



ELIT MFT FTP User Guide

Table of contents

Introduction.....	3
Terminology and Acronyms	3
How to use this document.....	4
Feedback.....	4
About ELIT MFT.....	5
ELIT MFT FTP Server.....	5
FTP Client Software	5
Configuration Options	6
Getting set up with ELIT MFT.....	6
Connecting to ELIT MFT	6
File Types.....	7
IDS and Passwords.....	8
ELIT MFT Switchers.....	9
Data Elements for ELIT MFT Transfers	10
Routing a File Through ELIT MFT.....	11
Parameters for file delivery	11
Considerations	11
FROMDATA Options.....	11
Session Flow.....	12
Command Options	13
Examples of routing files	15
Listing Files.....	18
Retrieving Files From Your ELIT MFT Mailbox	20
Session Flow to retrieve all obtainable files	20
Session Flow to retrieve a specific file	20
Examples of retrieving files	21
Command Reference	23
CD Command	23
Using ELIT MFT to pull files from another FTP Server.....	40
Example of an FTTPULL trigger file.....	44
Example of how to initiate an FTTPULL.....	44
Appendix A – Sending files to a z/OS Mainframe.....	45
Carriage return/Line Feed Characters – KEEPLF, STRIPLF & STRIPSTREAM.	45
Appendix B – Feedback files.....	47

Appendix C – Frequently Asked Questions.....	49
Appendix D – PCI DSS V1.1 Password Information	51
Appendix E – Contacts and Assistance	52

Introduction

Terminology and Acronyms

Term	Definition
ASCII	American Standard Code for Information Interchange – a standard developed by the American National Standards Institute (ANSI) for the numeric representation of alphanumeric and special characters.
AUTH SSL	An FTP command that negotiates an explicit secure connection.
COMPORD	EDI standard for the steel industry.
Dataset Name	The equivalent to a file name on an IBM mainframe computer.
DNS	Domain Name Server
DSN	Data Set Name, the equivalent to a file name on an IBM mainframe computer.
EBCDIC	Extended Binary Coded Decimal Interchange Code – numeric code developed by IBM for the representation of alphanumeric and special characters.
EDI	Electronic Data Interchange – electronic application-to-application exchange of structured business data between companies.
EDI Data	Text files formatted according to an industry standard such as EDIFACT, X12 ISA, ICS, BG, COMPORD, VICS, or TRADACOM.
EDIFACT	Electronic Data Interchange for administration, commerce, and transport. ISO standard for EDI.
ELIT MFT ID	A unique code that identifies a user in the ELIT MFT network, i.e., an ELIT MFT User ID. User ID, Mailbox, and Location Code are synonymous.
ELIT MFT	DXC Technology's Managed File Transfer
ELIT MFT Filename	A name from 1 to 10 characters assigned by the Sender that identifies the transfer operation within ELIT MFT. This name is often dictated by the recipient to identify the receiving application or the specific transmission.
FTP	File Transfer Protocol
GUI	Graphical User Interface
IP	Internet Protocol
Location Code	A unique code that identifies a user in the ELIT MFT network, i.e., an ELIT MFT User ID. User ID, Mailbox, and Location Code are synonymous.
Mailbox	Identifies a holding place for files sent to your ELIT MFT ID. A mailbox ID is equivalent to an ELIT MFT ID or Location Code.
z/OS	A 64-bit operating system for mainframe computers
Obtainable Files	Files in your mailbox waiting to be retrieved.

PCI DSS V1.1	PCI DSS V1.1 stands for Payment Card Industry (PCI) Data Security Standard (DSS) V1.1. It was developed by the major credit card companies as a guideline to help organizations that process card payments to prevent credit card fraud, hacking and various other security issues. A company processing, storing, or transmitting credit card numbers must be PCI DSS compliant or they risk losing the ability to process credit card payments. Merchants and Service Providers must validate compliance with an audit by a PCI DSS Qualified Security Assessor (QSA) Company.
PGP	Pretty Good Privacy – data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication.
ReceiverID	ELIT MFT ID of the receiver of a file.
RFC	Request for Comments. A document in which an Internet standard or protocol is published.
SenderID	ELIT MFT ID of the sender of a file.
SSL	Secure Sockets Layer
Switcher	An ELIT MFT ID mailbox that can send and receive files on behalf of one or more other ELIT MFT IDs.
TCP/IP	Transmission Control Protocol/Internet Protocol
*THS/THD/TTR	ELIT MFT proprietary envelope records that contain file routing information.
Trading Partner	The sending and/or receiving parties involved in exchanging electronic data.
VAN	Value-Added Network
X12	ANSI standard for EDI.

How to use this document

This document describes how to access ELIT MFT using FTP. It assumes that you have a basic understanding of FTP commands and functions, and that you are using the FTP command line interface.

References of the FTP commands supported by the ELIT MFT FTP server are detailed in the [Command Reference](#) section. However, the entire document should be reviewed to gain an understanding of the ELIT MFT FTP Server.

Feedback

Comments on this document are welcome and encouraged. Please send your feedback to mft.services@dx.com.

About ELIT MFT

ELIT MFT is the acronym for DXC Technology's Managed File Transfer, a highly scalable, multi-platform, multi-protocol B2B communication service suite. ELIT MFT allows organizations and their trading partners (suppliers, customers, banks, freight forwarders, and the like) to access and exchange data across a single secure, global network. ELIT MFT supports multiple types of data file transfer: EDI, Electronic Funds Transfer (EFT), and proprietary data structures. ELIT MFT also supports multiple platforms and network communication protocols [, File Transfer Protocol (FTP), FTP/s, HyperText Transfer Protocol (HTTP), E5 and AS2] and network interconnections with both public and private third-party Value-Added Networks (VANs).

ELIT MFT FTP Server

The ELIT MFT FTP Server implements the FTP standard as documented by RFC-959, and the AUTH command specification as documented by RFC-2228 (security extensions). There are multiple FTP servers within the ELIT MFT network that support ELIT MFT FTP users. They run on IBM z/OS mainframes and ELIT MFT midrange. Certain FTP commands (such as `cd` and `pwd`) have been enhanced in ELIT MFT and function differently from other FTP implementations. For example, in ELIT MFT, the `cd` command specifies file routing and other file transfer parameters, whereas in a “typical” FTP environment the `cd` command changes a directory. The same is true for the second parameter in the `put` command. The [Command Reference](#) section describes each command parameter and how it is used with ELIT MFT. Although the ELIT MFT FTP Server has been enhanced in this way, standard FTP client software may be used to access ELIT MFT.

FTP Client Software

FTP client software is required to access ELIT MFT. Access over the Internet may also require FTP software that supports Secure Sockets Layer (SSL) RFC-4217. Refer to [Connection to ELIT MFT](#) for more information on connectivity options.

FTP client software is not standardized, therefore, support for common FTP commands may not be in all implementations. You may need to consult your software vendor for assistance on the options discussed in this document. Likewise, you may need to consult the vendor if your software uses a graphical user interface (GUI) instead of the command line interface. Executing FTP commands from within a Web browser to access ELIT MFT is not supported.

ELIT MFT FTP Server has been tested and is compatible with, but not limited to, the following FTP packages:

- WS/FTP Pro (Ipswitch) Note: ELIT MFT does not support interactive mode feature.
- ZMOD Exchange FTP (Trailblazer)
- FTP Voyager (RhinoSoft)
- CuteFTP Pro (GlobalSCAPE)
- Connect:Enterprise (Sterling Commerce)

- EDI*ASSET (DXC)
- BSDftpd-ssl FTPS (freeware)
- SSLFTP Secure SSL/TLS FTP Client
- Kermit FTP Client
- LexiCom (Cleo)

Configuration Options

ELIT MFT FTP IDs can be configured in two different ways.

Mailbox ID

An ELIT MFT FTP "mailbox" ID is one that is accessed by the user for both the uploading and downloading of data. Files for these sites are "mailboxed" (queued) for retrieval by the user.

Push ID

An ELIT MFT FTP "push" ID is one in which data is automatically delivered ("pushed") to the destination FTP server by the ELIT MFT Network application. There are specific requirements for defining an FTP "push" ID:

- The connection must be a secure one (for example: FTP/S using SSL).
- The owner must provide a 24x7 contact so that ELIT MFT support personnel will be able to notify the user of any connectivity issues.

The password for the userid on the destination server must be set to non-expiring.

Getting set up with ELIT MFT

Before using the ELIT MFT FTP service, you must have an ELIT MFT ID configured for FTP use. You can request a new ELIT MFT ID or have an existing ELIT MFT ID converted by contacting ELIT MFT Services via e-mail, mft.services@dx.com.

Connecting to ELIT MFT

This section describes the ways you can connect to the ELIT MFT FTP server.

Internet via a Secure Connection

Use secure FTP software, i.e., software that supports SSL. Secure FTP software can use either Explicit Session Security (AUTH command) or Implicit Session Security. See your software documentation on the capabilities of your software. The ELIT MFT FTP Server implements the AUTH command specification as documented by RFC-2228 (security extensions).

This option requires a "passive" FTP connection, and outbound ports 20900 through 20999 must be open on your firewall (contact your firewall administrator regarding this

option).

When connecting to ELIT MFT using Secure FTP software, use one of the Domain Name Server (DNS) names below as instructed by ELIT MFT Services.

Secure FTP connections using Explicit Session Security AUTH SSL:

mft1.svcs.entstvcs.com port 21 (LXA) or mft2.svcs.entstvcs.com port 21 (LXP)

Secure FTP connections using Implicit Session Security:

mft1.svcs.entstvcs.com port 990 or mft2.svcs.entstvcs.com port 990

DXC internal users can transmit files to MFT FTP Server using one of the following DNS names:

mft1.svcs.entstvcs.com or mft2.svcs.entstvcs.com

File Types

ELIT MFT can accept any type of data. Data files are considered text files (human-readable) or binary data.

Text files are referred to as either Electronic Data Interchange (EDI) data or proprietary data. EDI data files are formatted according to an industry standard such as EDIFACT, X12 ISA, ICS, BG, COMPORD, VICS, or TRADACOM.

Binary files are often not human-readable, such as compressed or encrypted files.

FTP File Transfer Modes

FTP supports ASCII and BINARY file transfer modes.

ASCII mode is intended for text files. Because of different computer platforms, text files may be converted from one character set to another, as discussed in the [Character Set Conversions](#) section. To avoid character set conversion, BINARY mode should be used.

With BINARY mode, files are unaltered; the receiver gets an exact copy of the file sent. This mode is appropriate for files such as compressed or encrypted files.

Character Set Conversions

To be readable, text files must be in the appropriate character set for the platform they reside on. The EBCDIC character set is used primarily by IBM computers, and the

ASCII character set is used by most non-IBM, mid-range and personal computers. ELIT MFT runs on IBM z/OS mainframes and ELIT MFT midrange servers and stores data internally in a native character encoding set for a given platform.

Files transferred into ELIT MFT in ASCII mode are assumed to be in the ASCII character set, and ELIT MFT may convert files into EBCDIC mode if necessary; if the receiving platform is an IBM mainframe.

From an ASCII platform, text files should be transferred to ELIT MFT in ASCII mode. From an EBCDIC platform, the transfer mode depends on your FTP client software. According to the FTP RFC, FTP client software on EBCDIC platforms should convert outgoing files from EBCDIC to ASCII. If your FTP client software conforms to these standards, text files may be transferred to ELIT MFT successfully in ASCII mode.

