

# Server FTP

---

This chapter describes the server program for FTP within Cisco IOS for S/390. It contains these sections:

- **Introducing Server FTP**  
Provides a brief overview of the Server FTP.
- **File Handling by the Server FTP**  
Describes how the Server FTP handles files.
- **Server FTP Commands**  
Describes the Server FTP commands and includes a table with brief descriptions for quick reference.

ALLO	HELP	MKD	REST
RMD	SITE	STAT	

- **Data Set Attributes**  
Describes the attributes of the data sets that can be read or written by the Server FTP.
- **Data Transfer Operations**  
Describes the commands (**STOR**, **APPE**, and **RETR**) that initiate data transfer.
- **Character Type Rules**  
Describes the transformation rules for creating and retrieving character-type files.
- **Binary-Type Rules**  
Describes the transformation rules for creating and retrieving binary-type files.
- **Non-Invertible Retrieval**  
Describes some of the changes Server FTP makes to a file that might not be able to be undone when the file is retrieved.
- **Other Features**  
Lists some other useful features of the FTP program.

## Introducing Server FTP

The Server FTP supports large-scale remote computing on a large IBM (and compatible) mainframe. It includes these types of support:

- File System Support

The Server FTP supports creation and retrieval of a subset of the disk formats provided by the file system of MVS.

- Record Structure Support

The MVS file system is record oriented. The FTP Server can translate character files between record structure locally and file structure remotely.

- Binary Files Support

The Server FTP can transfer large files of binary data efficiently. The parameters required for record-structured binary files (STRU R, MODE B, and TYPE I) are implemented. Restart markers support restarts of large file transfers.

- JES Internal Reader Support

Server FTP lets data transfer to MVS be submitted as a batch job to MVS via the JES Internal Reader facility (see the section of this chapter on procedures for JES Internal Reader support).

## File Handling by the Server FTP

The Server FTP can read an existing disk data set with a wide variety of disk formats and map it correctly into the specified FTP parameters for transmission across the network. The record-structured MVS file system forces you to set limits on the size of a record when a file is created. Many processors require this record limit to be a “card image” (80 characters). A source file prepared on a stream-oriented system and transferred to the mainframe can contain records that are too long, and you may want to specify a larger record size with the **SITE** command.

## Handling a Record Which is Too Long

When the Server FTP receives a file and finds a record too long, it does not discard data; it folds it into multiple records and informs you of its action. As each source language has a different continuation convention, folding the data in this manner is unlikely to match any of these conventions. When the Server FTP preserves data in this manner, you can easily fix the error later.

Individual warning messages are not issued for folded records. The Server FTP counts records folded and sends that count at the end of file transfer (if it is non-zero).

You can also have the Server FTP truncate rather than fold with the **ALLO R** command. See **ALLO** and **SITE** for details.

---

## Transferring Files to a Host

The FTP lets a character file be transferred to a host for one of three purposes: for printing, for storage and later retrieval, or for processing. Under MVS, each of these purposes requires a different file format that must be chosen when the file is created.

- By default, the Server FTP assumes “processing” and records the data in a format that is likely to be acceptable to most MVS processing programs (a card image data set).
- When doing a **STOR** into a data set, the Server FTP infers the purpose from the FTP parameters and disk parameters and performs the appropriate translations.
- To create a print file, a print type must be specified (TYPE AT, TYPE ET, TYPE AC, or TYPE EC).

The translations the Server FTP performs for printing or processing are not exactly *invertible* if the file is later retrieved with FTP. If you want information to be stored for later retrieval in exactly the same form, you must override the default parameters with the **SITE** command.

---

**Note** An exact representation of the network file is called *raveled*. Raveled files are invertible, meaning you can FTP them back to the originating operating system in exactly the same format they started with. Specify a raveled file when you want to store the file on MVS for later retrieval to the originating operating system. In most cases, a raveled file cannot be passed as input to any IBM processing program or sent to the printer.

---

For a detailed description of raveled and non-invertible files, see Non-Invertible Retrieval.

As an aid in setting data set attributes, canned attribute sets for the most common cases can be chosen by mnemonic name on the **SITE** command. For example, **SITE PRINT** chooses appropriate attributes for a print file.

## Sophisticated File Handling

An experienced user can use additional facilities for more control over Server FTP operations. You can supply explicit Data Control Block (DCB) parameters in the **SITE** command to cause the FTP Server to create any reasonable sequential data set format used by MVS (the Server FTP may also create unreasonable formats).

Also, the Server FTP’s default formats for creating new data sets are those generally used by MVS programs. You can override these formats to obtain information-conserving storage on the mainframe. The Server FTP lists the full MVS data set attributes as well as the FTP transfer parameters when a transfer starts and reports full statistics when the transfer completes.

## Transferring Files To a Tape

The FTP server supports file transfer to and from magnetic tape volumes. This option can be specified dynamically with the FTP Server **SITE** command described in this chapter.

### Configuration

To use this facility, you will need to set the parameters **LABEL** and **MOUNT** for the FTP statement in your APPCFGxx member of the PARM data set.

To provide installation defaults for tape data set allocation, the system administrator should set up a special GAT TYPE(TAPE) entry in the Generic Attributes Table (GAT) in APPCFGxx. Parameters of interest are COMPACT, LABEL(), PARALLELMOUNT, PRIVATE, and UNITCOUNT().

For more information on these parameters, refer to the *Cisco IOS for S/390 Customization Guide*.

### SITE Command Parameters

The Server FTP **SITE** command has parameters specifically for using FTP to transfer files to magnetic tape. These are:

- AUTOINDEX
- COMPACT
- DSEQ
- LABEL()
- MOUNT()
- PARALLELMOUNT
- PRIVATE
- TAPE

These parameters are available for using FTP to transfer files to disk, but may have special significance for transferring files to magnetic tape:

- EXPDT
- RETPD
- UCNT
- VCNT
- VOL(volser, volser)
- VSEQ

FTP Server Commands **APPE** and **RESTART** are not currently supported for the FTP to Tape facility.

For more information on these **SITE** parameters, read **SITE**.

### Using FTP to Write to Magnetic Tape

#### Cataloging Data Sets

FTP attempts to catalog all data sets created on tape. If the data set name matches an existing name, the transfer will occur but the catalog will not be updated. This may create a problem if a retrieve is issued for the tape version. Therefore, it is recommended that all data sets—disk or tape—have unique names. Use the **DELETE** command to uncatalog tape data sets.

Cataloged tape data sets are assumed to exist on standard label tapes. Use the **SITE** command with LABEL and DSEQ parameters when retrieving a cataloged tape data set if the tape does not have standard labels. It is recommended that you use standard label tapes whenever possible.

---

## Writing Multiple Data Sets to Tape

Users who want to write several data sets to tape in one FTP session should be aware that each file to be transferred will generate a mount request, but it may be for a different tape unit. Dynamic allocation does not support **RETAIN** or **UNIT AFFINITY**. A workaround would be to arrange with Operations to hard mount a tape and then reference the unit in the **SITE** command.

## Submitting Mount Requests

The mount request is issued by Cisco IOS for S/390, even if running client FTP in batch. This means that Operations should not terminate the mount request by cancelling the job requesting the mount. A WTOR message, ACC511A, is issued along with the mount request. A reply of **NO** to this message cancels the request.

## Preventing Timeouts on Data Transfers

FTP will timeout a data transfer request if the remote does not complete the data connection in a certain time. If the remote is another MVS system using tapes (or recalling a data set) it will require a tape mount on the remote system before it can complete the data connection. For this reason, FTP will use the longer of **MOUNT** or **HSM** times, or 30 minutes if neither tape support nor **HSM** support is configured.

## Using Tape Data Sets on Remote Hosts

The problem of a remote system using tape data sets should also be considered when configuring **DATAIDLE** time. If a remote is reading a multivolume , for example, it might have to stop the data transfer between volumes while the next tape is mounted. The **DATAIDLE** time could expire while this is happening.

# Server FTP Commands

Table 5-1 lists the Server FTP commands. The Cisco IOS for S/390 Server FTP program supports most of the FTP commands defined in the FTP specification, *RFC 959, File Transfer Protocol (FTP)*. Server FTP commands are accepted by the local FTP server when submitted by a remote client. The January 16, 2008 FTP server will reply or respond to the FTP commands listed here:

**Table 5-1 Server FTP Commands Supported by the Cisco IOS for S/390 Server**

Command	Function
<b>ABOR</b>	Abort
<b>ACCT</b>	Account
<b>ALLO</b>	Allocate
<b>APPE</b>	Append (with create)
<b>CDUP</b>	Change to parent directory
<b>CWD</b>	Change working directory
<b>DELE</b>	Delete
<b>HELP</b>	Help
<b>LIST</b>	List
<b>MKD</b>	Make directory

**Table 5-1 Server FTP Commands Supported by the Cisco IOS for S/390 Server (Continued)**

Command	Function
<b>MODE</b>	Transfer mode
<b>NLST</b>	Name list
<b>NOOP</b>	No operation
<b>PASS</b>	Password
<b>PASV</b>	Passive
<b>PORT</b>	Data port
<b>PWD</b>	Print working directory
<b>QUIT</b>	Logout
<b>REIN</b>	Re-initialize
<b>REST</b>	Restart
<b>RETR</b>	Retrieve
<b>RMD</b>	Remove directory
<b>RNFR</b>	Rename from
<b>RNTO</b>	Rename to
<b>SITE</b>	Site parameters
<b>STAT</b>	Status
<b>STOR</b>	Store
<b>STRU</b>	File structure
<b>TYPE</b>	Representation type
<b>USER</b>	User name

Not all of the commands listed in are documented in this publication. Only those which have been enhanced for Cisco IOS for S/390 are included. The others in this list conform to their descriptions in *RFC 959*. Table 5-2 lists commands not supported by the Cisco IOS for S/390 FTP server.

**Table 5-2 Commands Not Supported by the Cisco IOS for S/390 Server**

Command	Description
<b>SMNT</b>	Structure mount
<b>STOU</b>	Store unique
<b>SYST</b>	System

The supported FTP commands are described in detail throughout the remainder of this chapter. The **HELP** command also provides information about the Server FTP. Read **HELP** for guidelines on using the **HELP** command.

---

# ALLO

The **ALLO** command allocates a specified amount of disk space for a subsequent **STOR** or **APPE** command. When an **ALLO** command specifies an upper limit on the size of the file as stored in the MVS file system, a **STOR** that starts successfully is guaranteed not to fail because of disk space.

## Syntax Description

These are the allowable variations of the **ALLO** command syntax:

**ALLO R** *logical\_record\_length*                      This form is not included in the FTP specification but is a useful extension allowed by the Server FTP. A comma can replace **R**.

**ALLO** *byte\_count*

**ALLO** *byte\_count* **R**                                      A comma can replace **R**.  
*logical\_record\_length*

If an **ALLO** command is sent, the subsequent **STOR** or **APPE** commands operate with these MVS **SPACE** parameters in effect:

- $T = \text{floor}(\text{byte\_count} / \text{track\_length})$  (where T is the number of disk tracks needed)
- $S = \max(1, \text{floor}(T/5))$
- $\text{SPACE} = (\text{TRK},(S+T,S),\text{RLSE})$  for **STOR** or  $\text{SPACE}=(\text{TRK},(,S),\text{RLSE})$  for **APPE**

## Default

The default, if no **ALLO** is given, is **(5,3)** tracks, unless this default has been changed by your Cisco IOS for S/390 site administrator.

