the remote host.

```
MGET [remote_path_name] [( REPLACE)]
```

### Syntax Description

*remote\_path\_name* Name of the files to be copied from the remote host.

**REPLACE** Specifies that the data set on the local host be overwritten.

---

**Note** The action of the **REPLACE** option is dependent on the configuration of the **FTP** statement in your **APPCFGxx** file. If **OVERWRITE** is configured, the file name will be overwritten even without the **REPLACE** option. If **NOOVERWRITE** is configured, the file will not be overwritten even if the **REPLACE** option is specified. If **SITEOVERWRITE** is configured, you must issue a **SITE OVERWRITE** command and specify **REPLACE** to overwrite an existing file.

---

### Example

```
ftp> lcd mget.pds
"'GAM2.MGET.PDS'" partitioned data set is current directory
ftp> cd tstdir
----> CWD tstdir
250 CWD command successful.
ftp> mget TST*
----> PORT 138,42,160,55,17,34
200 PORT command successful.
----> NLST TST*
150 ASCII data connection for /bin/ls (138.42.160.55,4386) (0 bytes).
226 ASCII Transfer complete.
GET TST1
----> PORT 138,42,160,55,17,35
200 PORT command successful.
550 Catalog structure invalid or user lacks authority to catalog
GET TST2
----> PORT 138,42,160,55,17,36
200 PORT command successful.
----> RETR TST2
150 ASCII data connection for TST2 (138.42.160.55,4388) (29 bytes).
226 ASCII Transfer complete.
GET TST3
----> PORT 138,42,160,55,17,37
200 PORT command successful.
----> RETR TST3
150 ASCII data connection for TST3 (138.42.160.55,4389) (37 bytes).
226 ASCII Transfer complete.
ftp>
```

---

# MKDIR

The **MKDIR** (make directory) command directs a remote Server FTP to create the specified directory.

**MKDIR** [ *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 depends on the associated Server FTP.

## Example

A UNIX Server FTP in session with the Client FTP3 program has /u/user1/work as the current directory. If a **mkdir junk** command is issued by the Client FTP3 to that UNIX Server FTP, the subdirectory junk is created in the current directory (/u/user1/work/junk). The same result is achieved by specifying **MKDIR /u/user1/work/junk**.

```
MKDIR /u/1pn/d.new  
257 MKD command successful.
```

## MODE

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

### **MODE S | B | C**

- Block mode

Block mode formats the data and allows for restart procedures.

- 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.

- Compressed mode

Compressed mode transmits the data blocks with no repetitive characters or blanks.

For the purpose of standardized transfer, 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, and the receiving host performs the inverse translation to its internal representation. Since 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.

### Syntax Description

<b>S</b>	Stream mode.
<b>B</b>	Specifies block mode.
<b>C</b>	Compressed mode. Data transfer type must be set to EBCDIC.

### Usage Guidelines

One of the three codes (either S, B, or C) is required as an argument on the **MODE** command.

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 Server FTP documentation if you have questions concerning transmission mode support.

### Block Mode

Block mode is indicated 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.

---

### Stream Mode

Stream mode is set 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.

### Compressed Mode

Compressed mode is set with the character **C**. In compressed mode, data blocks are transmitted with one or more header bytes. Logical record boundaries of the data set are preserved with compressed mode. Data is transmitted with no repetitive characters or blanks.

## MPUT

The **MPUT** command copies multiple files from the local host to the remote host.

You cannot specify a different name for the files that are transferred. Filenames created on the remote host will be identical to those on the local host.

**MPUT** [*local\_path\_name*] | [*mask*]

### Syntax Description

*local\_path\_name*      File name on the local host.

*mask*                      Mask to use to describe files. This mask complies with mask conventions on the remote host.

### Example

```
MPUT userid.myfiles.*  
MPUT file199?
```

---

## **NOOP**

The **NOOP** command is used to test the response of the remote host.

**NOOP**

## OPEN

The **OPEN** command is used to open a connection to the FTP server on the remote host. This command can be issued while you are in the FTP environment.