Be aware that when the `fromdata` option is used (discussed in the [FROMDATA Option](#) section), ELIT MFT must be able to read the data in the incoming file in ASCII. Logging on to ELIT MFT.

IDS and Passwords

Your assigned ELIT MFT ID is the logon ID to your ELIT MFT mailbox. A password must be provided when prompted for one during FTP connection initiation. An initial password value will be provided by ELIT MFT Services, or will be provided via email when your ID is generated. See [Appendix D](#) for more information related to PCI Password Policies.

The example below illustrates logging on to the ELIT MFT FTP Server. For security reasons, the password does not echo on your screen. Note the exact text that displays on your screen may be slightly different depending on your connection method (described in the [Connecting to ELIT MFT](#) section of this document).

```
ftp> open mft1.svcs.entsvcs.com
Connected to mft1.svcs.entsvcs.com.220-
220-      Welcome to the ELIT FTP Server
220-
220- Use of this system is restricted to authorized users.
220- User activity is monitored and recorded by system
personnel.
220- Anyone using this system expressly consents to such
monitoring
220- and recording.  BE ADVISED: if possible criminal
activity is detected,
220- system records, along with certain personal
information,
220- may be provided to law enforcement officials.
220
User (mft1.svcs.entsvcs.com:(none)): tdl
331 Username "tdl" needs password
Password:
230 User "TDL" successfully logged in
ftp>
```

You can change your password during the logon process by using the following syntax at the password prompt. Again, passwords are not echoed on the screen.

Password:*old_password/new_password*

Your new password will be active the next time you log on.

ELIT MFT Switchers

In certain instances, a company or site can have multiple ELIT MFT IDs. If your company has multiple ELIT MFT IDs, one ID can be used to send and receive files on behalf of the other IDs (a parent-child relationship). This type of ELIT MFT ID is referred to as a switcher. You can discuss requesting this option with any member of the ELIT MFT Services team.

Data Elements for ELIT MFT Transfers

Trading partners are the sending and/or receiving parties involved in exchanging electronic data. The Sender sends the data to ELIT MFT to be delivered to the Receiver of the data. The Sender and Receiver are identified by their assigned ELIT MFT IDs.

ELIT MFT needs the following data elements to route a file to its destination.

- The ELIT MFT ID of who sent the file, referred to as the SenderID.
- The ELIT MFT ID of who is to receive the file, referred to as the ReceiverID (or destination).
- ELIT MFT Filename, which is a label for the transmission. This is specified by the Sender at the time of transmission. However, it is often dictated by the Receiver to route the file to the receiving application.

In addition to the above, every ELIT MFT transmission also has the following associated with it.

- Reference (Reference). This is an optional string of 1 to 35 characters specified by the Sender to further identify the file.
- File Control Number (Control#). This is a number assigned by ELIT MFT to identify a specific transmission. This distinguishes transmissions from others with the same SenderID, ReceiverID, and ELIT MFT Filename.

Routing a File Through ELIT MFT

Standard FTP commands are used to specify ELIT MFT routing parameters. Detailed descriptions of all of the commands used in this section can be found in the [Command Reference](#) section.

Parameters for file delivery

To route a file, you need to specify a minimum of three data elements as follows: SenderID, ReceiverID, and the ELIT MFT Filename. Your trading partner should be able to provide you with the ReceiverID and the required ELIT MFT Filename, if any. The Reference element is optional and can be used to further identify the data.

For files sent to an z/OS mainframe or midrange destination, you must also provide ELIT MFT with the following. Your trading partner should be able to provide you with this information, as well.

- Logical record length required by the receiving application (specified with `quote site lrecl=n`).
- Indicate how carriage return/line feed characters should be handled (`keep1f`, `striplf`, or `stripstream` parameter). See [Appendix A - Sending Files to a z/OS Mainframe](#) for more explanation

Considerations

Using standard FTP commands, there are multiple ways to specify the file routing for ELIT MFT. You can instruct ELIT MFT to extract routing information from the data itself using the `fromdata` option (see the [FROMDATA Option](#) section), or you can explicitly route each individual file. All of the explicit routing options are supported by the `cd`, `quote site`, and `put` commands.

If you are not using `fromdata`, the SenderID defaults to the ELIT MFT ID that logged in. However, if your ELIT MFT ID is defined as a switcher (can send and receive files on behalf of other ELIT MFT IDs) you may need to specify the SenderID for each file if it is different from the ELIT MFT ID that logged in.

FROMDATA Options

When the `fromdata` option is used, ELIT MFT extracts the routing information, i.e., SenderID, ReceiverID, and ELIT MFT Filename, from the data within the file. `fromdata` works with two types of files.

- A file that contains EDI data (EDIFACT and X.12 ISA, ICS, or BG, COMPORD, VICS, or TRADACOM) and has the SenderID and ReceiverID predefined in ELIT MFT. One file can contain multiple documents, each with its own header and trailer records. The header record (e.g., ISA, UNB) for a subsequent document can immediately follow the trailer record (e.g., IEA, UNZ) for a previous document.

Optionally, the subsequent header can start in column 1 of the next record, with the previous trailer padded out to the `lrecl` value.

- For backward compatibility, `fromdata` can be used with files that contain ELIT MFT proprietary header and trailer records of `*THS/*THD/*TTR, *P*, or *#F/*#S/*#D`. These records contain the ELIT MFT SenderID and ReceiverID among other data and must begin in column 1. A file can contain multiple documents, each with its own header and trailer records.

*THS records must contain the correct transmission logical record length (`lrecl`) in columns 20-23 and the correct transmission block size (`blkksz`) in columns 24-27. In addition, the total size of the file has to be a multiple of the record length specified; the last data record must be padded with spaces to the complete length of the record (the `lrecl` value).

Contact ELIT MFT Services for more information on this option.

Here are a few things to note when using `fromdata`:

- The SenderID and ReceiverID in the headers must be valid, predefined, ELIT MFT IDs. In addition, the SenderID must be one of the following:
 - The same ELIT MFT ID as the one that logged in
 - Must convert to the ELIT MFT ID that logged in, i.e., be defined in ELIT MFT as an alias to the ELIT MFT ID that logged in
 - Defined as switching through the ELIT MFT ID that logged in
- Parameter values extracted from a file override those set with the `cd`, `quote site`, or `put` commands.
- The commands `cd fromdata`, `quote site rid=fromdata`, or `put localfile fromdata` will set `keeplf` for all files within the data stream. If `striplf` is required, then the parameter must be explicitly specified on the command (i.e. `cd fromdata/./striplf`, `quote site rid=fromdata,striplf`, `put localfile fromdata/./striplf`).
- Since ELIT MFT needs to read the routing data from within the file, ASCII files need to be transferred in ASCII mode, and EBCDIC files need to be transferred in BINARY mode.

Session Flow

This section describes what needs to occur in an FTP session to route a file through ELIT MFT for delivery. If your FTP client software uses a GUI, please consult with the software vendor on how to configure the software for the options discussed here.

Your FTP procedure should include the following steps. The [Examples of Sending Files](#) section at the end of this chapter illustrate the use of the commands.

- Log on to ELIT MFT.
- Set the transfer mode for the session. Text files should be transferred in ASCII mode, as opposed to binary mode. This is recommended for files sent from both ASCII and EBCDIC platforms.
- Specify the routing (SenderID, ReceiverID, and ELIT MFT Filename) and optional parameters (Feedback Filename, Application Reference) with the appropriate commands:
 - `cd command`
 - `quote site command`
- Note: routing parameters can also be specified on the `put` command (see below).

- Note: the `pwd` command can be used to display the parameter settings that are currently in effect.
- If routing to an z/OS destination (an IBM mainframe), specify the record length with the `quote site lrecl=n` command, and how to handle carriage return/line feed characters (see the [Appendix A - Sending Files to a z/OS Mainframe](#) section).
- Include the `put` command to transfer the file. If routing and session parameters were not set with `quote site` or `cd`, specify them on the `put` command line.
- Acknowledge the transmission with a subsequent FTP command of `pwd`, `dir`, `get`, `mget`, or `put`. This step is critical because it acknowledges completion of the transfer to ELIT MFT, indicating that the connection was not broken in the middle of a transmission. If a subsequent FTP command is not given, your file will not be delivered and your session will time out after 10 minutes.

Command Options

This section illustrates the different command options when routing a file through ELIT MFT.

Assume a file is to be sent to the destination of QE5 with the ELIT MFT Filename of DESADVT. QE5 resides on a z/OS mainframe and requires a record length of 80. The file contains EDI data and does not contain carriage return/line feed characters.

The following groups of commands are equivalent to each other, and each would successfully transfer the file. Note the SenderID defaults to the ID that logged on. Also, the `type ascii` statement sets the transfer to text/ascii mode; if this is the default mode, that statement can be eliminated.

```
ftp> type ascii
ftp> cd qe5/desadv
ftp> quote site lrecl=80, keep
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> cd qe5/desadv/./keep
ftp> quote site lrecl=80
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> quote site lrecl=80,rid=qe5,fnm=desadv,keep
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> quote site lrecl=80
ftp> put asn.txt qe5/desadv/./keep/asn.txt
```

```
ftp> pwd
ftp> quit
```

Using the same example, the commands below illustrate how a company with multiple mailboxes can specify a different SenderID from the one that is logged in. Again, these equivalent groups of commands show how different commands can achieve the same results.

```
ftp> type ascii
ftp> cd /lam/qe5/desadv
ftp> quote site lrecl=80, keep
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> cd /lam/qe5/desadv/./keep
ftp> quote site lrecl=80
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> quote site
lrecl=80,sid=lam,rid=qe5,fnm=desadv,keep
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> quote site lrecl=80
ftp> put asn.txt /lam/qe5/desadv/./keep/asn.txt
ftp> pwd
ftp> quit
```

When `fromdata` is used, the routing (SenderID, ReceiverID, ELIT MFT Filename) is extracted from the data. In addition, if ELIT MFT *THS/*THD/*TTR records are used, the record length is also taken from the header.

The following groups of commands are equivalent to each other.

```
ftp> type ascii
ftp> cd fromdata
ftp> put asn.txt
ftp> pwd
ftp> quit

- or -

ftp> type ascii
put asn.txt fromdata/asn.txt
```

```

ftp> pwd
ftp> quit

- or -

ftp> type ascii
ftp> quote site rid=fromdata
ftp> put asn.txt
ftp> pwd
ftp> quit

```

Examples of routing files

The examples below illustrate the commands that can be used to route files to another ELIT MFT ID. In both examples, the LAM ELIT MFT ID is logged in.