## Usage Guidelines

You can set the space parameters for creating a disk data set either explicitly with a **SITE SPACE(..)** command or implicitly with an **ALLO integer** command. If both commands are given, the **SITE SPACE** parameter takes precedence. If the **ALLO integer** is given after a **SITE SPACE(..)** command, the reply is “200 NOTE: Ignored, overridden by site space.”

**ALLO R** sets the LRECL value for a new data set. Once an **ALLO R** value has been set, a file received with a record longer than this limit is truncated rather than folded.

An **ALLO R** value makes sense only with a record-structured file. If the Server receives an **ALLO R** command when **STRU F** (file-structure) is specified, the **ALLO** command fails and returns the reply: “503 Command conflicts with previous commands.”

## HELP

The **HELP** command gives you introductory and reference information on the Server FTP. Output from the **HELP** command is delivered to you by the control connection; the output can be terminated by the Telnet Break facilities. Read Telnet Break.

This **HELP** syntax allows you to request **HELP** for a *command\_name* or *section\_title* option, but not with both. When no option is specified, general help information is given.

**HELP** [*command\_name* | *section\_title*]

### Syntax Description

*command\_name*      Command for which help is being requested; valid command-name strings are **ALLO**, **HELP**, **REST**, **SITE**, **STAT**, **STRU**, and **TYPE**.

*section\_title*      Title of a reference source of introductory or reference information provided through the control connection.

### Default

Use the **DEFAULT** string to request information on the default data set attributes (DCB parameters) created or used by the Server FTP.

### Usage Guidelines

Each MVS site supporting a Server FTP can provide additional help information beyond what is shown here. Valid *section\_title* strings for Cisco IOS for S/390 are:

- AECF | AECR | AENF | AENR | AETF | ILF | ILR
- Use these parameters to request help about Server FTP operation with various **TYPE** and **STRU** settings. Table 5-3 shows the **TYPE** and **STRU** settings corresponding to each parameter:

**Table 5-3      Using the TYPE and STRU Settings**

<b>HELP Parameters</b>	<b>Type</b>	<b>Stru</b>
<b>AECF</b>	<b>AC</b> or <b>EC</b>	<b>STRU F</b>
<b>AECR</b>	<b>AC</b> or <b>EC</b>	<b>STRU R</b>
<b>AENF</b>	<b>AN</b> or <b>EN</b>	<b>STRU F</b>
<b>AENR</b>	<b>AN</b> or <b>EN</b>	<b>STRU R</b>
<b>AETF</b>	<b>AT</b> or <b>ET</b>	<b>STRU F</b>
<b>ILF</b>	<b>I</b> or <b>L</b>	<b>STRU F</b>
<b>ILR</b>	<b>I</b> or <b>L</b>	<b>STRU R</b>

### INTRO

- Use the **INTRO** string to request an introduction to the use of the FTP Server.

### NEWS

- Use the **NEWS** string to request help on accessing Cisco IOS for S/390 news.



---

**PATH**

- Use the **PATH** string to request information on MVS path names (data set names, member names, and volumes) and their relationship to the Server FTP.

**SPACE**

- Use the **SPACE** string to request information on MVS space allocation in relationship to the Server FTP.

## MKD

The **MKD** (MaKe Directory) command creates a partitioned data set (PDS).

**MKD** *pathname*

### Syntax Description

*pathname*                    Path of the PDS to be created.

### Default

The special **GAT TYPE(LIBRARY)** statement (if present) overrides defaults for the **MKD** command.

### Usage Guidelines

The pathname can be either a fully or a partially qualified data set name.

These are the possible PDS file attributes:

- Cisco IOS for S/390 space allocation defaults are SPACE(5,3) DIR(5);
- The DEFGAT initialization statement can provide installation defaults
- Any **SITE** commands entered override any of the above

### Example

```
ftp> pwd
257 "'MVS.'" is current prefix
ftp> mkd mkd.pds
257-"MVS.MKD.PDS" partitioned dataset created with attributes:
Volser ICSPK1 Unit SYSALLDA Dsorg PO Recfm FB Lrecl 80
Blksize 6160 Space 5 15 Tracks Rlse Dir 46
257
ftp> mkd 'mvs.help.pds'
521 "'MVS.HELP.PDS'" data set already exists.
```

---

**Note** Read the *Cisco IOS for S/390 Customization Guide* for information about the **GAT** statements.

---

---

# REST

The **REST** (restart) command specifies that the data transfer command that follows immediately is to restart at a specified intermediate point in the file.

**REST** *restart\_marker*

## Syntax Description

*restart\_marker* Marker from which the restart is to begin.

## Default

The default interval is every 500,000 data bytes.

## Usage Guidelines

After a **REST** command, **STOR** and **APPE** have identical meanings (**APPE** is taken to mean **STOR**).

Data transfer must be in MODE B (block mode). The Server can send and accept restart markers in either **STRU F** or **STRU R**.

A file retrieved from the Server FTP includes restart markers at a specific interval. The **SITE** command **RESTART** option can change this interval or suppress restart markers entirely. When the count of bytes read from the disk since the last marker reaches the specified interval, a marker is sent at the next end of a complete logical record or segment of a spanned record.

## Restart markers

Table 5-4 sent by the Server FTP consist of twelve characters that are the ASCII representation of six bytes in the format *VTRBB*.

**Table 5-4 Restart Markers Sent by the Server FTP**

<b>Restart Marker</b>	<b>Description</b>
V	Volume sequence number
TTR	Standard MVS disk block address (referred to in IBM publications as a relative <i>track and record address</i> )
BB	A byte offset within a TTR block

## RMD

The **RMD** (ReMove Directory) command deletes an empty PDS. It will not delete a PDS that contains members. To delete a sequential file or a PDS containing members, use **DELE**.

**RMD** *path\_name*

### Syntax Description

*path\_name*      Name of the PDS to be deleted.

---

## SITE

The **SITE** command supplies host-dependent parameters to the Server FTP, for MVS data management controls, for special FTP controls, and for generic data set attributes.

**SITE** *parameter* [ , ]...

### Syntax Description

Available parameters include the following:

```
ATtr (gat_name)
AUTOIndex
AUTOMount | NOAUTOMount
AUTOREcall | NOAUTOREcall
BLKsize (max_physical_block)
BLocks
BLOCKSize (blocksize)
CARds | SORuce | FORtran | OBJect | LOadlib | PRINT
CD | NOCD
CHarset (table_name)
CHKptint (checkpoint_interval)
COmpact | NOCOmpact
CONDDisp (CATLG|DELETE)
CYlinder
DAclass (sms_data_class)
DATAClas (data_class_name)
DATASetmode
DBcsset (table_name)
DCBdsn (data_set_name)
DClose (data_port_close_time)
DEVNull
DIDle (data_port_idle_time)
DIRectory (blocks)
DIRECTORYMODE
DOpen (data_port_open_time)
DSeq (number)
DUMMY
Expdt (expiration_date) | RETPD (retention_period)
FILEtype (SEQ|JES|VTOC)
FORtran | CARds | SORuce | OBJect | LOadlib | PRINT
FULLtrk | Halftrk | VBs | VS
Halftrk | FULLtrk | VBs | VS
IBuf (numbuf bufsize)
IRBlksize (max_physical_block)
IRLrecl (logical_record_length)
IRRecfm (record_format)
ISPFEnq | NOISPFEnq
ISPFRes | NOISPFRes
JESLrecl (record_length)
JESRecfm (record_format)
LAbel (type)
LIne (logical_record_length)
LIStfmt (OLD | IBM | SHORT)
LKEDres | NOLKedres
LOadlib | CARds | SORuce | FORtran | OBJect | PRINT
LRecl (logical_record_length)
MAnagementclas (management_class_name)
MGmtclass (sms_management_class)
MIGratevol (migration_volume_serial)
MOunt (time)
NCP (number_of_DASD_buffers)
NDab (number1 number2)
NLstcase (UPPER | LOWER)
```

```

OBJect
OBUf(numbuf bufsize)
OVerwrite
PAD(char, char)
PARallelmount
PDse | NOPDse
PErsist | NOPErsist
POp
PREmount
PRImary
PRINT | LOadlib | CARds | SOurce | FORtran | OBJect
PRIVate
PUsh | POp
Qdisk(volume_serial_mask)
RDw | NORDw
RECall(integer) | NOREcall
RECFm(record_format)
RESEt
REStart(integer)
RETPd(retention_period) | Expdt(expiration_date)
RLse | NORLse
SECOndary
SOurce | CARds | FORtran | OBJect | LOadlib | PRINT
SPace(primary_allocation, secondary_allocation)
STClass(sms_storage_class_name)
STORclas(storage_class_name)
STRip | NOSTrip
SUBmit | NOSUBmit
TABs(integer)
TAPE
TErse
TRACkS
TRANopt(char_translation_mode)
UCNT(unit_count)
Unit(unit_name)
VCNT(volume_count)
VERbose
VOLUME(volume_name, volume_name, ...)
VSEQ(volume_sequence_number)
VBS | VS | FULLtrk | Halftrk
VS | VBS | FULLtrk | Halftrk
WRAPrecord | NOWRAPrecord

```

**record\_format = F | FB | FBS | FBA | FBSA | V | VB | VS | VBA | VBS | U**

### Site Command Parameters

These parameters are described in Table 5-4.

**Table 5-5 SITE Command Parameters**

Parameter	Description
<b>ATTR</b> ( <i>gat_name</i> )	Specifies any entry in the Generic Attributes table. Read Generic Attribute Names for details.  This command is not supported for the FTP to Tape Facility.
<b>AUTOINDEX</b>	Requests that the data set sequence number be increased by one for each subsequent file transfer.
<b>AUTOMOUNT   NOAUTOMOUNT</b>	<b>AUTOMOUNT</b> is an alias for <b>MOUNT</b> . <b>NOAUTOMOUNT</b> is an alias for <b>NOMOUNT</b> .