**OPEN** *remote\_host* [ *port\_number* ]

### Syntax Description

*remote\_host*      Name of the remote host. Client FTP3 will automatically connect to this host at initialization. This parameter is required. If you do not supply this parameter, you will be prompted for it.

*port\_number*      Port number of the FTP server on the remote host.  
Default: 21

---

**Note** If you are currently connected to a remote host, you must disconnect before using the **open** command to connect to another host

---

---

## PASS

The **PASS** command is used to send a password to the remote host.

**PASS** *password*

### Syntax Description

*password*        Specifies your password on the remote host.

### Usage Guidelines

The **PASS** command must follow the **USER** command.

Read **USER** for more information.

## PUT

The **PUT** command requests that the local Server FTP copy a file to the remote system. The file to be copied is always at the local Server and the file destination is always at the remote Server.

**PUT** [*local\_pat*] [*remote\_path*]

### Syntax Description

<i>local_path</i>	File name of the file to be copied from the local server.
<i>remote_path</i>	File name of the file at the remote Server into which the file from the local server is copied. If no name is specified, the name from the local host is used.

### Usage Guidelines

The syntax for each path depends on the associated Server FTP. If you omit either path, you are prompted for the file name.

---

## PWD

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

**PWD**

### Syntax Description

This command has no arguments or keywords.

### Example

```
PWD  
257 "/u/lpn" is current directory.
```

## QUIT

The **QUIT** command terminates the Client FTP3 program.

### QUIT

#### Example

```
QUIT  
221 Goodbye.
```

---

## QUOTE

The **QUOTE** command sends an uninterpreted, unaltered character string to the remote Server FTP over the control connection. This mechanism sends FTP commands to the Server that the Client FTP3 program might not be able to send.

**QUOTE** [ *text* ]

### Syntax Description

*text* Sent to the Server over the control connection exactly as you enter it; if the text is omitted, you are prompted to enter it

### Example

```
QUOTE pasv  
227 Entering Passive Mode (26,131,0,17,4,216).
```

# RENAME

The **RENAME** command directs a remote Server FTP to rename a file.

```
RENAME [ 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.

### Example

```
RENAME titlecol coltitle  
350 File exists, ready for destination name  
250 RNT0 command successful.
```

---

## RESTART

The **RESTART** command restarts the last checkpointed file transfer.

### **REST**

The data set *USERID.FTP.CHKPOINT* on the local host holds the valid checkpoint for the last checkpointed file transfer. Any parameters that may have changed for the last transfer, such as transfer mode and type, must be re-set before restarting the transfer.

When file transfer is in blocked or compressed mode (mode=b/c), and Client FTP3 processes a **GET** or **PUT** command, it allocates a fixed-block data set named *userid.FTP.CHKPOINT*. It uses this file to save the **GET** or **PUT** command and checkpoint record returned from the FTP server. If the *userid.FTP.CHKPOINT* file already exists, Client FTP3 will rewrite the checkpoint data set from the beginning of the data set.

If the **GET** or **PUT** command is successful, the checkpoint data set is deleted upon completion of the command processing. If the **GET** or **PUT** command fails or is interrupted, the *userid.FTP.CHKPOINT* data set will be kept in the system for the **RESTART** command to use to restart the file transfer.

If the **RESTART** command is executed successfully, Client FTP3 will delete the checkpoint data set. If the **RESTART** command fails or is interrupted, the checkpoint data set is kept for future use.

Users are responsible for deleting the checkpoint data set once it is determined the data set is no longer needed.

For more information, read about the **RESTART** interval option for the **SITE** command in the **SITE** section in Server FTP.

## RMDIR

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

**RMDIR** [ *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 specified directory is removed.

The syntax of *path\_name* depends on the associated Server FTP.

---

**Note** Many systems require the directory to be empty before it can be removed.

---

### Example

As an example, if a UNIX Server FTP in session with the Client FTP3 program has /u/user1/work as the current directory, and a **RMDIR junk** command is issued by Client FTP3 to that UNIX Server FTP, the junk subdirectory of the current directory is removed. The same result is achieved by specifying **RMDIR /u/user1/work/junk**.