NOTE: When files are 'Rejected' due to Unknown ID's (or other error conditions), by default, the MFT FTP server will NAK (reply 426) the entire transmission and delete all received files. If the 'fdack' location parameter is specified within MFT for the logged on user, then the MFT FTP server will ACK (reply 226) the entire transmission and keep all received files. In this case, all 'Accepted' or 'Verified' files will be processed immediately, and all 'Rejected' files will be held until the Unknown ID's have been properly defined within MFT tables.

```

ftp> cd pol/release/./keep1f (set parameters for the upload)
250 SenderId: "LAM" ReceiverId: "POL" FileName: "RELEASE" RefNo: "."
KEEPLF
ftp> quote site lrecl=80 (set the recordsize required by the receiver)
200 Site Command Accepted
ftp> put release.dat (transfer the file)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 840000 bytes sent in 1.14Seconds 737.49Kbytes/sec.
ftp> pwd (acknowledge the transfer)
257 SenderId: "LAM" ReceiverId: "POL" FileName: "RELEASE" RefNo: "."
KEEPLF RELE
ASE.POL...release.dat Acked.

ftp> quote site lrecl=80,rid=pol,fnm=schedule,keep1f (alternate way to set
parameters)
200 Site Command Accepted
ftp> put schedule.dat (transfer the file)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 32763 bytes sent in 0.00Seconds 32763000.00Kbytes/sec.
ftp> pwd (acknowledge the transfer)
257 SenderId: "LAM" ReceiverId: "POL" FileName: "SCHEDULE" RefNo: "."
KEEPLF SCH
EDULE.POL...schedule.dat Acked.
ftp>

ftp> quote site rid=fromdata (direct ELIT MFT to get the routing from the data)
200 Site Command Accepted
ftp> pwd (display current parameter settings, note
KEEPLF)
257 SenderId: "LAM" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo:
"TCPIP" KEEPLF
ftp> cd fromdata (get the routing from the data, note KEEPLF)

```

```

250 SenderId: "LAM" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo:
"TCPIP" KEEPLF
ftp> put pos.dat (transfer the file)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 720 bytes sent in 0.00Seconds 720000.00Kbytes/sec.
ftp> dir fd (list the last file sent with "FROMDATA" and ack)
200 PORT Command Successful
150 Opening data connection
Doc# Control# Sender Receiver Status (1 Accepted - 0
Rejected)
1 000026221 LAM CNP Delivered
226 Transfer completed successfully. Data connection closed.
$DEFAULT.FROMDATA.
TCPIP.pos.dat(1 Accepted - 0 Rejected) Acked.
ftp: 127 bytes received in 0.00Seconds 127000.00Kbytes/sec.

```

This example shows how to specify a sender other than the one that is logged in. The ELIT MFT ID that logged in, POL, must be set-up by ELIT MFT Services as a switcher, and PO7 must be set-up as switching through POL. Note that a "/" preceding the first parameter of the cd command indicates it is the SenderID.

```

User (mft1.svcs.entsvcs.com: (none)): pol (log in)
331 Username "pol" needs password
Password:
230 User "POL" successfully logged in from IP
W2GZJ2JY02.amer.corp.eds.com/130.1
75.140.192
ftp> cd /po7/tcl/invoice (specify parameters; note the sender)
250 SenderId: "PO7" ReceiverId: "TDL" FileName: "INVOICE" RefNo:
"TCPIP" STRIPLF
ftp> put invoice.dat (transfer the file)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 98910 bytes sent in 0.11Seconds 899.18Kbytes/sec.
ftp> pwd (acknowledge the transfer)
257 SenderId: "PO7" ReceiverId: "TDL" FileName: "INVOICE" RefNo:
"TCPIP" STRIPLF
INVOICE.TDL.TCPIP.invoice.dat Acked.

```

This example shows an ERROR response when fromdata is used, and the data files contain unknown ID's. In this case, the data file is not successfully delivered to ELIT MFT, due to the FTP 426 response.

```

ftp> put pos2.dat (transfer the file)
200 PORT Command Successful
150 Opening data connection
426 ELIT has Rejected this Transmission, All Files Deleted, EDIDirect
Results: (1 Verified - 1 Rejected)
ftp: 720 bytes sent in 0.00Seconds 720000.00Kbytes/sec.
ftp> dir fd (list the last file sent with "FROMDATA")
200 PORT Command Successful
150 Opening data connection
Doc# Control# Sender Receiver Status (1 Verified - 1
Rejected)
1 000026222 LAM CNP Verified/Deleted

```

2 000026223 LAM ZZBADR Rejected/Deleted - Unknown
ReceiverID
226 Transfer completed successfully. Data connection closed.
ftp: 127 bytes received in 0.00Seconds 127000.00Kbytes/sec.

Listing Files

Files in your mailbox are displayed with the `dir` and `ls` (or `nlst`) commands. The `dir` command has several formats that display detailed information on each file. The status of `OBTAINABLE` denotes files sent to you that are available for pick up. Note that a unique control number is assigned to each transmission by ELIT MFT. This is to distinguish a particular transmission from others with the same SenderID, ReceiverID, and ELIT MFT Filename.

The sample session below illustrates some of the command options with the `dir` command. Detailed information on these commands can be found in the [Command Reference](#) section.

```
220-
220-      Welcome to the ELIT FTP Server
220-
220- Use of this system is restricted to authorized users.
220- User activity is monitored and recorded by system personnel.
220- Anyone using this system expressly consents to such monitoring
220- and recording.  BE ADVISED: if possible criminal activity is
220- detected,
220- system records, along with certain personal information,
220- may be provided to law enforcement officials.
220
User (mft1.svcs.entsvcs.com: (none)): tdl      (log in)
331 Username "tdl" needs password
Password:
230 User "TDL" successfully logged in
ftp> dir                                     (list files)
200 PORT Command Successful
150 Opening data connection
-r----- 1 POL      RUSHORDER  00839956 Nov 11 2005
RushOrder.txt.18602
-r----- 1 JAZ      ORDERS      00005280 Nov 11 2005
C0000000.C0169814.34295
-r----- 1 JAZ      ORDERS      00000560 Nov 11 2005
C0000000.C0169732.08587
-r----- 1 POL      ORDERS      00001024 Nov 11 2005 orders.dat.31160
226 Transfer completed successfully.  Data connection closed.
ftp: 305 bytes received in 0.01Seconds 30.50Kbytes/sec.
ftp> dir sn                                (list files, different format)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver Ctl#   RefNo   Deliver Date
RUSHORDER    OBTAINABLE POL      TDL      18602  TCPIP   11/11/2005
16:36
ORDERS        OBTAINABLE JAZ      TDL      34295  02011   11/11/2005
16:33
ORDERS        OBTAINABLE JAZ      TDL      08587  02011   11/11/2005
16:02
ORDERS        OBTAINABLE POL      TDL      31160  TCPIP   11/11/2005
15:56
226 Transfer completed successfully.  Data connection closed.
ftp: 366 bytes received in 0.00Seconds 366000.00Kbytes/sec.
ftp> dir c=18602                            (display details for a specific file)
200 PORT Command Successful
150 Opening data connection
===== Information for Control# 18602 =====
Source      -> POL
Dest        -> TDL
FileName    -> RUSHORDER
```

```

DSN          -> RushOrder.txt
Size         -> 00839956
RefNo        -> TCPIP
Date/Time    -> 11/11/2005 16:36
Status       -> OBTAINABLE
NetStatus    -> 010-FILE MAILBOXED, WAITING FOR PICKUP
226 Transfer completed successfully. Data connection closed.
ftp: 294 bytes received in 0.00Seconds 294000.00Kbytes/sec.
ftp> dir f=rushorder                                (list files named "rushorder")
200 PORT Command Successful
150 Opening data connection
-r----- 1 POL          RUSHORDER  00839956 Nov 11 2005
RushOrder.txt.18602
226 Transfer completed successfully. Data connection closed.
ftp: 75 bytes received in 0.00Seconds 75000.00Kbytes/sec.
ftp> dir s=pol                                       (list files sent from "POL")
200 PORT Command Successful
150 Opening data connection
-r----- 1 POL          RUSHORDER  00839956 Nov 11 2005
RushOrder.txt.18602
-r----- 1 POL          ORDERS      00001024 Nov 11 2005 orders.dat.31160
226 Transfer completed successfully. Data connection closed.
ftp: 147 bytes received in 0.00Seconds 147000.00Kbytes/sec.
ftp> ls                                              (brief list of files)
200 PORT Command Successful
150 Opening data connection
RushOrder.txt.18602
C0000000.C0169814.34295
C0000000.C0169732.08587
orders.dat.31160
226 Transfer completed successfully. Data connection closed.
ftp: 89 bytes received in 0.00Seconds 89000.00Kbytes/sec.
ftp> quit                                           (end session)
221 User logged out and closing connection

```

Retrieving Files From Your ELIT MFT Mailbox

You can retrieve files from your ELIT MFT mailbox with the FTP `mget` and `get` commands. The `mget` command retrieves all files waiting to be picked up from your mailbox. The `get` command retrieves one specific file. Detailed descriptions on the commands used in this section can be found in the [Command Reference](#) section.

ELIT MFT generates unique filenames for your downloaded files. By default, the name is the sender's original dataset name plus the file's control number, a unique identifier. To change the default filename, see the `mget sn` command and the `prefix`, `suffix`, and `mindsn` parameters under the `quote site` command.

In addition to secure retrieval of files using FTP/secure sessions, ELIT MFT has a capability of PGP encrypting files as they are being retrieved from a mailbox. Please contact ELIT MFT Services to request your mailbox to automatically PGP encrypt, and optionally PGP sign, files. You will need to provide a PGP public key to be used for encryption and may want to request ELIT MFT's, or your trading partner's public key for signature verification.

Session Flow to retrieve all obtainable files

Your FTP procedure should include the following steps. The [Examples of Retrieving Files](#) section at the end of this chapter illustrates use of the commands.

- Log on to ELIT MFT.
- Issue a `prompt off` command, if desired. By default, the `mget` command assumes an interactive session and asks for confirmation before transferring each file. This may not be desirable when running in an automated mode.
- Issue the `mget` command to retrieve all pending files.
- Acknowledge the transmission with a subsequent FTP command (i.e. `pwd`, `dir`, `get`, `mget`, or `put`). This step is critical because it acknowledges completion of the transfer to ELIT MFT, indicating that the connection was not terminated mid-transmission. If you omit this step, your session will time out after 10 minutes and the last file will remain in your mailbox.

Note: If an error/timeout occurs during initial file retrieval, then the file will remain ready for pickup in your mailbox. If an error/timeout occurs during a subsequent file retrieval (i.e. re-transmission of a file; file was successfully retrieved once before), then the file will be marked completed again, and it will not appear in your mailbox.

Session Flow to retrieve a specific file

ELIT MFT assigns a unique control number to each file transfer. This distinguishes a specific file from others with the same SenderID, ReceiverID, and ELIT MFT Filename. Files can be retrieved individually by supplying the control number on the `get` command line.

Your FTP procedure should include the following:

- Log on to ELIT MFT.
- Display files with the `dir` command to retrieve the ELIT MFT control number.

- Issue the `get` command with the control number.
- Acknowledge the transmission with a subsequent FTP command (i.e. `pwd`, `dir`, `get`, `mget`, or `put`). This step is critical because it acknowledges completion of the transfer to ELIT MFT, indicating that the connection was not terminated mid-transmission. If you omit this step, your session will time out after 10 minutes and the last file will remain in your mailbox.

Note: If an error/timeout occurs during initial file retrieval, then the file will remain ready for pickup in your mailbox. If an error/timeout occurs during a subsequent file retrieval (i.e. re-transmission of a file; file was successfully retrieved once before), then the file will be marked completed again, and it will not appear in your mailbox.