**Table 5-5 SITE Command Parameters (Continued)**

<b>AUTORECALL</b>   <b>NOAUTORECALL</b>	<b>AUTORECALL</b> is an alias for <b>RECALL</b> . <b>NOAUTORECALL</b> is an alias for <b>NORECALL</b> .
<b>BLKSIZE</b> ( <i>maximum_physical_block_length</i> ) <b>BLOCKSIZE</b> ( <i>maximum_physical_block_length</i> ) <b>LRECL</b> ( <i>logical_record_length</i> ) or <b>LINE</b> ( <i>logical_record_length</i> ) <b>RECFM</b> ( <i>record_format</i> )	Explicitly sets the <b>DCB</b> or format attributes of a new data set referenced by a <b>STOR</b> or <b>APPE</b> . If the data set is being created, these parameters override the defaults determined by the <b>FTP TYPE</b> or <b>STRU</b> commands. If the data set exists, these parameters must exactly match the corresponding attributes of the data set. <b>BLOCKSIZE</b> is an alias for <b>BLKSIZE</b> .
<b>BLOCK</b>	Space allocation is to be in blocks.
<b>CARDS</b>   <b>SOURCE</b>   <b>FORTRAN</b>   <b>OBJECT</b>   <b>LOADLIB</b>   <b>PRINT</b>	Specifies one of the standard Generic Attribute Names supplied with Cisco IOS for S/390. Read Generic Attribute Names for details.
<b>CD</b>   <b>NOCD</b>	<b>CD</b> enables directory commands ( <b>CWD</b> , <b>PWD</b> , <b>CDUP</b> ). <b>NOCD</b> disables directory commands.  Note The <b>CD</b>   <b>NOCD</b> parameter is not reset when data transfer begins.
<b>CHARSET</b> ( <i>table_name</i> )	Selects an alternate character set (translation) table for single-byte data transfer ASCII data. This table is validated for single-byte data.
<b>CHKPTINT</b> ( <i>checkpoint_interval</i> )	Specifies the number of logical records between restart markers.
<b>CONDDISP</b> ( <b>CATLG</b>   <b>DELETE</b> )	Specifies the conditional disposition for new data sets created by <b>STOR</b> or <b>APPE</b> when the file transfer fails.
<b>COMPACT</b>	Specifies <b>IDRC</b> compaction for 3480 tapes.
<b>CYLINDER</b>	Space allocation is to be in cylinders.
<b>DACLASS</b> ( <i>sms_data_class</i> )	Alias for <b>DATACLASS</b> .
<b>DATACLASS</b> ( <i>data_class_name</i> )	Specifies the SMS data class.
<b>DATASETMODE</b>	Requests the FTP server to display directory output ( <b>LIST/NLST</b> ) in data set mode. Each data set is listed individually.
<b>DBCSSET</b> ( <i>table_name</i> )	Selects an alternate character set (translation) table used for double-byte data transfer ASCII data. This table will be validated for double-byte data.
<b>DCBDSN</b> ( <i>data_set_name</i> )	Specifies a model data set for data set attributes.
<b>DCLOSE</b> ( <i>data_port_close_time</i> )	Specifies the time, in seconds, FTP will wait to close a data port.
<b>DEVNULL</b>	Requests that the FTP server allocate a dummy ( <b>NULLFILE</b> ) data set for storing a data set with the <b>STOR</b> command.
<b>DIDLE</b> ( <i>data_port_idle_time</i> )	Specifies the time, in seconds, FTP will wait on an idle data port. Minimum value = 60 seconds. Maximum value=1439 minutes.
<b>DIR</b> ( <i>blocks</i> )	Integer number of 256-byte blocks to be reserved for a <b>PDS</b> directory. One block holds from 7 (load module) to 16 (source module) member entries. This parameter is required to create a new <b>PDS</b> with a <b>STOR</b> or <b>APPE</b> command.

Table 5-5 SITE Command Parameters (Continued)

<b>DIRECTORYMODE</b>	<p>Requests that the FTP server display directory output (<b>LIST/NLST</b>) in directory mode. Data sets which have the same qualifier at the level immediately below the prefix level are grouped together as “pseudo-directories”.</p> <p>Example</p> <p>The following example illustrates the use of the <b>DIRECTORY MODE</b> parameter.</p> <pre>ftp&gt; quote site directorymode 200 OK, Ready ftp&gt; dir v* 125 List started OK. Volume Unit Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname ICS009 3390 03/19/96 1 1 VS  29389 29393 PS VS Pseudo Directory V191 Pseudo Directory V211 Pseudo Directory V311 Pseudo Directory V410 250 List completed successfully.</pre>
<b>DOPEN</b> ( <i>data_port_open_time</i> )	Specifies the time, in seconds, FTP will wait to open a data port.
<b>DSEQ</b>	Specifies data set sequence number.
<b>DUMMY</b>	This is an alias for <b>DEVNULL</b> .
<b>EXPDT   RETPD</b>	<p>Specifies expiration date and retention period.</p> <p><b>EXPDT</b>(<i>expiration_date</i>)—Specifies an expiration date for a new data set, in the format: <i>expiration_date</i> = <i>yyyddd</i> or <i>yyyy/ddd</i></p> <p>Note <i>yyyy</i> is a year from 1900 to 2155, <i>ddd</i> is a julian date from 1 to 366. You must include any leading zeroes in the <i>ddd</i> value.</p> <p><b>RETPD</b>(<i>retention_period</i>)—Specifies a retention period for a new data set <i>retention_period</i> is a number of days between 1 and 9999.</p> <p>Note <b>EXPDT</b> and <b>RETPD</b> are mutually exclusive.</p>
<b>FILETYPE</b> ( <b>SEQ</b>   <b>JES</b>   <b>VTOC</b> )	<p>Specifies the type of file the FTP server is working with:</p> <p><b>SEQ</b>—Sequential files. This command is the same as issuing a <b>SITE NOSUBMIT</b> command.</p> <p><b>JES</b>—<b>JES</b> spool. Data is written to the JES internal reader. <b>FILETYPE=JES</b> is the same as issuing a <b>SITE SUBMIT</b> command.</p> <p><b>VTOC</b>—<b>DASD VTOC</b> records. Directory commands list statistics from the Volume Table of Contents for DASD volumes.</p> <p>Default: <b>SEQ</b></p> <p>Note You cannot do a <b>GET</b> of spool files or display the <b>JES</b> spool queue with the <b>DIR</b> command.</p>
<b>FORTRAN</b>   <b>CARDS</b>   <b>SOURCE</b>   <b>OBJECT</b>   <b>LOADLIB</b>   <b>PRINT</b>	See the description for <b>CARDS</b> .
<b>FULLTRK</b>   <b>HALFTRK</b>   <b>VBS</b>   <b>VS</b>	Specifies general attributes for a data set.
<b>HALFTRK</b>   <b>FULLTRK</b>   <b>VBS</b>   <b>VS</b>	See the description for <b>FULLTRK</b> .
<b>IBUF</b> ( <i>numbuf bufsize</i> )	<p>Specifies in sublist notation the number of network input buffers (<i>numbuf</i>) and the buffer size (<i>bufsize</i>) to be used during data transfer. The maximum value for each number is 32767.</p> <p>Note Your Cisco IOS for S/390 system administrator might have set restrictions on use of the <b>IBUF</b> parameter.</p>



**Table 5-5 SITE Command Parameters (Continued)**

<b>IRBLKSIZE</b> ( <i>max_physical_block</i> )	Explicitly set the <b>DCB</b> or format attributes to be used to allocate the internal reader data set when <b>SITE SUBMIT</b> has been entered and a data transfer is performed.
<b>IRLRECL</b> ( <i>logical_record_length</i> )	
<b>IRRECFM</b> ( <i>record_format</i> )	
<b>ISPFENQ</b>   <b>NOISPFENQ</b>	Specifies that the <b>ISPF</b> enqueue facility be activated ( <b>ISPFENQ</b> ) or deactivated ( <b>NOISPFENQ</b> ). Default: <b>NOISPFENQ</b>
<b>ISPFRES</b>   <b>NOISPFRES</b>	Enables ( <b>ISPFRES</b> ) or disables ( <b>NOISPFRES</b> ) the <b>RESERVE</b> logic for the <b>SPFEDIT ENQ</b> , if the volume on which the <b>PDS</b> resides is shared by Multiple Systems (UCB shared bit ON). This assures data integrity while the <b>PDS</b> you are accessing is being simultaneously accessed by an <b>ISPF</b> user from another system. Default: <b>NOISPFRES</b>
<b>JESLRECL</b> ( <i>record_length</i> )	Alias for <b>IRLRECL</b> .
<b>JESRECFM</b> ( <i>record_format</i> )	Alias for <b>IRRECFM</b> .
<b>LABEL</b>	Specifies label type. Label options supported are: <b>SL</b> , <b>NL</b> , <b>BLP</b> , <b>LTM</b> , <b>AL</b> . See <b>LABEL</b> parameter description in the description of the <b>GAT</b> statement in the <i>Cisco IOS for S/390 Customization Guide</i> .
<b>LKEDRES</b>   <b>NOLKEDRES</b>	Enables ( <b>LKEDRES</b> ) or disables ( <b>NOLKEDRES</b> ) the <b>RESERVE</b> logic for the <b>SYSIEWLP ENQ</b> , if the volume on which the <b>PDS</b> resides is shared by Multiple Systems (UCB shared bit ON). This assures data integrity while the <b>PDS</b> you are accessing is being simultaneously accessed by the linkage editor from another system. Default: <b>NOLKEDRES</b>
<b>LINE</b> ( <i>logical_record_length</i> )	See <b>BLKSIZE</b> .
<b>LISTFMT</b> ( <b>OLD</b>   <b>IBM</b>   <b>SHORT</b> )	Specifies whether output from the data set <b>LIST</b> command will be in the old Cisco IOS for S/390 format, in IBM-standard format, or in a shortened IBM-compatible format. The short format leaves out data set extents and tracks allocated, but improves <b>LIST</b> response time. Default: <b>SHORT</b> Note Certain PC-based client FTP packages expect the <b>LIST</b> output from a host configured as OS/MVS to be in standard IBM format.  The <b>LIST</b> parameter is not reset when data transfer begins.
<b>LOADLIB</b>   <b>CARDS</b>   <b>SOURCE</b>   <b>FORTTRAN</b>   <b>OBJECT</b>   <b>PRINT</b>	See the description for <b>CARDS</b> .
<b>LRECL</b> ( <i>logical_record_length</i> )	See <b>BLKSIZE</b> .
<b>MANAGEMENTCLASS</b> ( <i>management_class_name</i> )	Specifies the <b>SMS</b> management class.
<b>MGMTCLASS</b> ( <i>sms_management_class</i> )	Alias for <b>MANAGEMENTCLASS</b> .
<b>MIGRATEVOL</b> ( <i>migration_volume_serial</i> )	The volume serial number of migrated data sets.
<b>MOUNT</b> ( <i>time</i> )	Enables tape support ability for this file transfer. The time value specifies the maximum wait time in minutes for the tape mount to complete. If the time expires, the request is aborted. <b>MOUNT</b> without a time value specified will use the default system wait time.
<b>NCP</b> ( <i>number_of_DASD_buffers</i> )	Alias of <b>NDAB</b> .