```
RMDIR /u/1pn/d.samp
250 RMD command successful.
```

---

## SENDSITE

The **SENDSITE** command is used to automatically send the **SITE** commands when sending a file to a remote host. This command is useful only with other IBM-based FTP servers.

### **SENDSITE**

The **SENDSITE** command is turned on when you start your FTP session. You can toggle it on and off by issuing the **SENDSITE** command. To determine the current state of the **SENDSITE** command, issue the **LOCSTAT** command.

# SITE

The **SITE** (site parameters) command provides the local 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 FTP3 command to find the **SITE** requirements for a specific local Server FTP. Otherwise, review the Server FTP documentation for the **SITE** requirements. **SITE** command parameters are described in the SITE section in Server FTP.

**SITE** *text*

The *text* argument is required and is passed through unchanged to the specified server.

---

## STATUS

The **STATUS** command requests status information from the remote system.

**STATUS** [ *path\_name* ]

### Syntax Description

*path\_name*            Directory or file name for which information is requested.

## **STRUCT**

The **STRUCT** (file structure) command provides information on file structure to the remote Server FTP.

**STRUCT F | R**

### Syntax Description

**F** File structure is specified by F. This is the default if no **STRUCT** command has been used. File structure is used for files with no internal structure, and the file is considered to be a contiguous sequence of data bytes. File structure is accepted for text files (in other words, files with type ASCII or EBCDIC) by all FTP implementations.

**R** Record structure is set by R. This is for files made up of sequential records.

### Usage Guidelines

Record structure is set by R. This is for files made up of sequential records.

### Example

```
STRUCT F
```

## **SUNIQUE**

Use the **SUNIQUE (STORE UNIQUE)** command to store transferred files by unique file names on a remote machine. **SUNIQUE** is a toggle command (meaning it is either off or on). When used, the target server automatically ensures that files received in FTP transfer are stored under a unique name.

**SUNIQUE**

### Usage Guidelines

The target remote FTP server must support the **STOU** (store unique) command.

If **SUNIQUE** is not used, FTP3 will issue the standard **STOR** command. If file names are not unique, files in the target directory could be overwritten.

Default for **SUNIQUE** is **OFF**.

---

## SYSTEM

The **SYSTEM** command displays the name of the operating system on the remote host.

**SYSTEM**

### Example

**SYSTEM**

215 MVS is the operating system of this server.

# TRACE

Use the **TRACE** command to activate or deactivate the tracing option. With the tracing option active, you can rerun failing commands to discover the reason for the failure.

### **TRACE**

#### Usage Guidelines

The **TRACE** command toggles the previous state. If **TRACE** or **DEBUG** was turned on previously, then a subsequent **TRACE** or **DEBUG** command will turn it off.

#### Example

**DEBUG**

---

## TSO

The **TSO** command requests the client FTP3 program to execute a TSO command for you.

**TSO** *command*

### Syntax Description

*command*        Specifies the TSO command.

## 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** Indicates 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*** Indicates the 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.

**Note** **A** is the default argument for the **TYPE** command.

---

**E** Sets the files type to EBCDIC. This type performs efficient transfer between hosts that use EBCDIC. Cisco IOS for S/390 Client FTP3 users usually use this type when copying files to their MVS host.

For transmission, data is 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 receiving host must know how the vertical format control is represented. Therefore, the ASCII and EBCDIC types have a second parameter specifying non-print, Telnet, or carriage control (ASA).

These are the vertical format control specification options:

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 (in other words, overprint).

### Default

ASCII is the default file type.

For both ASCII and EBCDIC file types, vertical format control **N** 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**, can also be set.

# USER

The **USER** command identifies your userid to the remote FTP server.

```
USER user_id [password] [old_password] [/ new_password]
```

### Syntax Description

<i>user_id</i>	Your login name on the remote host.
<i>password</i>	Your password on the remote host. You will be prompted for your password if you do not supply it.
<i>old_password</i>	Your current password on the remote host.
<i>new_password</i>	Your new password on the remote host.

### Usage Guidelines

You can issue the **USER** command to change your userid at any time while you are in the FTP environment.

Your password will not be printed if you allow the client FTP3 to prompt you for it.

For more information about using the NETRC file, read [The NETRC File](#).