Examples of retrieving files

The following example session illustrates various ways to retrieve files from ELIT MFT.

```
220-
220-           Welcome to the ELIT FTP Server
220-
220- Use of this system is restricted to authorized users.
220- User activity is monitored and recorded by system personnel.
220- Anyone using this system expressly consents to such monitoring
220- and recording.  BE ADVISED: if possible criminal activity is
220- detected,
220- system records, along with certain personal information,
220- may be provided to law enforcement officials.
220
User (mft1.svcs.entsvcs.com: (none)): tcl           (log in)
331 Username "tcl" needs password
Password:
230 User "TCL" successfully logged in
ftp> prompt off                               (disable prompt for mget command)
Interactive mode Off .

ftp> dir                                         (list files)
200 PORT Command Successful
150 Opening data connection
-r----- 1 POL          RUSHORDER  00839956 Nov 11 2005
RushOrder.txt.18602
-r----- 1 JAZ          ORDERS      00005280 Nov 11 2005
C0000000.C0169814.34295
-r----- 1 JAZ          ORDERS      00000560 Nov 11 2005
C0000000.C0169732.08587
-r----- 1 POL          ORDERS      00001024 Nov 11 2005 orders.dat.31160
226 Transfer completed successfully.  Data connection closed.
ftp: 305 bytes received in 0.01Seconds 30.50Kbytes/sec.

ftp> dir sn                                     (list files, different format)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver  Ctl#   RefNo    Deliver Date
RUSHORDER    OBTAINABLE POL     TDL       18602    TCPIP    11/11/2005
16:36
ORDERS       OBTAINABLE JAZ     TDL       34295    02011    11/11/2005
16:33
ORDERS       OBTAINABLE JAZ     TDL       08587    02011    11/11/2005
16:02
ORDERS       OBTAINABLE POL     TDL       31160    TCPIP    11/11/2005
15:56
226 Transfer completed successfully.  Data connection closed.
ftp: 366 bytes received in 0.00Seconds 366000.00Kbytes/sec.

ftp> mget f=rushorder                           (pull file with filename of rushorder)
200 representation type now set to 'A'
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully.  Data connection closed.
```

```

ftp: 839956 bytes received in 10.51Seconds 79.96Kbytes/sec.
ftp> dir sn (list files and acknowledge mget)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender      Receiver Ctl#  RefNo      Deliver Date
ORDERS        OBTAINABLE JAZ      TDL        34295 02011     11/11/2005
16:33
ORDERS        OBTAINABLE JAZ      TDL        08587 02011     11/11/2005
16:02
ORDERS        OBTAINABLE POL      TDL        31160 TCPIP    11/11/2005
15:56
226 Transfer completed successfully. Data connection closed.
RushOrder.txt.1860
2 Acked.
ftp: 292 bytes received in 0.00Seconds 292000.00Kbytes/sec.
ftp> get 8587 pol_orders.dat (pull file with control # of 8587)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 560 bytes received in 0.85Seconds 0.66Kbytes/sec.
ftp> dir sn (list files and acknowledge get)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender      Receiver Ctl#  RefNo      Deliver Date
ORDERS        OBTAINABLE JAZ      TDL        34295 02011     11/11/2005
16:33
ORDERS        OBTAINABLE POL      TDL        31160 TCPIP    11/11/2005
15:56
226 Transfer completed successfully. Data connection closed. 8587
Acked.
ftp: 218 bytes received in 0.00Seconds 218000.00Kbytes/sec.
ftp> mget * (pull all obtainable files)
200 representation type now set to 'A'
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 5280 bytes received in 0.84Seconds 6.28Kbytes/sec.
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
C0000000.C0169814
.34295 Acked.
ftp: 1024 bytes received in 0.82Seconds 1.25Kbytes/sec.
ftp> dir sn (list files and acknowledge mget)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender      Receiver Ctl#  RefNo      Deliver Date
226 Transfer completed successfully. Data connection closed.
orders.dat.31160 A
cked.
ftp: 70 bytes received in 0.00Seconds 70000.00Kbytes/sec.
ftp> quit (end session)
221 User logged out and closing connection

```

Command Reference

This section describes the commands that can be used with the ELIT MFT FTP Server.

Wildcards can be applied to the “f=”, “s=”, “r=”, “d=”, and “cc=” parameters of the `dir`, `ls`, and `mget` commands. Including an asterisk (*) character at the beginning and/or end of the value matches any character (or none) that meets the rest of the criteria. For example, the command

```
ftp> dir f=*data1,r=n621
```

lists all files with an ELIT MFT Filename ending with `data1` and a Reference value equal to `n621`. Also, the command

```
ftp> dir s=user*,d=*edi*
```

lists all files with a SenderID that begins with `user`, and a DataSetName that contains the string `edi`.

Note that parameter values supplied to ELIT MFT (to the right of the equals sign) are not case sensitive.

CD Command

The `cd` command specifies parameters for subsequent `put` commands.

cd

/SenderID/ReceiverID/ELITMFTFilename/Reference/keep|f|strip|f|strip|stream

<i>SenderID</i>	Optional. 3 to 17 characters. The ELIT MFT ID that is sending files. This value defaults to the ELIT MFT ID that logged on. This is mostly used by ELIT MFT IDs defined as switchers.
<i>ReceiverID</i>	Required; 3 to 17 characters. The ELIT MFT ID to receive the files being sent or the fromdata or mailbag verb (refer to CD FROMDATA and CD MAILBAG for more information). If a SenderID is not specified, a “/” should not prefix the ReceiverID.
<i>ELITMFTFilename</i>	Optional. 1 to 10 alphanumeric characters. The name to be given to identify the next transmission. This value is often dictated by the Receiver. The default is \$default.

<i>Reference</i>	Optional. 1 to 35 alphanumeric characters. The reference string to be assigned to any files sent. This value is optional, however if <code>keeplf</code> , <code>striplf</code> , or <code>stripstream</code> is specified, a "." must be used here as a placeholder. The default is <code>tcpip</code> .
<code>keeplf</code>	Optional. This qualifier keeps carriage return/line feed characters in your files. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>striplf</code>	Optional. This qualifier removes carriage return/line feed characters from your files but uses their position to determine the end of the record. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>stripstream</code>	Optional. This qualifier removes carriage return/line feed characters but does not use their position to determine the end of the record. Instead, the data is filled in using all of the characters except the carriage return/line feed. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .

Notes

- The session default is `striplf`. Refer to the [Appendix A - Sending Files to a z/OS Mainframe](#) section for more information.
- The values extracted from a file with `fromdata` overrides any value set with the `cd` command. Please see the [FROMDATA Option](#) section for more information. The session default for `fromdata` changes to `keeplf`.

Examples

Assume that ELIT MFT ID TDL has issued the following commands.

```
ftp> cd pol
250 SenderId: "TDL" ReceiverId: "POL" FileName: "$DEFAULT" RefNo:
"TCPIP" STRIPLF
```

```
ftp> cd pol/prodrpt
250 SenderId: "TDL" ReceiverId: "POL" FileName: "PRODRPT" RefNo:
"TCPIP" STRIPLF
```

```
ftp> cd pol/prodrpt/./keeplf
250 SenderId: "TDL" ReceiverId: "POL" FileName: "PRODRPT" RefNo: "."
KEEPLF
```

```
ftp> cd pol/summrpt/decem/keeplf
250 SenderId: "TDL" ReceiverId: "POL" FileName: "SUMMRPT" RefNo:
"DECEM" KEEPLF
```

CD FROMDATA

cd fromdata/*ELITMFTFilename/Reference/keeplf|striplf|stripstream*

See the previous section for parameter descriptions; all of the parameters except for **fromdata** are optional.

The **cd fromdata** command instructs ELIT MFT to extract the file routing from the data in the file. This can be used with two types of files: 1) Files that contain EDI data. 2) Files wrapped in ELIT MFT proprietary headers, such as *THS/*THD/*TTR headers. The routing and certain other fields (ELIT MFT Filename, Record Length, feedback, etc.) are taken from the headers. The SenderID and ReceiverID in the data must be predefined in ELIT MFT.

Notes

- The values extracted from a file with **fromdata** overrides any value set with the **cd**, **quote site**, or **put** commands. Please see the [FROMDATA Option](#) section for more information.
- This command changes the session default to **keeplf**.

Example

Assume that ELIT MFT ID TDL has issued the following command.

```
ftp> cd fromdata
250 SenderId: "TDL" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo:
"TCPIP" KEEPLF
```

CD MAILBAG

The **cd mailbag** command informs ELIT MFT that EDI data is being sent using the X12.56 Mailbag standard.

cd mailbag/*ELITMFTFilename/Reference/keeplf|striplf|stripstream*

See the **cd command** section for parameter descriptions; all of the parameters except for **mailbag** are optional.

Example

Assume that ELIT MFT ID TDL has issued the following command.

```
ftp> cd mailbag
250 SenderId: "TDL" ReceiverId: "MAILBAG" FileName: "$DEFAULT" RefNo:
"TCPIP" STRIPLF
```

DIR MAILBAG

The `dir` command displays files in your mailbox waiting to be picked up, as indicated by the OBTAINABLE status. The maximum default number of files displayed is 250. See the `quote site` command to learn how to override the maximum default value of listed files.

```
dir      s=SenderID ,cc=ReceiverID, f=ELITMFTFilename,  
r=Reference, c=Control#, d=DataSetName, sn, fd
```

<i>no parameter</i>	This is the default display of File Permissions, SenderID, ELITMFTFilename, File Size, Deliver Date, and DataSetName for files sent to you (ReceiverID).
<i>s=SenderID</i>	Optional. Displays files sent to you with the SenderID specified. A wildcard value can be used within the SenderID value.
<i>cc=ReceiverID</i>	Optional. This is used by ELIT MFT IDs that are defined as switchers. It displays files for only the ReceiverID specified. <code>sn</code> can be used with this parameter, i.e., <code>dir sn,cc=ReceiverID</code> . A wildcard value can be used within the ReceiverID value.
<i>f=ELITMFTFilename</i>	Optional. Displays files sent to you with the ELITMFTFilename specified. A wildcard value can be used within the ELITMFTFilename value.
<i>r=Reference</i>	Optional. Displays files sent to you with the Reference specified. A wildcard value can be used within the Reference value.
<i>c=Control#</i>	Optional. Displays more detailed status information on the specific file.
<i>d=DataSetName</i>	Optional. Displays files sent to you with the DataSetName specified. A wildcard value can be used within the DataSetName value.
<i>sn</i>	Optional. Includes inbound and outbound files in the display. Displays ELITMFTFilename, Status, SenderID, ReceiverID, Control#, Reference, and Deliver Date for files sent to you.
<i>snfcn</i>	Optional. Includes inbound and outbound files in the display. Displays ELITMFTFilename, Status, SenderID, ReceiverID, Control#, ELIT MFT File Control#, and Deliver Date for files sent to you.

fd	Optional. Displays the last file sent from the current session with the destination of FROMDATA.
snacr	Optional. Displays ELITMFTFilename, Status, SenderID, ReceiverID, Control#, ELIT MFTACR# and Deliver Date for files sent to you.
snref	Optional. Displays ELITMFTFilename, Status, SenderID, ReceiverID, Control#, Reference and Deliver Date for files sent to you.
sndoc	Optional. Displays ELITMFTFilename, Status, SenderID, ReceiverID, Control#, DocType# and Deliver Date for files sent to you.

Notes

- Files sent to yourself (loopback files) will not display unless you issue a “`cd ELIT MFT ID`” command, where *ELIT MFT ID* is the ID that logged in.