Table 5-5 SITE Command Parameters (Continued)

<b>NDAB</b> ( <i>number1 number2</i> )	Specifies the number of <b>DASD</b> buffers used by FTP for reading or writing disk data sets. The maximum value is 99.  Default: 4  Note Your Cisco IOS for S/390 administrator might have set restrictions on the use of the NDAB parameter.
<b>NLSTCASE</b> ( <b>UPPER</b>   <b>LOWER</b> )	Specifies whether the output from an <b>NLST</b> command will be upper- or lowercase. If <b>LOWER</b> is specified and the data set or member list is part of the current directory, the names will be returned in lowercase.  Note <b>NLSTCASE(LOWER)</b> is supplied to facilitate <b>MGET</b> functions from FTP clients on systems that use lowercase file names.  Note This parameter is not reset when data transfer begins.
<b>OBJECT</b>   <b>LOADLIB</b>   <b>CARDS</b>   <b>SOURCE</b>   <b>FORTRAN</b>   <b>PRINT</b>	See the description for <b>CARDS</b> .
<b>OBUF</b> ( <i>numbuf bufsize</i> )	Specifies, in sublist notation, the number of network input buffers ( <i>numbuf</i> ) and the buffer size ( <i>bufsize</i> ) to be used during data transfer. The maximum value for each number is 32767.  Note Your Cisco IOS for S/390 system administrator might have set restrictions on use of the OBUF parameter.
<b>OVERWRITE</b>   <b>NOOVERWRITE</b>	This parameter is a toggle. <b>OVERWRITE</b> requests that the FTP server overwrite an existing data set when transferring files if a data set of the same name already exists on the target server. This parameter is necessary when the <b>SITEOVERWRITE</b> configuration parameter is in effect.  Note This parameter is reset after each transfer, regardless of the <b>PERSIST</b> option in effect.
<b>PAD</b> ( <i>pad_code</i> )	Overrides the default characters that pad network records or lines to fixed-length logical records when data is stored (via <b>STOR</b> or <b>APPE</b> ) or deleted from fixed-length logical records when data is retrieved (via <b>RETR</b> ). The defaults are blanks for character types and zeros for binary types. These are the pad-codes:  Z – Pad with zeros O – Pad with ones B – Pad with blanks
<b>PARALLELMOUNT</b>	Specifies parallel mounting (mutually exclusive with <b>UCNT</b> ).
<b>PDSE</b>   <b>NOPDSE</b>	Allocates <b>PDSEs</b> instead of <b>PDSs</b> , or vice versa.
<b>PERSIST</b>   <b>NOPERSIST</b>	Specifies whether <b>SITE</b> parameters will be reset following data transfer. If <b>PERSIST</b> is used, all <b>SITE</b> parameters remain in effect until explicitly changed via subsequent <b>SITE</b> commands, or reset with <b>SITE RESET</b> . If <b>NOPERSIST</b> is used, all <b>SITE</b> parameters are reset after each data transfer.  Default: <b>NOPERSIST</b>
<b>PRIMARY</b> ( <i>primary_allocation</i> )	Specifies the primary space allocation for data sets.
<b>PRINT</b>   <b>LOADLIB</b>   <b>CARDS</b>   <b>SOURCE</b>   <b>FORTRAN</b>	See the description for <b>CARDS</b> .
<b>PRIVATE</b>	Requests a private volume.
<b>PUSH</b>   <b>POP</b>	<b>PUSH</b> causes the server FTP to save the current settings of parameters entered with previous <b>SITE</b> commands; <b>POP</b> restores those parameters.  Note <b>PUSH</b> and <b>POP</b> can be nested. Each <b>PUSH</b> adds an entry to a push-down stack. Each <b>POP</b> pulls the last entry from the stack. When there are no entries in the stack, a <b>POP</b> will result in an error reply.

**Table 5-5 SITE Command Parameters (Continued)**

<b>QDISK</b> ( <i>volume_serial_mask</i> )	Requests the FTP server to provide volume information for the volume(s) specified in the <i>volume_serial_mask</i> .  Example <pre>ftp&gt; quote site q=ics001 200-          %      Free Free Largest Free 200- Volume Unit Free Cyls Trks Cyls-Trks Exts Address Use Attr 200- ICS001 3390  1  23  102 15  14  23   420  Storage 200 Site command was accepted</pre>
<b>RDW   NORDW</b>	Specifies whether <b>RDWs</b> (Record Descriptor Words) will be sent as data for <b>RECFM=VB</b> and <b>RECFM=VBS</b> files. If <b>RDW</b> is selected, the <b>RDW</b> will be sent for binary, <b>ASCII</b> , or <b>EBCDIC</b> transfers.  Default: <b>NORDW</b>  Note <b>RDW</b> will not be translated for ASCII file transfers. This parameter is ignored for data sets with carriage control ( <b>RECFM=VBA</b> or <b>RECFM=VBM</b> ).
<b>RECALL</b> ( <i>integer</i> )   <b>NORECALL</b>	<b>RECALL</b> —Enables Hierarchical Storage Manager (HSM) recall ability for this file transfer. The integer value specifies the maximum wait time in minutes for HSM to complete the recall of the migrated file. If the time expires, the request is aborted. <b>RECALL</b> without an integer value specified enables HSM with the default system wait time.  <b>NORECALL</b> —Disables HSM recall ability for this file transfer.
<b>RECFM</b> ( <i>record_format</i> )	See <b>BLKSIZE</b> .
<b>RESET</b>	Resets all previous <b>SITE</b> commands. Can be used if <b>SITE PERSIST</b> is specified.
<b>RESTART</b> ( <i>integer</i> )	When a file is <b>RETR</b> ieved in block mode, the FTP Server includes a Restart Marker in the data stream every integer data bytes. To suppress these markers entirely, specify <b>RESTART(0)</b> . The default is <b>RESTART(500,000)</b> . Read <b>REST</b> for information about Restart processing.  This command is not supported for the FTP to Tape Facility.
<b>RETPD   EXPDT</b>	See the description for <b>EXPDT</b> .
<b>RLSE   NORLSE</b>	<b>RLSE</b> —Cancels a previous <b>SITE NORLSE</b> . This is needed occasionally to prevent building up too many extents when many <b>APPE</b> operations to the same data set are performed.  <b>NORLSE</b> —Specifies that unused disk space not be released following a <b>STOR</b> or <b>APPE</b> .  Default = <b>RLSE</b>
<b>SECONDARY</b> ( <i>secondary_allocation</i> )	Specifies the secondary space allocation for data sets.
<b>SOURCE   CARDS   FORTRAN   OBJECT   LOADLIB   PRINT</b>	See the description for <b>CARDS</b> .
<b>SPACE</b> ( <i>primary_allocation, secondary_allocation</i> )	Specifies primary and secondary disk space allocation (the default allocation is in tracks). Only required for <b>STOR</b> or <b>APPE</b> commands.
<b>STCLASS</b> ( <i>sms_storage_class_name</i> )	Alias for <b>STORCLASS</b> .
<b>STORCLAS</b> ( <i>storage_class_name</i> )	Specifies the <b>SMS</b> storage class.
<b>STRIP   NOSTRIP</b>	Specifies whether pad characters will be stripped from fixed-length logical records when data is retrieved (using <b>RETR</b> ).
<b>SUBMIT   NOSUBMIT</b>	<b>SUBMIT</b> —Specifies the MVS FTP Server to send the output of a file transfer to a <b>JES</b> internal reader for execution.  <b>NOSUBMIT</b> —Cancels a previous <b>SITE SUBMIT</b> .

Table 5-5 SITE Command Parameters (Continued)

<b>TABS</b> ( <i>integer</i> )	Specifies the tab stop interval to be used in receiving the next file. The default is 8, and the limit is 25. <b>TABS</b> (0) translates tab characters (for example, ASCII x'09' is translated to EBCDIC x'05'); <b>TABS</b> (1) replaces each tab character with a blank.
<b>TAPE</b>	Specifies that attributes are to be taken from the <b>GAT TYPE(TAPE)</b> entry.
<b>TERSE</b>	Requests the FTP server to issue single-line 150 and 226 replies.
<b>TRACKS</b>	Requests space allocation in tracks.
<b>TRANOPT</b> ( <i>char_translation_mode</i> )	Defines the character translation mode. These are the choices: <b>CHAR</b> —Defines character translation mode as single-byte. <b>DBCS</b> —Defines character translation mode as double-byte. <b>MIX</b> —Defines character translation mode as single-byte and/or double-byte.
<b>UCNT</b> ( <i>unit_count</i> )	Specifies the maximum number of generic units an output data set can require. A value of 1 to 59 can be entered.
<b>UNIT</b> ( <i>unit_name</i> )	Specifies a generic unit for creating a new data set.
<b>VBS   VS   FULLTRK   HALFTRK</b>	See the description for <b>FULLTRK</b> .
<b>VCNT</b> ( <i>volume_count</i> )	Specifies the maximum number of volumes an output data set can require. A value of 1 to 255 can be entered.
<b>VERBOSE</b>	Requests the FTP server to issue multi-line 150 and 226 replies, showing data set attributes and transfer statistics.
<b>VOLUME</b> ( <i>volume_name, volume_name, ...</i> )	Specifies an explicit volume(s) for creating a new data set or referencing an old uncataloged data set. Normally not required. A total of 255 can be entered. If <b>VOLUME</b> is entered with no <i>volume_name</i> , all <b>VOLUME</b> information is reset.
<b>VS   VBS   FULLTRK   HALFTRK</b>	See the description for <b>FULLTRK</b> .
<b>VSEQ</b> ( <i>volume_sequence_number</i> )	Specifies that processing should begin at a requested volume within a multi-volume data set. A value of 1 to 255 can be entered.
<b>WRAPRECORD   NOWRAPRECORD</b>	Network records which exceed <b>LRECL</b> will be wrapped to the next record when receiving data.

### Usage Guidelines

**SITE** command keywords are evaluated in left to right order. Successive **SITE** commands are cumulative. A later **SITE** command can add to or change attributes established by an earlier **SITE** command.

**SITE** command parameters can be entered in either PL/I or BAL formats. The examples for FTP3 are shown in PL/I format. Any **SITE** parameter that takes a keyword can be entered in either format.

If **NOPERSIST** is used, all **SITE** parameters are reset after each data transfer. If **PERSIST** is used, all **SITE** parameters remain in effect until explicitly changed via subsequent **SITE** commands, or reset with **SITE RESET**.

When a sub-parameter takes a list and more than one value is contained in the list, the list of values must be enclosed in parenthesis. Even if you choose to enter the command in BAL format, you must use parentheses around the list.

The **SITE** command verb is followed by a list of keyword parameters. Each keyword may normally be abbreviated to the minimum-length unambiguous string, as in TSO. (In the preceding list, the minimum abbreviation for each parameter is shown in upper-case.)

**SITE** parameters must be separated by a comma or a blank.

---

A single FTP command is limited to 80 characters. In the (unlikely) event that a **SITE** command exceeds 80 characters, it can be broken into two or more successive **SITE** commands.

If an error is found in parsing a **SITE** parameter, an error message is issued indicating the bad parameter, and the FTP server continues with the next parameter.

### Examples

The following example illustrates the BAL format:

**SITE ATTR**=*gat\_name*

The following example illustrates the PL/I format:

**SITE ATtr** (*gat\_name*)

The following example illustrates the obligatory use of parentheses when the command is entered in BAL format:

**SITE VOLUME**=(*vol1, vol2, vol3*)

## STAT

The **STAT** command provides partial or complete status of the Server FTP.

### STAT

#### Syntax Description

<i>selector</i>	A string containing any subset of the letters <b>F   A   P   T   I</b> <ul style="list-style-type: none"><li>• <b>F</b>—FTP parameters, such as <b>MODE</b> and <b>TYPE</b>.</li><li>• <b>A</b>—Access control, such as user ID and account.</li><li>• <b>P</b>—Path data, such as data set name and attributes.</li><li>• <b>T</b>—Data transfer status (such as number of bytes, records transmitted); null if no transfer is in progress</li><li>• <b>I</b>—Internal control blocks of Server FTP.</li></ul>
<i>item_numbers</i>	One or more positive integers separated by commas.
<i>path_name</i>	A valid MVS directory identifier optionally enclosed in quotes.

#### Usage Guidelines

The commands **STAT** or **STAT ?** mean **STAT ? FAPT**.

The command **STAT ? item\_numbers** gives only the specified item of the full **STAT** display. This is useful in debugging since the full display can be lengthy.

The command **STAT path\_name** (where *path\_name* is a valid MVS data set prefix specified in the form *myuid.*) gives catalog information on a specific group of data sets. **STAT path\_name** gives the same information as **LIST** but sends it over the Telnet control connection instead of a data transfer connection.

The command **STAT \*** gives the catalog list for the default (logged-in) directory.

#### Example

**STAT ?**

**STAT ? selector**

**STAT ? item\_numbers**

**STAT path\_name**

#### Data Set Attributes

This section describes the attributes of data sets that can be retrieved (read) or stored (written) by the Server FTP.

#### Units and Volumes

The Server FTP can write and retrieve only disk data sets stored on permanent-resident disk volumes.

---

## Data Set Names

The Server FTP rules for naming and accessing data sets described here are the same as those for TSO.

User disk data sets generally have names of the form:

```
defprfx.name1[.name2[.name3 .. [.namen]]]
```

## Syntax Description

<i>defprfx</i>	That portion of the data set name that is the defined default prefix of the installation.
<i>namex</i>	Data set name indexes, made up of 1 to 8 alphanumeric characters, the first of which is alphabetic.

---

**Note** The total length of the data set name, including the periods between indexes, cannot exceed 44 characters.

---

Data set naming conventions can vary between MVS sites. Consult personnel at your site to learn naming conventions.

When the data set name is enclosed in single quotes, it is a fully qualified data set name. When the data set name is not enclosed in quotes, it is partially qualified. Under this Server FTP (and TSO), you normally specify data set names in a partially qualified fashion, allowing the system to prefix the installation's default prefix to the data set name.

Server FTP uses one of the three possible default prefixes:

- User ID
- none
- character strings.

If User ID is defined at your installation, then server FTP uses the User ID provided by the **USER** FTP command as the *defprfx* for prefixing. If no prefixing is defined, the data set is fully qualified and quotes are not required. If character string is defined, the installation selected a common qualifier for all data sets. Individuals can select their own data set prefix with the Change Working Directory (**CWD**) or Change to Parent Directory (**CDUP**) FTP commands.

## Example

These are some examples of how to set your prefix. The first method shows the **cd** and **cdup** User commands; the second shows sending the **cwd** and **cdup** commands to the server using **quote**.

```
ftp > pwd
257 "'MYID.'" is current prefix.
ftp > cd level1
250 "'MYID.LEVEL1.'" is current prefix.
ftp > cdup
200 "'MYID.'" is current prefix
ftp > cd level2
250 "'MYID.LEVEL2.'" is current prefix.
ftp > cd 'newid'
250 "'NEWID.'" is current prefix.
ftp > quote cwd 'nextid'
250 "'NEXTID.'" is current prefix.
ftp > quote cdup
200 No prefix defined.
```

The Server FTP supports both simple sequential and partitioned data sets. A PDS contains an internal directory to a set of sub-data sets called members. All members of a PDS share the same data set name (*dsname*) and attributes. The fully qualified name of a PDS member is:

```
defprfx.name1[.name2[.name3 ... [.namen]]](member_name)
```

The *member\_name* field has the same syntax as *name1* through *namen*.

The Server FTP also supports Generation Data Group (GDG) data sets. A GDG data set has a similar format to a PDS but the member name takes the form 0, *+n*, or *-n*, where *n* is the relative generation number.

## FTP Path Name Syntax

The data transfer commands **STOR**, **APPE**, **RETR**, **DELE**, **RNFR**, and **RNTO** have *path\_name* as a parameter. With this Server FTP, *path\_name* specifies a data set or PDS member in one of these forms:

- *name1[.name2[.name3 ... [.namen]]]*
- *name1[.name2[.name3 ... [.namen]]](member\_name)*
- *name1[.name2[.name3 ... [.namen]]](GDG\_number)*
- *'name1[.name2[.name3 ... [.namen]]]'*
- *'name1[.name2[.name3 ... [.namen]]](member\_name)'*
- *'name1[.name2[.name3 ... [.namen]]](GDG\_number)'*

The first three forms are partially qualified data set names. An installation defined default prefix is prefixed to names in these forms. Typically, you use one of these forms to refer to your own data sets. The last three forms are fully qualified and are used as is by the FTP Server. The second and fourth forms are used for PDS libraries and the third and sixth forms are used for GDGs.

In addition, a *member\_name* can be entered alone when the current directory is a partitioned data set. Read VTOC and Catalog.

---

**Note** The length of the 1- to 8- character member name or the GDG number and the enclosing parentheses are not included in the 44-character limit.

---



---

## Using Wildcard Characters in FTP

The server commands **STAT**, **LIST**, and **NLST** accept a data set or member name mask as a parameter. The output from the command lists all data sets or members that match the mask criteria.

These are the rules for masking:

- An asterisk (\*) represents zero or more consecutive characters
- A percent sign (%) represents a single character

---

**Note** If a member name or member name mask is included, the data set name must not include a mask.

---

### Example

```
ftp> ls v*.obj
200 OK, Ready
125 Transfer started
V111.OBJ
V20.OBJ
V201.OBJ
226 Transfer complete
ftp> ls v2%.obj
200 OK, Ready
125 Transfer started
V20.OBJ
226 Transfer complete
ftp> cd v20.obj
250 "'MVS.V20.OBJ'" partitioned data set is current directory
ftp> ls ftps*
200 OK, Ready
125 Transfer started
FTPS
FTPSFTDR
226 Transfer complete
ftp> ls v*.obj (*)
200 OK, Ready
501 Wildcard characters are not permitted within a Partitioned dataset name
```

## Partitioned Data Sets

When a **CWD** or **CDUP** command causes the prefix to match the data set name of a cataloged PDS, the PDS becomes the working directory. Subsequent data transfer commands **STOR**, **RETR**, **DELE**, **RNFR**, and **RNTO**, as well as the **LIST**, **NLST**, and **STAT** commands, will treat unquoted path names as PDS member names. In addition, the **LIST** and **NLST** commands without a path name cause a list of members for the PDS directory to be output.

## Examples

```
ftp> pwd
257 "'MVS.'" is the current prefix
ftp> cd help.pds
250 "'MVS.HELP.PDS'" partitioned data set is current directory
ftp> dir
200 OK, Ready
125 Transfer started
  Name      VV.MM  Created   Changed   Size  Init  Mod  Id
FTPDEFAU   04.00  12/08/93  12/09/93   6:16   98   98  0 MVS
FTPINTRO   01.00  12/09/93  12/09/93   6:18  134  134  0 MVS
FTPNEWS    01:00  12/09/93  12/09/93  10:30   25   25  0 MVS
GREETING   01:00  12/08/93  12/09/93   6:19   12   12  0 MVS
226 Transfer complete
ftp> get ftpnews
200 OK, Ready
150-Dataset open with attributes:
Type A N Tabs 8 Stru F Mode S Path MVS.HELP.PDS(FTPNEWS)
Volser MVSVOL Unit SYSALLDA Dsorg PO Recfm FB Lrecl 80
Blksize 3120 Rlse
150
226 Transfer complete
```

In addition, the **LIST**, **NLST**, and **STAT** commands will accept member name masks as path names. The member name mask can be entered either alone (if the current directory is a PDS) or as part of a fully qualified PDS data set name.

```
ftp> ls 'mvs.help.pds(ftp*)'
200 OK, Ready
125 Transfer started
'MVS.HELP.PDS(FTPDEFAU)'
'MVS.HELP.PDS(FTPINTRO)'
'MVS.HELP.PDS(FTPNEWS)'
226 Transfer complete
```

If you want to treat a PDS data set path name as a prefix, enclose the fully-qualified name in quotes and append a period (.) at the end.

```
ftp> cd 'mvs.help.pds.'
250 "'MVS.HELP.PDS.'" is current prefix
```

## VTOC and Catalog

Under MVS, each disk volume contains a file directory of the data sets on that volume called the volume table of contents (VTOC). Hence, any disk data set can be located via the logical path name: *volume,dsname*. The Server FTP lets you specify the volume name in the **SITE** command. Read **SITE** for details.

MVS has a central file directory, called the catalog, that provides the volume name as a function of the *dsname*. TSO generally requires that all user data sets except scratch files be cataloged (in other words, listed in a system catalog). A cataloged data set name is unique on the system while an uncataloged data set name need be unique only on the disk volume where it resides. You need to give only the *dsname* (and *member\_name* for a PDS), not the volume, to locate an existing cataloged data set.

This Server FTP generally does not change the catalog status of a data set operated on by a **STOR**, **APPE**, **RETR**, or rename operation. A data set created by an FTP **STOR**, **APPE**, or **RNTO** operation is cataloged. If the *dsname* conflicts with an existing cataloged data set, the operation is refused in the case of **RNTO**.

---

When the Server FTP performs an **APPE** or **RETR** operation on an existing uncataloged data set (using **SITE VOLUME**), it tries to catalog it. If the attempt fails because another data set with the same name is being cataloged or because the *dsname* has the wrong tree structure, a warning message is sent but the operation proceeds.

## MVS Space Allocation

MVS allocates disk space to a data set in variable-sized pieces called extents. Each extent is a contiguous set of disk tracks. The byte capacity of a disk track depends on the disk model. For example, A 3380 holds a maximum of 47476 bytes per track. A data set used by the Server FTP is limited to a maximum of 16 extents. Therefore, if the disk space is fragmented, you might run out of extents before running out of total space. In such a situation, choice of appropriate space parameters is important.

Space allocation may have two parameters to the operating system: a primary space quantity, and a secondary space quantity. When an FTP **STOR** operation is requested, FTP first tries to allocate the primary quantity. If it succeeds, data transfer starts. Each time that space is exhausted during the **STOR** (or **APPE**) operation, FTP tries to allocate the secondary quantity. The **STOR** (or **APPE**) fails at that point if it cannot satisfy the request. Each allocation, whether primary or secondary, takes place as follows:

- MVS tries to find a contiguous area (extent) to satisfy the request using the best-fit algorithm
- If no single area is large enough, it uses the fewest extents possible (up to five) to satisfy the request

This allocation process continues until the total space or the extent limit is exhausted.

The default space parameters for FTP **STOR** are (primary, secondary) = (5,3) tracks. The maximum space that can be allocated with these parameters depends on the degree of fragmentation of the volume (assuming that enough total space exists). It ranges from 50 tracks down to 14 tracks (if the largest contiguous area is only 1 track and extents are exhausted first).

These methods obtain more space or a larger file:

- Send an **ALLO** command before the **STOR**.