Examples

```
ftp> dir (list files)
200 PORT Command Successful
150 Opening data connection
-r----- 1 JAZ      ORDERS      00016829 Feb 19 2003
C0000000.C0687243.01776
226 Transfer completed successfully. Data connection closed.
ftp: 79 bytes received in 0.00Seconds 79000.00Kbytes/sec.
ftp> dir sn (list files, different format)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender      Receiver Ctl#  RefNo      Deliver Date
ORDERS        OBTAINABLE JAZ         TDL         01776 00035      02/19/2003
13:44
226 Transfer completed successfully. Data connection closed.
ftp: 142 bytes received in 0.01Seconds 36.60Kbytes/sec.
ftp> ls (list files, brief format)
200 PORT Command Successful
150 Opening data connection
C0000000.C0687243.01776
226 Transfer completed successfully. Data connection closed.
ftp: 25 bytes received in 0.00Seconds 25000.00Kbytes/sec.

ftp> dir fd (list the last file sent with FROMDATA)
200 PORT Command Successful
150 Opening data connection
Doc# Control#  Sender      Receiver      Status (1 Accepted - 0
Rejected)
1      000025697 LAM          CNP           Delivered
226 Transfer completed successfully. Data connection closed.
$DEFAULT.FROMDATA.
TCPIP.salesdata.dat(1 Accepted - 0 Rejected) Aoked.
ftp: 127 bytes received in 0.00Seconds 127000.00Kbytes/sec
```

GET Command

The `get` command downloads an individual file.

get *Control# Localfile*

Control# Required. The control number of the file to retrieve (displayed with the `dir sn` command).

Localfile Required. The name of the local file to contain the data.

Notes

- A subsequent FTP command of `pwd`, `dir`, `get`, `mget`, or `put` must be issued after a `get` command to acknowledge that you received the file. Otherwise, the file is not marked as transferred and will stay with the `OBTAINABLE` status in your mailbox. To avoid this scenario, issue a `pwd` command after all FTP activity is complete and before you exit the FTP session.

Example

```
ftp> dir sn (list files)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender      Receiver Ctl#  RefNo      Deliver Date
ORDERS        OBTAINABLE JAZ         TDL       01776 00035      02/19/2003
13:44
226 Transfer completed successfully. Data connection closed.
ftp: 142 bytes received in 0.00Seconds 366000.00Kbytes/sec.
ftp> get 01776 orders.txt (retrieve file with control # of 01776)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 16829 bytes received in 0.42Seconds 39.97Kbytes/sec.
ftp> quit (acknowledge the get and end the session)
221 User logged out and closing connection 1776 Acked.
```

LS or NLS Command

This command displays a brief list of files waiting to be picked up. Only the *DataSetName* is displayed. The default maximum number of displayed files is 250. See the `quote site` command to learn how to override this limit.

ls *s=SenderID, f=ELITMFTFilename, c=Control#*

no parameter Displays the *DataSetName* of files not yet retrieved.

s=SenderID Optional. Displays files sent to you with `OBTAINABLE` status and the *SenderID* specified.

	A wildcard value can be used within the SenderID value.
<i>f=ELITMFTFilename</i>	Optional. Displays files sent to you with OBTAINABLE status and the given ELITMFTFilename value. A wildcard value can be used within the ELITMFTFilename value.
<i>c=Control#</i>	Optional. Displays detailed information on the specific file.

Example

```
ftp> ls (list files)
200 PORT Command Successful
150 Opening data connection
C0000000.C0687243.01776
226 Transfer completed successfully. Data connection closed.
ftp: 25 bytes received in 0.00Seconds 25000.00Kbytes/sec
```

MGET Command

The `mget` command retrieves all files in your mailbox up to a maximum of 250, however the maximum can be increased with the quote site `maxlist` command. If the number of files in the mailbox exceeds the maximum, multiple `mget` commands can be used to retrieve all files. See the quote site command for details.

ELIT MFT generates unique filenames for the downloaded files. By default, the name is the sender's original dataset name plus the file's control number. To change the default filename, see the `sn` parameter below and the `prefix`, `suffix`, and `mindsn` parameters under the `quote site` command.

mget *

or

mget sn,

s=SenderID, cc=ReceiverID, f=ELITMFTFilename, r=Reference, d=DataSetName

*

Required if no other parameters are specified. Retrieves all files waiting to be picked up. When issued by an ELIT MFT ID defined as a switcher, files with the status of OBTAINABLE for all subordinate ELIT MFT IDs are retrieved.

sn

Optional. Create DataSetNames of ELITMFTFilename + SenderID + Reference + DataSetName + Control#. If not specified, the

	default DataSetName is DataSetName + Control#.
<i>s=SenderID</i>	Optional. Retrieve only those files with the specified SenderID. A wildcard value can be used within the SenderID value.
<i>cc=ReceiverID</i>	Optional. Retrieves files with the OBTAINABLE status for that ReceiverID, only. This is used by ELIT MFT IDs that are defined as switchers. A wildcard value can be specified within the ReceiverID value.
<i>f=ELITMFTFilename</i>	Optional. Retrieve only those files with the given ELITMFTFilename. A wildcard value can be specified within the ELITMFTFilename value.
<i>r=Reference</i>	Optional. Retrieve only those files with the specified Reference. A wildcard value can be used within the Reference value.
<i>d=DataSetName</i>	Optional. Retrieve only the file with the given DataSetName. A wildcard value can be used within the DataSetName value.

Notes

- A subsequent FTP command of `pwd`, `dir`, `get`, `mget`, or `put` must be issued after an `mget` command to acknowledge that the last file was received. Otherwise, the file is not marked as transferred and will stay with the OBTAINABLE status in your mailbox. To avoid this scenario, issue a `pwd` command after all FTP activity is complete and before you exit the FTP session.
- By default, `mget` assumes an interactive session and asks for confirmation before it transfers each file. This can usually be turned off by entering the prompt off command before issuing the `mget`.

Examples

The following sample session illustrates different **mget** parameters.

```
ftp> dir (list files with dataset names)
200 PORT Command Successful
150 Opening data connection
-r----- 1 TDL      ORDER      00259279 Nov 29 2005 order.txt.16628
-r----- 1 TQL      ORDERS     00005280 Nov 29 2005
C0000000.C0212688.16629
-r----- 1 TQL      ORDERS     00055350 Nov 29 2005
C0000000.C0212686.16625
-r----- 1 POL      ORDERS     00001024 Nov 29 2005 orders.dat.16627
-r----- 1 POL      RUSHORDER 00001024 Nov 29 2005 orders.dat.16622
-r----- 1 JAZ      WKLYORDERS 00005280 Nov 29 2005
C0000000.C0212578.16621
```

```

226 Transfer completed successfully. Data connection closed.
ftp: 452 bytes received in 0.00Seconds 452000.00Kbytes/sec.
ftp> dir sn (list files with control #s and status)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver Ctl#  RefNo  Deliver Date
ORDER         OBTAINABLE TDL     LAM      16628 TCPIP   11/29/2005
20:50
ORDERS        OBTAINABLE TQL     LAM      16629 02029 11/29/2005
20:48
ORDERS        OBTAINABLE TQL     LAM      16625 02029 11/29/2005
20:47
ORDERS        OBTAINABLE POL     LAM      16627 TCPIP   11/29/2005
20:41
RUSHORDER     OBTAINABLE POL     LAM      16622 TCPIP   11/29/2005
20:29
WKLYORDERS    OBTAINABLE JAZ     LAM      16621 02029 11/29/2005
20:24
226 Transfer completed successfully. Data connection closed.
ftp: 514 bytes received in 0.00Seconds 514000.00Kbytes/sec.
ftp> mget f=rushorder (retrieve files named rushorder)
200 representation type now set to 'A'
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 1270 bytes received in 1.77Seconds 0.72Kbytes/sec.
ftp> dir sn (list files & ack the mget; note that
rushorder is not there)

200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver Ctl#  RefNo  Deliver Date
ORDER         OBTAINABLE TDL     LAM      16628 TCPIP   11/29/2005
20:50
ORDERS        OBTAINABLE TQL     LAM      16629 02029 11/29/2005
20:48
ORDERS        OBTAINABLE TQL     LAM      16625 02029 11/29/2005
20:47
ORDERS        OBTAINABLE POL     LAM      16627 TCPIP   11/29/2005
20:41
WKLYORDERS    OBTAINABLE JAZ     LAM      16621 02029 11/29/2005
20:24
226 Transfer completed successfully. Data connection closed.
orders.dat.
16622 Acked.
ftp: 440 bytes received in 0.00Seconds 440000.00Kbytes/sec.
ftp> mget s=t* (retrieve files from IDs that start with
"T")

200 representation type now set to 'A'
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 259525 bytes received in 4.20Seconds 61.85Kbytes/sec.
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
order.txt.16628 Acked.
ftp: 5526 bytes received in 1.74Seconds 3.17Kbytes/sec.
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
C0000000.C0212688
.16629 Acked.
ftp: 55596 bytes received in 2.07Seconds 26.82Kbytes/sec.
ftp> dir (list files & ack the mget; note which
ones are left)

200 PORT Command Successful
150 Opening data connection
-r----- 1 POL      ORDERS      00001024 Nov 29 2005 orders.dat.16627

```

```

-r----- 1 JAZ          WKLYORDERS 00005280 Nov 29 2005
C0000000.C0212578.16621
226 Transfer completed successfully. Data connection closed.
C0000000.C0212686.
16625 Acked.
ftp: 151 bytes received in 0.00Seconds 151000.00Kbytes/sec.
ftp> dir sn                                (list files, alternate display)
200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver Ctl#  RefNo      Deliver Date
ORDERS        OBTAINABLE POL      LAM      16627 TCP/IP      11/29/2005
20:41
WKLYORDERS    OBTAINABLE JAZ      LAM      16621 02029      11/29/2005
20:24
226 Transfer completed successfully. Data connection closed.
ftp: 28 bytes received in 0.00Seconds 218000.00Kbytes/sec.
ftp> mget *                                (retrieve all files)
200 representation type now set to 'A'
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 1270 bytes received in 1.76Seconds 0.72Kbytes/sec.
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
orders.dat.16627 Acked.
ftp: 5526 bytes received in 1.76Seconds 3.14Kbytes/sec.
ftp> dir sn                                (list files & ack the mget; note
                                                that none are left)

200 PORT Command Successful
150 Opening data connection
Filename      Status      Sender  Receiver Ctl#  RefNo      Deliver Date
226 Transfer completed successfully. Data connection closed.
C0000000.C0212578.16621 Acked.

```

PUT Command

Sends a file to ELIT MFT to be delivered to the *ReceiverID* specified.

put *Localfile*

/SenderID/ReceiverID/ELITMFTFilename/Reference/keep|f|strip|f|stripstream
/DatasetName

<i>Localfile</i>	Required. The name of the file to upload; the filename must start with an alphabetic character.
<i>SenderID</i>	Optional. 3 to 17 characters. The ELIT MFT ID that is sending files. This value defaults to the ELIT MFT ID that logged on. This is mostly used for ELIT MFT IDs defined as switchers.
<i>ReceiverID</i>	Optional. 3 to 17 characters. The ELIT MFT ID to receive the files being sent or the <i>fromdata</i> or mailbag verb (refer to CD FROMDATA and CD MAILBAG for more information). If a SenderID is not specified, a “/” should not prefix the ReceiverID.

<i>ELITMFTFilename</i>	Optional. 1 to 10 alphanumeric characters. The name to be given to identify the next transfer. This value is often dictated by the Receiver. The default is \$default. If this parameter is specified, the ReceiverID must also be specified.
<i>Reference</i>	Optional. 1 to 35 alphanumeric characters. The Reference value to be assigned to any files sent. This value is user-defined and further identifies the file. It is optional unless <code>keeplf</code> , <code>striplf</code> , or <code>stripstream</code> is specified, in which case a "." can be used here as a placeholder. If this parameter is specified, the <code>ELITMFTFilename</code> must also be specified.
<code>keeplf</code>	Optional. This qualifier keeps carriage return/line feed characters in your files. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>striplf</code>	Optional. This qualifier removes carriage return/line feed characters from your files but uses their position to determine the end of the record. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>stripstream</code>	Optional. This qualifier removes carriage return/line feed characters but does not use their position to determine the end of the record. Instead, the data is filled-in using all of the characters except the carriage return/line feed characters. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<i>DataSetName</i>	Required only when the second parameter is specified (see the explanation under Notes below). This is the name to assign the file at its destination.

Notes

- The session default is `striplf`. Refer to the [Appendix A - Sending Files to a z/OS Mainframe](#) section for more information.

- The values extracted from a file with `fromdata` overrides any value set with the `put` command. Please see the [FROMDATA Option](#) section for more information. The session default for `fromdata` changes to `keep1f`.
- This command is made up of 2 parameters: the filename and the routing parameter. The routing parameter is made up of several elements: `SenderID`, `ReceiverID`, `ELITMFTFilename`, `Reference`, `keep1f|strip1f|stripstream`, and `DataSetName`.
- A subsequent FTP command must be issued after a `put` command to acknowledge file transfer. Otherwise, the file is considered incomplete and will not be delivered to the destination. The subsequent FTP command can be `pwd`, `dir`, `get`, `mget`, or `put`; it is good practice to issue a `pwd` command after all FTP activity is complete and before you exit the FTP session.
- The second parameter is optional and can include file routing elements, but if the second parameter is used the last element, `datasetname`, is required (as shown below). However, all other elements in the second parameter are optional and are read from left-to-right. Consider the following example.

```
ftp> put asn1400.dat QE5/DESADVT/asn.dat
```

ELIT MFT takes the last element, `asn.dat`, as the new `datasetname`, then reads the other values in order from left-to-right. Since it does not start with a `/`, the first value of `QE5` is taken as the `ReceiverID` (rather than the `SenderID`); the ELIT MFT Filename is `DESADVT`; `DataSetName` is `asn.dat`.

A fully qualified `put` command is shown below. It routes the file `DATA.DAT` to ELIT MFT using the `SenderID` of `TD1`, the `ReceiverID` of `LAM`, the `ELITMFTFilename` of `JUNREPORT`, the `Reference` of `0331`, the `keep1f` qualifier, and assigns `RPT.DAT` as the dataset name at the destination.

```
ftp> put data.dat /td1/lam/junreport/0331/keep1f/rpt.dat
```

Be aware that with some FTP client software defaults the second parameter to the first, if both parameters are not given. For example, assume uploading the local file `d:/data/mfg/asn.dat` with the following `put` command.

```
ftp> put d:/data/mfg/asn.dat
```

Since the second parameter was not given, the software defaults the second parameter to the values of the first, as follows.

```
ftp> put d:/data/mfg/asn.dat d:/data/mfg/asn.dat
```

ELIT MFT assumes the values between slashes (`/`) in the second parameter are the file routing. This results in error because `d:` is considered the *ReceiverID*. If your FTP client behaves this way, you can resolve this by specifying *just* a dataset name so no defaults are taken from the first parameter, such as:

```
ftp> put d:/data/mfg/asn.dat asn.dat
```

Note that if the above command is used, the routing must be set with a previous `cd` or `quote site` command.

Examples

The following examples are equivalent and show the different parameters that can be used with the `put` command. They all send a file named `NOVSALES.DAT` to the ELIT MFT ID of TDL, assigning it the ELIT MFT Filename of `SALES`. `KEEPPLF` indicates that carriage return/line feed characters are not to be removed. Since the dataset name is given in the second parameter, the file is named `NOVSALES.DAT` on TDL's system.

```
ftp> put novsales.dat tdl/sales/./keeplf/novsales.dat      (transfer the
                                                            file)

200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 258955 bytes sent in 0.26Seconds 995.98Kbytes/sec.
ftp> pwd                                                    (acknowledge the transfer
                                                            was successful)

257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "."
KEEPPLF SALES.
TDL...novsales.dat Acked.
ftp> quote site rid=tdl,fnm=sales,keeplf                  (set parameters for the
                                                            transfer)

200 Site Command Accepted
ftp> pwd                                                    (display parameters)

257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "."
KEEPPLF
ftp> put d:\alamb\novsales.dat novsales.dat              (transfer the file)

200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 258955 bytes sent in 0.29Seconds 892.95Kbytes/sec.
ftp> pwd                                                    (acknowledge the transfer was
                                                            successful)

257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "."
KEEPPLF SALES.
TDL...novsales.dat Acked.
ftp> cd tdl/sales/./keeplf                                (set parameters for the
                                                            transfer)

250 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "."
KEEPPLF
ftp> put novsales.dat                                     (transfer the file)

200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 258955 bytes sent in 0.29Seconds 892.95Kbytes/sec.
ftp> pwd                                                    (acknowledge the transfer was
                                                            successful)

257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "."
KEEPPLF SALES.
TDL...novsales.dat Acked.
ftp> cd tdl/sales                                         (set parameters for the
                                                            transfer)

250 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "TCPIP"
STRIPPLF
ftp> quote site keeplf                                    (set parameters for the
                                                            transfer)

200 Site Command Accepted
ftp> put novsales.dat                                     (transfer the file)

200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
```

```
ftp: 258955 bytes sent in 0.31Seconds 835.34Kbytes/sec.
ftp> pwd (acknowledge the transfer was successful)
257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "TCPIP"
KEEPPLF SALES.TDL.TCPIP.novsales.dat Acked
```

QUOTE SITE Command (not applicable with SSH, [See Appendix A](#))

Specifies ELIT MFT parameters for subsequent transfers.

quote site

```
sid=SenderId, rid=ReceiverID, fnm=ELITMFTFilename,
ref=Reference,
lrecl=RecordLength,
keepplf|striplf|stripstream,
fb=ELITMFTFilename, nat=IPAddress,
prefix=DataSetName, suffix=gdg|Suffix, mindsn,
chunk=RecordLength, pgp, nlst, cryptpass,
maxlist=#ofFiles
```

sid=SenderId	Optional. 3 to 17 characters. The ELIT MFT ID that is sending files. This value defaults to the ELIT MFT ID that logged on. This is mostly used by ELIT MFT IDs defined as switchers.
rid=ReceiverID	Optional. 3 to 17 characters. The ELIT MFT ID to receive the files being sent or the fromdata or mailbag verbs (refer to CD FROMDATA and CD MAILBAG for more information). If a SenderID is not specified, a "/" should not prefix the ReceiverID.
fnm=ELITMFTFilename	Optional. 1 to 10 alphanumeric characters. The name to be given to identify the next transmission. This value is often dictated by the Receiver. The default is \$default.
ref=Reference	Optional. 1 to 35 alphanumeric characters. The reference string to be assigned to any files sent. This value is optional unless keepplf, striplf, or stripstream is specified, in which case a "." can be used here as a placeholder
lrecl=RecordLength	Optional. This specifies the logical record length of the file, which is a value from 1 to

	32760. It's used when the receiver requires fixed-length records.
<code>keeplf</code>	Optional. This qualifier keeps carriage return/line feed characters in your files. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>striplf</code>	Optional. This qualifier removes carriage return/line feed characters from your files but uses their position to determine the end of the record. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>stripstream</code>	Optional. This qualifier removes carriage return/line feed characters but does not use their position to determine the end of the record. Instead, the data is filled-in using all of the characters except the carriage return/line feeds. Please refer to the Appendix A - Sending Files to a z/OS Mainframe section for more information. The default mode is <code>striplf</code> .
<code>fb=ELITMFTFilename</code>	Optional. Requests a feedback file indicating the status of the file sent. See Appendix B – Feedback Files for more information on Feedback Files.
<code>nat=IP-Address</code>	Optional. This is the Network Address Translation (NAT) address of the ELIT MFT FTP server as assigned by the accessing site. This is required when using Network Address Translation with FTP/S, and the FTP transfer mode is PASSIVE.
<code>prefix=</code>	Optional. Dataset name prefix to assign to files retrieved with the <code>mget</code> command.
<code>DataSetName</code>	The <code>DataSetName</code> is formed as <code>prefix.suffix<control#></code> (<code>control#</code> is 5 digits). If no suffix is given, the default is <code>fst</code> . If <code>suffix=gdg</code> is given, the <code>DataSetName</code> is formed as <code>"prefix(+1)"</code> . The <code>."</code> character is

allowed in the DataSetName. See the Examples at the end of this section.

`suffix=Suffix or GDG` Optional. Keyword of `gdg` or the string to use as the suffix of the dataset name. The DataSetName is formed as `prefix.suffix<control#>` (control# is 5 digits). If no suffix is given, it defaults to `fst`. If `suffix=gdg`, is given, (+1) is used as the suffix, and the DataSetName is formed as "prefix(+1)". See the Examples at the end of this section.

`mindsn` Optional. Requests a minimal dataset name as a reply to `ls` or `mget`. The original dataset name is replaced by `fstxxxxx`, where `xxxxx` is the control number of the entry. This is used for platforms such as the AS400.

`chunk=RecordLength` Optional. ELIT MFT inserts carriage return/line feed characters at the record length given for files retrieved with the `get` and `mget` commands. Used for platforms such as the AS400.

`pgp` Optional. This option indicates that the data file is encrypted using PGP security and should be decrypted by the ELIT MFT software as it enters the network.

`nlst` Optional. This option is valid for FTP 'pulls' only. Some remote servers will not respond correctly to the 'mget' command used in an FTP pull. The `nlst` command instructs the remote FTP server to provide the list of available files in a different format that will usually correct this issue.

`cryptpass` Optional. This option is valid for FTP 'pulls' only. This option indicates that the supplied password in the FTP pull file is encrypted and should be decrypted prior to being sent to the remote FTP server. The encrypted password can be supplied by a request to **mft.services@dx.com**.

`maxlist=#ofFiles` Optional. Overrides the default maximum number of files of 250 in `dir`, `ls` and `mget` commands.

Notes

- The session default is `striplf`. Refer to the [Appendix A – Sending Files to a z/OS Mainframe](#) section for more information.
- The values extracted from a file with `fromdata` overrides any value set with the `quote site` command. Please see the [FROMDATA Option](#) section for more information. The session default for `fromdata` changes to `keeplf`.

Examples

```
ftp> quote site lrecl=80,rid=pol,fnm=invoice,keeplf (set parameters for upload)
```

```
200 Site Command Accepted
```

```
ftp> pwd
```

(display parameters)

```
257 SenderId: "LAM" ReceiverId: "POL" FileName: "INVOICE" RefNo:
"TCPIP" KEEPLF
```

```
ftp> dir
```

(list files)

```
200 PORT Command Successful
```

```
150 Opening data connection
```

```
-r----- 1 POL          MNTHSALES  00659200 Nov 30 2005
```

```
mnthsales.txt.14527
```

```
226 Transfer completed successfully. Data connection closed.
```

```
ftp: 75 bytes received in 0.01Seconds 7.50Kbytes/sec.
```

```
ftp> mget * (download all files)
```

```
200 representation type now set to 'A'
```

```
mget mnthsales.txt.14527? n
```

("n" rejects download; note name of file)

```
ftp> quote site prefix=sales,suffix=dat
```

(set parameters for naming file)

```
200 Site Command Accepted
```

```
ftp> mget *
```

```
200 representation type now set to 'A'
```

```
mget SALES.DAT14527? y
```

("y" confirms download; note name of file)

```
200 PORT Command Successful
```

```
150 Opening data connection
```

```
226 Transfer completed successfully. Data connection closed.
```

```
ftp: 659446 bytes received in 7.82Seconds 84.36Kbytes/sec.
```

```
ftp> dir
```

(acknowledge transfer was successful)

```
200 PORT Command Successful
```

```
150 Opening data connection
```

```
226 Transfer completed successfully. Data connection closed.
```

```
SALES.DAT14527 Aced.
```

```
ftp> quote site rid=fromdata
```

```
200 Site Command Accepted
```

```
ftp> pwd
```

```
257 SenderId: "TDL" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo:
```

```
"TCPIP" KEEPLF
```

Using ELIT MFT to pull files from another FTP Server

ELIT MFT can be used to pull data files into your ELIT MFT mailbox from FTP servers outside of the ELIT MFT network. This is done by sending a trigger file to ELIT MFT which contains the FTP commands to execute on the remote server.