The parameter to **ALLO** is a file size in (network) bytes. The Server FTP converts that value to disk tracks and uses 1.2 times that value as the initial space allocation. Therefore, if the data transfer starts, you know the initial space allocation was successful and you cannot run out of space specified by the **ALLO**. The 1.2 factor is intended to take care of inter-record gaps. The secondary space quantity is 0.2 times the **ALLO** value.

- Specify explicit **SPACE** parameter on the **SITE** command.

The disk space allocation (**ALLO**) is in terms of the data stored on disk, including padding. A text file normally is padded to 80 byte records, so each line is 80 bytes on disk. The primary and secondary space quantities are recorded in the data set label (**DSCB**) by the operating system. A subsequent **APPE** uses the secondary quantity determined when the data set was created unless you override it with a new **ALLO** or **SITE SPACE** before the **APPE**.

## Multivolume Data Sets

If a data set might require more space than is available on a single volume, you can specify that it may reside on multiple volumes using the **SITE VOLUME**, **VCNT**, or **UCNT** parameters. Enough space must exist on the first volume for the primary extent. When the space on the first volume is exhausted, extents will be allocated on the next volume. Up to 16 extents can be allocated on each volume.

FTP will use the greatest of the following numbers to determine how many devices and volumes to allocate to a data set:

- Unit-count specified in the **SITE UCNT** parameter
- Volume-count specified in the **SITE VCNT** parameter
- Number of serial numbers specified in the **SITE VOLUME** parameter

## Data Set Organization

FTP supports sequential (**DSORG=PS**) and partitioned (**DSORG=PO**) data sets. Direct access (**DSORG=DA**) data sets can be read sequentially but cannot be written by the Server FTP. ISAM and VSAM data sets are not supported.

## Disk Format (DCB) Attributes

A data set under MVS has the DCB attributes listed in Table 5-6; the first column shows the **SITE** command keyword parameter(s) to set the corresponding DCB attribute:

**Table 5-6 Disk Format Attributes**

<b>Keyword</b>	<b>Corresponding data set (DCB) attribute</b>
<b>RECFM</b>	Record Format ( <b>RECFM</b> )
<b>LRECL</b> or <b>LINE</b>	Logical Record Length ( <b>LRECL</b> )
<b>BLKSIZE</b> or <b>BLOCK</b>	Physical Block Length ( <b>BLKSIZE</b> )

When an existing data set is retrieved, the Server FTP determines the attributes from MVS and uses them to read the disk data set correctly. However, when writing to a data set, either through a **STOR** or **APPE** operation, you may need to be concerned about setting the correct attributes. Whether Server FTP assigns attributes for an existing data set depends on whether the data set is partitioned (PDS) or sequential.

### Sequential Data Sets

A **STOR** or an **APPE** to a non-existent data set creates a new data set. The Server FTP assigns the attributes of the new data set. Additionally, the **STOR** operation assigns new attributes to an existing data set.

New attributes are set from default attributes chosen by the Server FTP based on the **TYPE** and **STRU** transfer parameters (read Default Data Set Attributes), or from explicit overrides of these defaults by **SITE** parameters.

### Partitioned Data Sets

A **STOR** of a member into an existing PDS adds information to the data set (similar to an **APPE**). The attributes of the data set do not change. Any attributes set by the **SITE** command must match those of the existing PDS, or the Server FTP responds with this message:

```
554 SITE LRECL, BLKSIZE or RECFM do not match those of existing data set
```

An **APPE** of a new PDS member is identical to a **STOR**. However, you cannot **APPE** into an existing member because the file system replaces the member. If you attempt to **APPE** into an existing member name, Server FTP responds with this message:

```
504 Not implemented for that parameter, ignored.
```

If a new PDS is being created, either by **STOR** or **APPE**, new data set attributes are assigned by the rules in Sequential Data Sets.

## Default Data Set Attributes

The Server FTP chooses default data set attributes for **STRU** based on the **TYPE** and chooses **STRU** transfer parameters as shown in Table 5-7:

**Table 5-7** Default Data Set Attributes

Type	STRU F	STRU R
AN or EN	SOURCE	SOURCE
AT or ET	PRINT	PRINT
AC or EC	PRINT	PRINT
I or L	VS	VBS

**Note** In the **STRU R** case, the default data set attributes depend on the size passed by the **ALLO** command. If the **ALLO** command has not been specified or if **ALLO R** is less than 81, the default data set attribute is **SOURCE**. Otherwise (**ALLO R** is greater than 80), the default data set attribute is **VBS**, with the **LRECL** set to **ALLO R** plus 4.

## Generic Attribute Names

You can explicitly provide a subset of DCB attributes and DD statement fields on a **SITE** command. Alternatively, the **SITE** command can specify one of the generic attribute names given in Table 5-8:

**Table 5-8** Generic Attribute Names

Attribute name	RECFM	LRECL	BLKSIZE
FULLTRK	FB	80	<b>full-track</b> (see Note 1 following this table)
HALFTRK	FB	80	half-track (see Note 1 following this table)
LOADLIB	U	0	rcmd (see Note 2 following this table)
OBJECT	FB	80	rcmd (see Note 2 following this table)
PRINT	VBA	133	rcmd
SOURCE CARDS FORTRAN	FB	80 (default)	rcmd
VBS	VS	rcmd-4	rcmd
VS	VS	rcmd-4	rcmd

**Note** Actual values for half and full track blocking depend on the output disk device type. These generic types create PDS libraries suitable for object modules and load modules.

Each generic attribute name includes values for RECFM, LRECL, and BLKSIZE, and optionally, UNIT, VOLUME, or SPACE, but the attributes set by a generic attribute name can be overridden by other generic or specific attribute keywords. When overriding **SITE** command keywords, the keywords are interpreted in left-to-right order.

In addition, other generic attribute names can be defined by your site. These names can stand for other combinations of RECFM, LRECL, BLKSIZE, UNIT, VOLUME, and SPACE. User-defined names must specify RECFM, LRECL, and BLKSIZE. They can optionally specify UNIT, VOLUME, or SPACE parameters.

The generic types SOURCE, CARDS, FORTRAN, OBJECT, LOADLIB, and PRINT can also carry UNIT, VOLUME, and SPACE parameters as defined by your site. They can be referenced by the **SITE** keywords SOURCE, CARDS, FORTRAN, OBJECT, LOADLIB, and PRINT. Other generic types may also have been defined by your site. They can be referenced by the **SITE** keyword ATTR (type).

In the preceding table, the block size is sometimes specified as rcmd, meaning a recommended value. This means the Server FTP chooses an actual default BLKSIZE that is optimum for the particular device on which the data set is allocated and less than or equal to a recommended size. The recommended size is a site-specific FTP parameter. If your site has not changed this parameter, the recommended size is 6K (6144) bytes. The choice of optimum BLKSIZE <=<rcmd depends on RECFM, LRECL, and the disk device type.

## Rules for Record Formats

These rules govern Server FTP support of the record format (RECFM) data set (DCB) attribute:

- The Server FTP writes into a disk data set using any of the record formats (RECFM) listed in Table 5-9:

**Table 5-9 Record Formats**

Record format	Description
U, F, V, VS	Unblocked
FB, FBS, VB, VBS	Blocked, nonprint format
FBA, FBSA, VBA	Blocked, print format

- The Server FTP does not support storage or retrieval of data sets with the T (record overflow) or M (machine carriage control) attributes. However, it does support VBM in binary mode.
- These unusual and unblocked print format RECFMs can be read but not written by the FTP Server:  
 FA, FSA, VA, VSA, UA
- When the TYPE is I or L (binary data), the data set RECFM cannot specify print format (A). A print data set can be created or retrieved only as text, not as binary data.

Violation of any of these rules results in a 554 illegal recfm error reply.

---

## Data Set Attribute Errors

These conflicting data set attributes create the following error reply and prevent the operation:

```
501 LRECL or BLKSIZE invalid or inconsistent
```

- RECFM ("V.."): BLKSIZE < 4
- RECFM ("VB" or "VBA"): LRECL > BLKSIZE-4 or LRECL < 8
- RECFM ("FB"): LRECL > BLKSIZE or BLKSIZE not an integer multiple of LRECL

In addition, for unblocked RECFMs the LRECL is forced to do the following:

- RECFM ("V"): BLKSIZE-4
- RECFM ("F"): BLKSIZE

## Appending to Empty Data Sets

These rules apply to append (**APPE**) operations performed to empty data sets.

- If the data set to be appended to has RECFM=0 or BLKSIZE=0, it is assumed to be empty and is scratched and recreated using the new attributes.
- If the data set is not empty but has LRECL=0 and blocked RECFM, Server FTP uses the new LRECL (the default) or the LRECL from **SITE**.
- Otherwise, Server FTP uses DCB parameters of the existing data set.

## JES Internal Reader Support Procedures

These rules apply to the Server FTP **SUBMIT** operation:

- Connect and log on to the host where the JCL resides.
- Connect and log on to your IBM host where Cisco IOS for S/390 is running.
- Use a **QUOTE** type command to send a **SITE SUBMIT** command to the IBM host (such as, **QUOTE SITE SUBMIT**).
- Issue a **PUT**, **GET**, or **SEND** command of the data set on the host where the JCL resides to any name on the IBM host (such as, **PUT IEBPTPCH any\_name**). The JCL can reside in a physical sequential data set or as a member of a partitioned data set. The new name (*any\_name*) on the IBM host is ignored and Cisco IOS for S/390 submits the job to the JES internal reader.

### Defaults

The server FTP defaults to RECFM=FB,LRECL=80,BLKSIZE=20000.

The defaults can be overridden by the Cisco IOS for S/390 administrator using the special **GAT TYPE(INTRDR)** entry.

## Usage Guidelines

The DCB attributes used by Server FTP to allocate the JES internal reader can be explicitly set by the user with the server FTP **SITE IRRECFM**, **IRLRECL**, and **IRBLKSIZE** commands.

Server FTP does a minimum of validity checking for internal reader file attributes. Be sure that the attributes selected are compatible with each other and are appropriate for the local Job Entry Subsystem.