An ELIT MFT ID has to be defined by ELIT MFT Services for the FTP server you want to access. The pull function is initiated by sending a file to that ELIT MFT ID with an ELIT MFT Filename of "FTPPULL".

The FTP commands allowed in the trigger file are listed below.

```
user Userid/Password
binary
nodelete
requeue
site sid=SenderID,rid=ReceiverID,fnm=FileName,ref=Reference
site failmail=eMailId,emptymail=eMailId,keep1f,striplf
site lrecl=RecordLength,as1.5
site pgp
dirempty ERROR 550 Mailbox Empty
fileempty ERROR 550
retryalways
cd RemoteDirectory
mget *
get RemoteFileName
ignore RemoteFileName
```

Minimally, for non-PCI ELIT MFT implementations the trigger file must contain a user command and either a `get` or `mget` command; the other commands are optional. For PCI ELIT MFT implementations the UserId/Password combination is pulled from the ELIT MFT Location Table of the destination site (the `user` command is ignored if specified). In addition, the destination of the FTTPULL file must be defined as a restricted receiver site within the ELIT MFT Location Table.

If a ReceiverID is not given in a `site` command, the files pulled from the remote FTP server will be sent to the SenderID that sent the trigger file. See previous descriptions for SenderID, ReceiverID, ELITMFTFilename, Reference, `striplf`, `keep1f`, and `lrecl`.

<code>user Userid/Password</code>	Specifies the logon UserId and Password on the remote system, NOTE: UserId and Password within the FTTPULL file are only valid for non-PCI ELIT MFT implementations. For PCI ELIT MFT implementations, the UserId and Password are pulled from the ELIT MFT Location Table of the destination site.
-----------------------------------	---

<code>binary</code>	This parameter will set the transfer mode to binary. If not specified, the transfer mode is ASCII.
<code>nodelete</code>	By default, ELIT MFT deletes the file from the remote system after the data has been retrieved. This prevents pulling the file repeatedly from subsequent “pull” sessions. Some FTP servers provide this operation automatically and disallows the delete function, hence, the ELIT MFT “delete” command results in error. This parameter removes the delete command from the dialogue.
<code>failmail=eMailId</code>	Specifies the eMailId to receive notice of a problem in the “pull” operation. Only <u>one</u> eMail address can be specified on a failmail= parameter, but multiple failmail= parameters can be specified.
<code>emptymail=eMailId</code>	Specifies the eMailId to receive notice of a “pull” attempt that did not find any files on the remote FTP server. Only <u>one</u> eMail address can be specified on a emptymail= parameter, but multiple emptymail= parameters can be specified.
<code>dirempty <error string></code>	Used to specify the <error string> exception message returned by the remote FTP server when a “pull” attempt using MGET did not find any files. If the remote server exception message (i.e. 'ERROR 550') matches the beginning of the <error string> (i.e. `dirempty ERROR 550' specified), then ELIT MFT will positively acknowledge (ACK) the FTTPULL file (meaning this is a one-time pull attempt), otherwise ELIT MFT will negatively acknowledge (NAK) the FTTPULL file and a retry of the same “pull” will be attempted again in 5 minutes.

Note: If the remote server exception message matches 'ERROR 550 ***

ERROR *** NO BATCHES' (returned by Sterling's FTP server), then ELIT MFT will always positively acknowledge the FTPPULL file, even if NO 'dirempty' parameter is specified.

`fileempty <error string>` Used to specify the <error string> exception message returned by the remote FTP server when a "pull" attempt using GET, did not find a matching file. If the remote server exception message (i.e. 'ERROR 550') matches the beginning of the <error string> (i.e. 'fileempty ERROR 550' specified), then ELIT MFT will positively acknowledge (ACK) the FTPPULL file (meaning this is a one-time pull attempt), otherwise ELIT MFT will negatively acknowledge (NAK) the FTPPULL file and a retry of the same "pull" will be attempted again in 5 minutes.

Note: If the remote server exception message starts with 'ERROR 550', then ELIT MFT will positively acknowledge (ACK) the FTPPULL file, even if NO 'fileEmpty' parameter is specified. (This can be overridden by specifying the `retryAlways` parameter).

`retryAlways`

This parameter overrides the 'fileempty' parameter described above and forces the FTPPULL file to be negatively acknowledged (NAK), whenever an error is detected during FTP GET processing, and a retry of the same "pull" will be attempted again in 5 minutes.

`cd RemoteDirectory`

Specifies the remote directory from which to "pull" files. If not specified, files are pulled from the default directory of the logon UserId.

`mget *`

Indicates that all files in the remote directory should be retrieved. Please note that the "mget" command will not process a masked request (I.E. `mget abc*`) – It will always pull all available files from the directory.

<code>get RemoteFileName</code>	Indicates to retrieve the single file specified by RemoteFileName.
<code>ignore RemoteFileName</code>	Instructs ELIT MFT to exclude specific files from being pulled when <code>mget *</code> command is used. Ignore can be used multiple times within the same script. This directive can be used to avoid problems associated with an attempt to pull none-retrievable files using <code>mget *</code> command.
<code>as1.5</code>	Indicates that the remote files are using the as1.5 protocol. This is a proprietary format that uses the AS1 data packaging format over the FTP protocol, where files are digitally encrypted and signed. ELIT MFT will decrypt the data in a staging area and verify the signature prior to “pulling” the data into the ELIT MFT network. The <code>requeue</code> function described below is designed to recover files that experience certain errors in the as1.5 processing.
<code>pgp</code>	This optional site command indicates that the data file is PGP encrypted at the remote host and should be decrypted by the ELIT MFT server when it is being pulled into the ELIT MFT network.
<code>requeue</code>	This function is used in conjunction with the as1.5 protocol and directs ELIT MFT to retrieve a file from the staging area.

When files are “pulled” using as1.5, they are placed in a staging area in order to decrypt and verify the signature. Several types of transient errors can cause the transfer to fail once the data has been staged (certificate not found, sender/receiver not defined in ELIT MFT, ...). After the error condition has been remedied, you can retrieve the file from the staging area (since the file was already deleted from the remote FTP server) by sending a trigger file including the `requeue` parameter, the `as1.5` parameter, and a `get` command with the filename returned in the “failmail” e-mail notice.

Example of an FTPPULL trigger file

The example below illustrates the contents of an FTPPULL trigger file. ELIT MFT uses the commands within the trigger file to PULL data from a remote FTP site.

Contents of a trigger file named 'pullfrompol.dat':

```
user lam/lampswd
site rid=lam,fnm=poldata,ref=pull020108
site emptymail=notify1@dxs.com
site emptymail=notify2@dxs.com
fileempty ERROR 550
get poldata.txt
```

Example of how to initiate an FTPPULL

The example below illustrates the commands that can be used to route an FTPPULL trigger file into ELIT MFT. Once the FTPPULL trigger file is successfully delivered, ELIT MFT will then issue the commands within the trigger file to PULL data from a remote FTP site, in this case, PULL data from location POL. See also [Routing a File Through ELIT MFT](#) for additional FTP parameter and session information.

```
User (mft1.svcs.entsvcs.com: (none)): lam                (log in)
331 Username "lam" needs password
Password:
230 User "LAM" successfully logged in from IP /130.1
75.140.192
ftp> cd pol/ftppull                                     (set parms for FTPPULL file
                                                         upload)
250 SenderId: "LAM" ReceiverId: "POL" FileName: "FTPPULL" RefNo:
"." STRIPLF
ftp> put pullfrompol.dat                                (transfer the FTPPULL file)
200 PORT Command Successful
150 Opening data connection
226 Transfer completed successfully. Data connection closed.
ftp: 8400 bytes sent in 1.14Seconds 737.49Kbytes/sec.
ftp> pwd                                                (acknowledge the FTPPULL file
                                                         transfer)
257 SenderId: "LAM" ReceiverId: "POL" FileName: "FTPPULL" RefNo:
"." STRIPLF FTPPULL.POL...pullfrompol.dat Acked.
ftp> quit                                              (end the session)
221 User logged out and closing connection.
```

Appendix A – Sending files to a z/OS Mainframe

Because of differences in computer platforms, you may need to specify parameters that allow ELIT MFT to provide a native text file to a receiving z/OS (mainframe) or midrange server. This is done by indicating the logical record length required by the receiving application, and how ELIT MFT should handle carriage return/line feed characters in your data. The latter depends on the type of text file you are sending to ELIT MFT, which can contain:

Un-delimited data, which is a continuous, unbroken stream of text without carriage return and/or linefeed characters separating the data into records. EDI data is often formatted this way.

Delimited data, which is data that contains carriage return/line feed characters to separate records. The records can be of fixed-length (all the same length) or of variable-length (each record can have a different length).

Carriage return/Line Feed Characters – KEEPLF, STRIPLF & STRIPSTREAM

Essentially, you need to instruct ELIT MFT whether to strip carriage return/line feed characters out of your data and whether records should be separated based on where carriage return/line feed characters appear in the data. This is controlled by the `keeplf`, `striplf`, and `stripstream` options, which are specified with the `cd`, `put`, and `quote site` commands. Note that this only applies to files you send to ELIT MFT, not files you retrieve from ELIT MFT.

`keeplf`, `striplf`, and `stripstream` work in conjunction with the `quote site lrecl= n` command, where `n` is the record length required by the receiving z/OS application. Your trading partner should be able to tell you what record length, if any, is required by their application.

KEEPLF

When you specify `keeplf`, carriage return/line feed characters are not removed from the file, and the data is separated into fixed-length records with the length given in the `lrecl` parameter.

This option is often used for EDI data files (which usually do not contain carriage return/line feed characters). EDI files requiring 80 byte fixed-length records are common and are sent using an `lrecl` of 80 and `keeplf`.

Notes

The session default when `cd fromdata` is used is `keeplf`. Please see the [FROMDATA Option](#) section for more information.

STRIPLF

`striplf` is the default setting for FTP file transfers and is often used for non-EDI, proprietary-formatted files or files that are meant for display on a terminal. `striplf` causes ELIT MFT to remove carriage return/line feed characters, however, their position in the data determines the end of each record.

In detail, if `quote site lrecl=n` is specified, the resulting file will contain records of the same size. If the position of the carriage return/line feed character is less than the value specified in the `quote site lrecl=n` command, the record is padded with spaces up to position `n`. If it is greater than `n`, the record is truncated to the `n`th position. If no carriage return/line feed characters are in the file, the entire file is considered one large record, and the record will be truncated to a length of `n`.

For example, assume an `lrecl` of 80 is specified. If a carriage return/line feed appears at position 100, the record is truncated at the 80th position, and the characters in positions 81-100 are discarded. If a carriage return/line feed appears in position 60, the record is padded with spaces from positions 60 to 80. If no carriage return/line feed characters are in the file, the file will contain one record that is 80 bytes long.

If the destination requires a file with fixed-length records and your file has a carriage return/line feed at the same position in every record (and that position is the same as the `lrecl` value), `striplf` can be used. For example, `striplf` would work well with a file that has carriage return/line feed characters in every record at positions 81 and 82, and the destination requires an `lrecl` value of 80.

Note that if `quote site lrecl=n` is not specified and your data does not contain carriage return/line feed characters, the maximum file size allowed is 32760.

STRIPSTREAM

`stripstream` also strips out carriage return/line feed characters from your data. However, unlike `striplf`, the characters are just skipped over and do not determine the end of the records. The records are continuously filled in with the data, with no truncation or padding, and are separated by the `lrecl=n` value. However, the last record is padded with spaces up to the `lrecl=n` value.

To illustrate, assume the following file. `<cr><lf>` indicates carriage return/line feed characters and `^` is a space character.

```
ABCDEFGH<cr><lf>
HIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz
mnopqrstuvwxyz
```

The resultant file using `STRIPSTREAM` and a `lrecl` of 30 looks like this:

```
ABCDEFGHGIJKLMNOPQRSTUVWXYZ0123
456789abcdefghijklmnopqrstuvwxyz
yz^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Appendix B – Feedback files

Feedback files are files sent to you containing delivery status information on specific files you sent to ELIT MFT for delivery. These files report on the delivery status only, and do not indicate application status.

You must specifically request a feedback file with the `quote site fb=ELITMFTFilename` command, where `ELITMFTFilename` is the name of the feedback file to be sent to your ELIT MFT ID.

A feedback file will be sent to you every time there is a status change in the data being reported on. For example, if you send data to another FTP mailbox, you will receive a feedback file when the data is delivered to the recipient's mailbox, then another when the data is pulled (assuming the destination is an FTP "pull" site as opposed to an FTP "push" site). Feedback files are also sent when data that was never pulled is deleted. Under usual circumstances, this occurs when files are purged from a mailbox after their retention period has expired.

If the feedback filename is prefixed with `$$`, the feedback file is only sent if the operation was not successful. The feedback filename sent back to you excludes the `$$` from the name.

The feedback file contains the following information.

Line 1: Columns 01-10: ELIT MFT Filename of the original file you sent

12-16: Reference (ACR) number of the original file you sent
18-26: File Control Number of the original file you sent
28-37: Date the status message occurred (see Line 2)
39-46: Time the status message occurred (see Line 2)
48-64: The destination of the original file you sent
66-72: Size, in bytes, of the original file you sent
74-76: ELIT MFT region generating the feedback

Line 2: Columns 01-80: Status Message of one of the following:

001-FILE UNLOADED INTO DSN-
002-FILE PUT ON xxxx HELD
003-FILE CANCELED BY OPERATOR
004-FILE PURGED BY OPERATOR
005-FILE UNLOADED (DELIVERED ON TIME)
006-FILE UNLOADED (DELIVERED LATE)
007-FILE NOT DELIVERED, PURGED
008-FILE UNLOADED INTO DSN-(DUMMY)
009-FILE UNLOADED INTO DSN-(LOST&FOUND)
010-FILE MAILBOXED, WAITING FOR PICKUP
012-FILE DELIVERED OFFNET TO
013-FILE PICKED UP OFFNET BY
014-FILE DELIVERY DELAYED-UNABLE TO CONNECT TO

The most common messages are described below.

001-FILE UNLOADED INTO DSN- Dataset name
The file reached its destination on a z/OS mainframe.

003-FILE CANCELED BY OPERATOR
The file was canceled by an operator, or the file was purged by the system (file retention period expired).

004-FILE PURGED BY OPERATOR
The file was deleted by an operator.

010-FILE MAILBOXED, WAITING FOR PICKUP
The file was delivered to a mailbox and is waiting for retrieval.

012-FILE DELIVERED OFFNET TO location
The file was sent to another VAN.

013-FILE PICKED UP OFFNET BY user
The file was destined to an ELIT MFT "pull" mailbox and was retrieved.

014-FILE DELIVERY DELAYED-UNABLE TO CONNECT TO location
The file was not delivered because ELIT MFT could not connect to the specified location (i.e. another MFT region/hub, or remote Trading Partner location).

Appendix C – Frequently Asked Questions

Q: I'm having problems sending a file to a z/OS application that requires 80-byte records. What could be the problem?

A: When you send a file with 80 byte records to a z/OS mainframe, your data cannot contain carriage return or line feed characters. You must also include the following command in your FTP procedure.

```
quote site lrecl=80, keep1f
```

This formats the file properly for a mainframe by blocking the data into 80 byte records.

Q: I can't transmit a file via FTP. What could be the problem?

A: There are many reasons why a file may not successfully transmit or route to the receiving application. You may want to check the following.

- Connectivity issues.
 - Ensure that you are connecting to the correct IP address or DNS name.
 - Check that you are not having a firewall issue and that all appropriate ports are open.
 - Ensure that you can log in with your ELIT MFT ID and password.
- Your transmission may not be authorized because of billing issues.
- If using the `fromdata` option, ensure that the SenderID and ReceiverID in your EDI-formatted documents are valid and predefined in the ELIT MFT Network. If `fromdata` is not used, ensure a valid ELIT MFT ReceiverID is specified with the `quote site, cd, or put` commands.
- If the file is very large, you might be experiencing a time-out condition.
- The file is transmitting successfully, but the receiving application may be expecting a file of a different format (`keep1f, strip1f, stripstream, lrecl`) or with a different ELIT MFT Filename.

Q: I cannot open the zip file that was sent to me. What is the problem?

A: The file may have been transmitted in text mode instead of binary mode. Contact the sender and ask that they retransmit the file in binary mode.

Note that trading partners who use a VAN to connect to ELIT MFT cannot exchange non-EDI data files.

Q: Does ELIT MFT support the EBCDIC or ASCII character set?

A: ELIT MFT stores data in a native character encoding set (ASCII or EBCDIC) for the platform MFT hub is running on. Text files inbound to ELIT MFT are assumed to be ASCII files and may be converted into EBCDIC mode if necessary, if the receiving platform is an IBM mainframe. Any file that should not be converted, such as compressed or encrypted files, should be transferred to ELIT MFT in binary

mode. This can usually be done with the type image or type binary ftp commands. Files sent using fromdata in binary mode must be ASCII files.

Q: Does ELIT MFT support non-EDI data?

A: Yes, however the receiver must not reside on a VAN other than ELIT MFT.

Q: I'm getting the error "451 Record greater than the max of 32760." What is going on?

A: You are probably trying to send a file that does not contain carriage return/line feed characters with the `striplf` parameter, and your file is greater than the maximum allowed of 32760 bytes. Revisit whether you need to use `striplf`. EDI data should be sent using the `keeplf` parameter instead of `striplf`.

Appendix D – PCI DSS V1.1 Password Information

The passwords for PCI DSS v1.1 must adhere to the following policies:

The system owner identity will be verified before performing password resets of the Location IDs. The location IDs will be assigned by a consistent identifier assignment. The identifiers will contain information such as name and related access information. (PCI DSS v1.1 8.5.2, EDSSCS0456, EDSSCS0467)

First-time passwords will be set to a unique value for each location ID and changed immediately after the first use. (PCI DSS v1.1 8.5.3)

Access for any terminated users associated to Location IDs will be immediately revoked. (PCI DSS v1.1 8.5.4)

Inactive Location IDs will be removed at least every 90 days. After 90 days of being disabled, the Location IDs will be removed from the Location ID table. (PCI DSS v1.1 8.5.5, EDSSCS0463)

There will be no group, shared, or generic accounts and passwords. (PCI DSS v1.1 8.5.8)

Location ID passwords will be changed at least every 90 days. (PCI DSS v1.1 8.5.9)

Location ID passwords will be 8 characters. (PCI DSS v1.1 8.5.10)

Passwords will contain both numeric and alphabetic characters. (PCI DSS v1.1 8.5.11)

The last 4 passwords will not be allowed for Location IDs during a password reset. (PCI DSS v1.1 8.5.12)

The Location ID password will be locked out after 5 bad attempts. (PCI DSS v1.1 8.5.13)

The lockout duration will be set to thirty minutes or until administrator enables the Location ID. (PCI DSS v1.1 8.5.14)

Appendix E – Contacts and Assistance

If a user encounters problems or needs assistance, contact the individual at your site responsible for ELIT MFT. Whenever a user has trouble transmitting data to another location, check with your local ELIT MFT support group first before assuming that something went wrong at the other end. Your local ELIT MFT support group will work with the receiver site, if necessary, to resolve the problem.

If you are unsure of who to contact please notify the ELIT MFT Service Desk for assistance at 866-214-0108 option 1,1,1 or **BESOperations@dx.com**.

Document Version History

Version No.	Version Date	Revised By	Affected Section & Description of Change
1.0	2006		Entire rewrite of user guide
1.1	06/28/2007		Updated doc for JDE071-002
1.2	08/09/2007		Updated doc for JDE071-001
1.3	02/26/2008		Updated for JDE071-039. Updated for JDE081-012/014/015. Updated for JDE081-007. Clarified 'dirempty', 'fileempty' parameters for FTPPULL's. Removed Software Code version from page 2.
1.4	05/23/2008		Updated doc for JDE081-021. Added pgp, nlst and cryptpass option for FTPPULL. Add comment for FTP PULL Mget.
1.5	10/27/2008		Updated format to meet EDS/HP standards
1.6	11/03/2008		Cosmetic changes
1.7	02/11/2009		Dir sn correction
1.8	03/05/2009		Updated for JDE091-009. Additional Dir sn commands
1.9	12/21/2009		Updated format to meet HP standards
1.10	03/04/2010		Cosmetic changes
1.11	08/12/2010		Updated Service Desk local number in Appendix F
1.12	10/14/2010		Updated for JDE091-011 & MDE091-018. Additional Dir sn commands
1.13	04/26/2011		Added ignore command to FTPPULL commands. JDE111-017
1.14	11/30/2011		Added PGP option for FTPPULL and pulled from a mailbox files. JDE111-066.
1.15	05/17/2012		Replaced ELIT with HP MFT
1.16	04/04/2014		Updated for Midrange. Removed ANX & dial-up, added PGP, Changed MVS to z/OS, Removed SSH section.
1.17	11/06/2014		Updated for JDE141-030/035/039. Revised 'Accepted' to 'Verified' when fromdata processing is being used and files are rejected. Added fromdata example when unknown ID's are encountered.
1.18	06/03/2015		Added new 014-feedback note.
1.19	10/23/2015	Julie Noelke	Changed HP MFT to ELIT MFT, Hewlett-Packard to Hewlett Packard

			Enterprise, @hp.com to @hpe.com, updated copyright.
1.20	11/06/2015	Julie Noelke	Converted document to new HPE template.
1.21	05/26/2017	Julie Noelke	Removed/Replaced references to HPE and Hewlett Packard with DXC and DXC Technology
1.22	06/10/2017	Julie Noelke	Updated copyright.
1.23	12/08/2017	Julie Biafore	Removed 248-365-2248 from Appendix E.
1.24	03/14/2019	Greg Friess	Revised failmail/emptyemail in section 8.
1.25	10/21/2020	Julie Biafore	Updated copyright.
1.26	07/20/2021	Julie Biafore	Converted to new DXC template format.