## Examples

The following examples show JCL streams that can be submitted using **SITE SUBMIT** to cause the printing of the data included in the submitted job:

```
//IEBPTPCH JOB CARD
// *
// *   IF THIS JOB STREAM RESIDES ON YOUR REMOTE HOST, YOU CAN
// *   SUBMIT THIS JOB TO THE JES INTERNAL READER OF
// *
// *   YOUR IBM SYSTEM BY ENTERING THE FOLLOWING COMMANDS:
// *
// *   FTP MVS * CONNECT TO THE IBM HOST
// *   USERID/PSW * LOGIN TO THE IBM HOST
// *   QUOTE SITE SUBMIT * NOTIFY SNS/TCP SFTP TO
// *   PUT IEBPTPCH * SUBMIT JCL TO JES INTERNAL READER
//PTPCH      EXEC PGM=IEBPTPCH
//SYSPRINT   DD SYSOUT=A
//SYSIN      DD *
            PRINT PREFORM=FBA
//SYSUT2     DD SYSOUT=*
//          *SYSUT1 CAN BE PS OR MEMBER OF A PDS THAT RESIDES ON THE
//          *MVS SYSTEM
//          *ONE CAN INCLUDE THE PRINT FILE AS INSTREAM DATA AS
//          *SHOWN IN THIS EXAMPLE
// *
//SYSUT1     DD DATA
            (include print file here)

//IEBGENER JOB CARD
// *
// *   THIS JOB STREAM RESIDES ON YOUR REMOTE HOST, YOU CAN
// *   SUBMIT THIS JOB TO THE JES INTERNAL READER OF YOUR IBM
// *   SYSTEM BY ENTERING THE FOLLOWING COMMANDS:
// *
// *   FTP MVS * CONNECT TO THE IBM HOST
// *   USERID/PSW * LOGIN TO THE IBM HOST
// *   QUOTE SITE SUBMIT * NOTIFY SNS/TCP SFTP TO
// *   PUT IEBGENER * SUBMIT JCL TO JES INTERNAL READER
//GENER      EXEC PGM=IEBGENER
//SYSPRINT   DD SYSOUT=*
//SYSIN      DD DUMMY
//SYSUT2     DD SYSOUT=*
// *
// *   SYSUT1 CAN BE PS OR MEMBER OF A PDS THAT RESIDES ON THE
// *   MVS SYSTEM. ONE CAN INCLUDE THE PRINT FILE AS INSTREAM DATA
// *   AS SHOWN IN THIS EXAMPLE
// *
//SYSUT1     DD DATA
            (include print file here)
```



---

## Data Transfer Operations

This section describes the operation of data transfers by the Server FTP.

### Transfer Commands

Table 5-10 lists options invoked by the **PUT**, **APPE**, and **GET** commands that initiate data transfer:

**Table 5-10**      **Transfer Options**

Option	Function
<b>STOR</b>	saves a file on an MVS system
<b>APPE</b>	appends to a file stored on an MVS system
<b>RETR</b>	retrieves a file from an MVS system

A 226-Transfer complete reply is sent only when the disk file being written has been closed successfully or when the data being retrieved and sent across your network has been fully acknowledged.

### Truncating and Folding Records

The Server FTP generally folds rather than truncates a network record that exceeds LRECL (or BLKSIZE, for an unblocked data set). You can force truncation with an **ALLO R** command.

### Raveled Files

A *raveled* file is a file that contains the network data concatenated into the file with no record markers. Such a file can be created and later retrieved with FTP without loss of information. However, it is not usually possible to process it with any IBM utility. The only transformation generally done on a raveled file is to translate between ASCII and EBCDIC for TYPE A.

Create or retrieve a raveled file with the FTP parameters **STRU F** and one of these:

- Character type (TYPE A or TYPE E) and unblocked logical records
- TYPE I or TYPE L

When a raveled file is stored by the Server FTP, it is folded into maximum-sized logical records.

### Padding Fixed-Length Records

If the data set has a RECFM containing F (in other words, it has fixed-length logical records), and if it is not raveled, the Server FTP does these steps:

- Pads each record being written to the next LRECL.
- Removes trailing pad characters from each record read from disk.

The pad character is normally blank for character types and zero for binary types. However, the **SITE** command PAD option can set it to a different value. Although this is not strictly invertible, in most cases it saves transmission time and network bandwidth. The statistics at the end of a file transfer contain the number of records padded if this number is nonzero.

### Translation

For the ASCII transfer (TYPE AN, TYPE AT, or TYPE AC), EBCDIC data in a disk file is translated to ASCII over the network and vice versa. The Server FTP always stores data on disk in EBCDIC.

### Line Image Files

A line image file is defined by FTP parameters **STRU F**, one of the character types (TYPE AN, TYPE EN, TYPE AT, or TYPE ET), and a blocked disk data set (in other words, not raveled). With line image files, end-of-line (EOL) in the network data is mapped to MVS end-of-record and vice versa. Network end-of-line is normally NL (EBCDIC) or CRLF (ASCII). However, CRLF is also recognized in EBCDIC network data. Isolated CR or LF characters are not recognized as an EOL sequence.

### Record Structured Files

For the FTP parameter **STRU R**, a network record is mapped into an MVS logical record and vice versa.

If **ALLO R** is specified, each network record that exceeds **ALLO R** is truncated.

With **STRU R**, a storage operation (in other words, **STOR** or **APPE**) that creates a print file (TYPE AT, TYPE AC, TYPE ET, or TYPE EC) requires the data set to be blocked. If the data set is not blocked, the operation fails with the reply

```
501 print type and STRU R requires blocked data set.
```

### Tabs

Horizontal tab characters (HT) received from the network are expanded and/or deleted in accordance with the current **SITE** command **TABS** option setting. Default is a tab every eight columns; this can be overridden by the **SITE TABS** command.

A vertical tab (VT) character is always treated as data and stored in the file.

### Carriage Control and Format Effectors

For character types with format

ple locally applied rules that do not require buffering data. These rules are invertible between storing and retrieving data with the same parameters.

Some obscure cases cannot be handled correctly in this way. In fact, to fully define the correspondence, you must compare the effects of the files on assumed printing mechanisms to achieve the same appearance. This requires that FTP define additional attributes (such as the size of a page and the effect of an overstrike).

The rules assume that at the beginning of a file, the ASA line printer is positioned on the first line. Therefore, the default carriage control character used to create the first line of an ASA print file is + (suppress space), with following lines normally using a blank (single space) before printing.

---

## Character Type Rules

This section describes the transformation rules for creating and retrieving character-type (TYPE A or TYPE E) files.

In all cases, the network data can be ASCII or EBCDIC, but a disk data set is always EBCDIC. In addition, an HT character in a received file being stored (via **STOR** or **APPE**) is always handled in accordance with the current TABS value.

## File Structure with No Format Control

The **STRU F** command with TYPE AN or TYPE EN parameters define a file-structured (line-image) character file with no format control. Generally, with these parameters, network lines are mapped to and from logical records. The network data can be parsed into a series of lines, using the following syntax

```
text...eol
```

### Syntax Description

<i>text</i>	A (possibly null) block of text.
<i>eol</i>	An end-of-line sequence (CRLF or NL).

### Usage Guidelines

Other format effectors, including isolated CR and LF characters, can be included in text.

## Line Image Files

If the data set to be written or retrieved is blocked (RECFM includes B), the file is assumed to contain line images.

## Storing Line Image Files

These rules apply to line image files being stored (via **STOR** or **APPE**):

- An ASCII file (TYPE A) is translated to EBCDIC.
- The text is scanned for the EOL sequence (CRLF or NL).
- Each line is mapped into an MVS logical record, and the EOL sequence is discarded.
- A line that exceeds the target file LRECL is folded into subsequent logical record(s) rather than being truncated.
- A null record is stored as all pad characters if the RECFM includes F, zero data bytes if the RECFM includes V, or x'00' if the RECFM is U.
- If the data set has fixed length records (RECFM includes F), each line is padded with blanks (or the PAD character specified via the **SITE** command) to the next logical record boundary.
- Any control characters (including format effectors but not including the EOL sequence) are left in the data stream.
- If the data set has ASA carriage control (RECFM includes A), a blank Carriage Control Character (CCC) is inserted at the beginning of each logical record.

### Retrieving Line Image Files

These rules apply to line image files being retrieved (via RETR):

- Each MVS logical record is mapped into a line with an EOL sequence inserted after each line.
- If the type is ASCII (TYPE A), the data is translated from EBCDIC to ASCII.
- If the data set has fixed-length logical records (RECFM includes F), all trailing blanks (or the PAD character specified via the **SITE** command) are stripped off.
- If the data set has ASA carriage control (RECFM includes A), an additional transformation is applied. If the ASA CCC is the first character of the logical record and *text* is the rest, the data shown in Table 5-11 is sent over the network:

**Table 5-11 Retrieving Line Image Files Using Carriage Control Characters**

CCC	Network data
blank	eol <i>text</i>
0	eol eol <i>text</i>
-	eol eol eol <i>text</i>
1	CR FF <i>text</i>
+ (first record)	<i>text</i>
+ (other)	CR <i>text</i>

A single logical record can result in a series of network lines, and a + CCC in the first record of the file is effectively ignored.

### Raveled Files

If the data set to be written or retrieved is unblocked (RECFM does not include **B**), the file is assumed to be a raveled file and is treated as a single-byte string.

### Storing Raveled Files

The rules in this list apply to a raveled file being stored (via **STOR** or **APPE**).

The data set stored on disk is a concatenation of the network data. That is, the network data is folded into logical records.

An ASCII file (TYPE A) is translated to EBCDIC.

### Retrieving Raveled Files

These rules apply to a raveled file being retrieved (via RETR):

- The network data set is a concatenation of the retrieved logical records. That is, the logical records retrieved from disk are raveled into a single stream of bytes for transmission across the network.
- If the data set has ASA carriage control (RECFM includes A), the CCC is deleted and is not included in the network data.
- If the type is ASCII (TYPE A), the data is translated from EBCDIC to ASCII.

---

## File Structure with Telnet Format

The **STRU F** command with **TYPE AT** or **TYPE ET** parameters define a file-structured character file with Telnet format effectors. Generally, with these parameters, a network line maps to/from a logical record. The network data can be parsed into a series of segments using the following format:

*fe* text

### Syntax Description

*fe* A sequence of ASCII format effectors.

*text* A (possibly null) block of text.

### Usage Guidelines

Data storage with these parameters follows the same rules as storage of **STRU F**, **TYPE AN** or **TYPE EN**, with one exception. In the line image case, if the data set being stored into (via **STOR** or **APPE**) has ASA carriage control (RECFM includes A), the CCC is not set to blank (as in the **STRU F**, **TYPE AN** or **TYPE EN** case) but is determined from the network data according to the table in Retrieving Line Image Files. Any format effectors not involved in this transformation are left in the data stream.

Retrieval of data using these parameters follows the same rules as retrieval of **STRU F**, **TYPE AN** or **TYPE EN**.

## File Structure with ASA Format

The **STRU F** command with **TYPE AC** or **TYPE EC** parameters define a file-structured (line-image) print file containing ASA carriage control. Generally, with these parameters, a network line maps to/from a logical record. The network data consists of these parameters:

*cc* text...*eol*

### Syntax Description

*cc* An ASA Carriage Control Character.

*ext* A (possibly null) block of text.

*eol* An end-of-line sequence (CR, LF, or NL).

### Line Image Files

With these parameters, if the data set to be written or retrieved is blocked (RECFM includes B), the file is considered to be line images.

### Storing Line Image Files

These rules apply to line image files being stored (via **STOR** or **APPE**).

- An ASCII file (**TYPE A**) is translated to EBCDIC.
- The text is scanned for the EOL sequence (CRLF or NL).

- Each line is mapped into a logical record, and the EOL sequence is discarded.
- A line that exceeds the target file LRECL is folded into subsequent logical record(s), and a warning message is issued. If the file being written has ASA carriage control, each of the subsequent logical records is stored with a blank CCC.
- A null record is stored as all pad characters if the RECFM includes F, zero data bytes if the RECFM includes V, or x'00' if the RECFM is U.
- If the data set has fixed length records (RECFM includes F), each line is padded with blanks (or the PAD character specified via the **SITE** command) to the next logical record boundary.
- Any control characters (including format effectors but not including the EOL sequence) are left in the data stream.
- If the data set has ASA carriage control (RECFM includes A), the ASA CCC from the network data is used as the CCC for the stored logical record. If the data set does not have ASA carriage control, the first character of the network record is not stored in the logical record.

The Server FTP writes such a print file only into a blocked data set (RECFM includes B).

### Retrieving Line Image Files

When a **RETR** operation is performed with these **TYPE** and **STRU** parameters from a blocked data set, a new line image file is created and sent over the network. These rules apply to line image files being retrieved (via **RETR**):

- Each MVS logical record is mapped into a line with an EOL sequence inserted after each line.
- If the type is ASCII (**TYPE A**), the data is translated from EBCDIC to ASCII.
- If the data set has fixed length records (RECFM includes F), all trailing blanks (or the PAD character specified on the **SITE** command) are stripped off before sending on the network.
- If the retrieved data set contains ASA carriage control (RECFM includes A), the CCC from the logical record is passed as part of the network record. Otherwise, a blank CCC is inserted at the beginning of each network record, except for the first network record into which a + CCC is inserted.

### Raveled Files

If the data set to be written or retrieved is unblocked (RECFM does not include **B**), the file is considered to be a raveled file and is treated as a single string of bytes.

### Storing Raveled Files

The Server FTP does not write a raveled (unblocked) file with these data transfer parameters.

### Retrieving Raveled Files

These rules apply to a raveled file being retrieved (via **RETR**):

- The network data set is a concatenation of the retrieved logical records. That is, the logical records retrieved from disk are raveled into a single stream of bytes for transmission across the network.

- If the type is ASCII (TYPE A), the data is translated from EBCDIC to ASCII.
- If the data set has ASA carriage control (RECFM includes A), the CCC of the first logical record is included in the network data. Any other carriage control is deleted. If the data set does not have carriage control, a + CCC is inserted at the beginning of the network data, and no other carriage control is inserted.

## Record Structure with No Format

The **STRU R** command with TYPE AN, TYPE EN, TYPE AT, or TYPE ET parameters define a character file with record structure. Generally, network records are mapped to/from logical records.

*fe text...eor*

### Syntax Description

<i>fe</i>	A (possibly null) sequence of ASCII format effectors.
<i>text</i>	A (possibly null) block of text.
<i>eor</i>	An end-of-record sequence.

## Storing Logical Records

These rules apply to files being stored (using **STOR** or **APPE**) with these parameters:

- An ASCII file (TYPE A) is translated to EBCDIC.
- A null record is stored as all pad characters if the RECFM includes F, zero data bytes if the RECFM includes V, or x'00' if the RECFM is U.
- A network record longer than **ALLO R** (if specified) is truncated, and a warning message is issued.
- If a network record (after possible **ALLO R** truncation) is longer than LRECL, it is folded into subsequent logical record(s) and a warning message is issued.
- If the data set has fixed length logical records (RECFM includes F), each network record is padded with blanks (or the PAD character specified via the **SITE** command) to the next logical record boundary.
- If the data set has ASA carriage control (RECFM includes A), the CCC is normally blank, and any embedded ASCII format effectors are treated as data. However, if the TYPE is AT or ET, the escape sequences shown in Retrieving Logical Records (F and R) are scanned for and used to set CCC if found in the network data. No scanning for the escape sequences occurs with TYPE AN or TYPE EN.

The Server FTP writes a print file with Telnet format effectors (TYPE AT or TYPE ET) only into a blocked data set (RECFM includes B).

## Retrieving Logical Records

These rules apply to files being retrieved (via **RETR**) with these parameters:

- Each logical record (RECFM includes B) or each physical record (RECFM does not include B) is mapped into a network record. If the data set is unblocked (raveled), the network data is a concatenation of the logical records. The other transformations described in the following steps still apply.
- If the data set has fixed length logical records (RECFM includes F), all trailing blanks (or the PAD character specified via the **SITE** command) are stripped off before data is sent on the network. This occurs even for unblocked data sets. A completely blank (or null) record is sent as a null record with the appropriate end of record sequences based on mode.
- If the type is ASCII (TYPE A), the data is translated from EBCDIC to ASCII.
- If the data set has ASA carriage control (RECFM includes A), an additional transformation is applied. Table 5-12 shows the transformations that map the CCC in the disk data set into network data. A one-to-many relationship can occur between disk records and network records:

**Table 5-12 Retrieving Logical Records Using Carriage Control Characters**

CCC	Network Data
blank	<i>text eor</i>
0	<i>eor text eor</i>
-	<i>eor eor text eor</i>
1	<i>F' text eor</i>
+ (first)	<i>R' text eor</i>
+ (other)	<i>text eor</i>

---

**Note** If the type is ASCII (TYPE AN or TYPE AT), the EBCDIC single quote (') escape character is translated to an ASCII backslash \.

---

## Record Structure with ASA Format

The **STRU R** command with TYPE AC or TYPE EC parameters define a record-structured print file containing ASA carriage control. Generally, with these parameters, network records map to/from logical records.

*cc text...eor*

### Syntax Description

- cc* An ASA CCC.
- text* A (possibly null) block of text.
- eor* An end-of-record sequence.



---

## Storing Print Files

These rules apply to files being stored (via **STOR** or **APPE**) with these parameters:

- An ASCII file (TYPE A) is translated to EBCDIC.
- A null record is stored as all pad characters if the RECFM includes F, zero data bytes if the RECFM includes V, or x'00' if the RECFM is U.
- A network record longer than **ALLO R** (if specified) is truncated, and a warning message is issued.
- If a network record (after possible **ALLO R** truncation) is longer than LRECL, it is folded into subsequent logical record(s) and a warning message is issued. If the file being written has ASA carriage control (RECFM includes A), each of the subsequent logical records is stored with a blank CCC.
- If the data set has fixed length logical records (RECFM includes F), each network record is padded with blanks (or the PAD character specified via the **SITE** command) to the next logical record boundary.
- If the data set has ASA carriage control (RECFM includes A), the ASA CCC from the network data (cc) is used as the CCC for the stored logical record. Otherwise, the first character of the network data is deleted and is not stored in the logical record.

The Server FTP writes such a print file only into a blocked data set (RECFM contains B).

## Retrieving Print Files

These rules apply to files being retrieved (via **RETR**) with these parameters:

- Each logical record (RECFM includes B) or each physical record (RECFM does not include B) is mapped into a network record. If the data set is unblocked (raveled), the network data set is a concatenation of the logical records. The other transformations described in the following steps still apply.
- If the data set has fixed length logical records (RECFM includes F), all trailing blanks (or the PAD character specified via the **SITE** command) are stripped off before sending on the network. This occurs even for unblocked data sets. A completely blank (null) record is sent as a null record with the appropriate end of record sequences based on mode.
- If the type is ASCII (TYPE A), the data is translated from EBCDIC to ASCII.
- If the retrieved data set has ASA carriage control (RECFM includes A), the CCC from the logical record is passed as part of the network record. Otherwise, a blank CCC is inserted at the beginning of each network record, except for the first record, into which a + CCC is inserted.

## Binary-Type Rules

This section describes the transformation rules for creating and retrieving binary-type (TYPE I or TYPE L) files. The term “image-type” is used interchangeably with “binary-type” in the information in this section.

## No Record Structure

The **STRU F** command with TYPE I or TYPE L parameters define a binary file without record structure.

### Storing Binary Files

These rules define the transformations to such a file when it is stored (via **STOR** or **APPE**) by the Server FTP:

- The received network data is folded into maximum-length logical records when stored by the Server FTP.
- If a null (zero-length) network record is received, it is ignored.
- No translation, truncation, or padding is performed on the data.

Since the contents of an image-type file are considered to be untranslatable as characters, it is assumed not to be a print file. Therefore, the Server FTP does not write image-type data into a print data set (RECFM includes A).

### Retrieving Binary Files

These rules define the transformations to such a file when it is retrieved (via **RETR**) by the Server FTP:

- The retrieved logical records are concatenated into a single stream of network data.
- No translation, truncation, or removal of padding is performed on the retrieved data.

Since the contents of an image-type file are considered to be untranslatable as characters, it is assumed not to be a print file. Therefore, the Server FTP does not retrieve from a print data set (RECFM includes A).

## Record Structure

The **STRU R** command with **TYPE I** or **TYPE L** parameters define a binary file with record structure.

### Storing Structured Binary Files

These rules define the transformations to such a file when it is stored (via **STOR** or **APPE**) by the Server FTP:

- If a network record with zero data bytes is received, it is stored as all pad characters if RECFM includes F, zero data bytes if RECFM includes V, and x'00' if RECFM is U.
- A network record longer than **ALLO R** (if specified) is truncated to **ALLO R**, and a warning message is issued.
- If a record (possibly truncated to **ALLO R**) is longer than the maximum logical record length, it is folded into multiple logical records, and a warning message is issued.
- If the data set has fixed-length logical records (RECFM includes F), each network record is padded with zeros (or the PAD character specified via the **SITE** command) to the next logical record boundary.
- No translation is performed on the data.

Since the contents of an image-type file are considered to be untranslatable as characters, it is assumed not to be a print file. Therefore, the Server FTP does not write image-type data into a print data set (RECFM includes A).

---

## Retrieving Structured Binary Files

These rules define the transformations performed on such a file when retrieved (via **RETR**) by the Server FTP:

- The retrieved logical records are mapped into network records.
- If the data set has fixed length records (RECFM includes F), all trailing zero bytes (or the PAD character specified via the **SITE** command) are stripped off before sending on the network.
- No translation is performed on the retrieved data.

Since the contents of an image-type file are considered to be untranslatable as characters, it is assumed not to be a print file. Therefore, the Server FTP does not retrieve from a print data set (RECFM includes A) for transmission as image-type data

## Non-Invertible Retrieval

Unless a file is raveled, storing it with the Server FTP might transform the data in ways that are not strictly invertible. (That is, if the file is later retrieved using the same parameters, it might not be identical to the original file sent to this Server FTP.) The file would generally mean the same thing, but some of its byte stream is changed.

## Sources of Non-Invertibility

Here are some of the sources of non-invertibility introduced when a file is stored by this Server FTP:

- All files

If the data is stored into a data set with fixed-length blocks and if a record/line ends with the pad character the Server FTP uses to pad the block, that pad character is stripped off when the file is retrieved.

- Character files

HTs might be expanded to blanks during storage and not be reintroduced when the file is retrieved.

LF and FF, which do not appear at the left margin, insert blanks to position the virtual print head correctly. On RETR, these blanks and the extra CRLF that were not in the original input are returned to the remote host.

NL is used interchangeably with CRLF in an EBCDIC file. Therefore, CRLF received in such a file is sent out as an NL.

LFs preceding an FF with no intervening data are removed.

## Other Features

This section lists some other useful features of the FTP program.

### Testing and Debugging

The STAT display includes the Internals section that contains a number of important control fields and pointers. This information is generally of interest only to the system programmer.

The TCP-level tracing and debugging facilities are described in the *Cisco IOS for S/390 Customization Guide*.

### Telnet Break

The Telnet commands **IP**, **BRK**, or **CONTROL-C** on the control connection stops output from commands such as **HELP** and **STATUS**.