



>> EDS\*ELIT Communications Network

eds.com

# FTP Server User's Guide

How FTP Interfaces with EDS\*ELIT

Version 1.0



## Table of Contents

<b>1</b>	<b>Document Conventions .....</b>	<b>1</b>
1.1	Terminology.....	1
1.2	How to Use This Document.....	2
1.3	Feedback.....	2
<b>2</b>	<b>About EDS*ELIT.....</b>	<b>3</b>
2.1	EDS*ELIT FTP Server.....	3
2.2	FTP Client Software.....	3
2.3	ID Configuration Options.....	3
2.3.1	Mailbox ID.....	3
2.3.2	Push ID.....	4
2.4	Getting Setup with EDS*ELIT.....	4
2.5	Technical Support.....	4
2.6	Connecting to EDS*ELIT.....	4
2.6.1	Internet via a Secure Connection.....	4
2.6.2	The ANX network.....	5
2.6.3	Dial-Up TCP/IP.....	5
2.6.4	Access from within the EDS Network (EDS Internal Only).....	5
<b>3</b>	<b>File Types.....</b>	<b>5</b>
3.1	FTP File Transfer Modes.....	6
3.2	Character Set Conversions.....	6
<b>4</b>	<b>Logging on to EDS*ELIT.....</b>	<b>6</b>
4.1	IDs and Passwords.....	6
4.2	EDS*ELIT Switchers.....	7
<b>5</b>	<b>Data Elements for EDS*ELIT Transfers.....</b>	<b>7</b>
<b>6</b>	<b>Routing a File Through EDS*ELIT.....</b>	<b>7</b>
6.1	Parameters for File Delivery.....	8
6.2	Considerations.....	8
6.3	FROMDATA Option.....	8
6.4	Session Flow.....	9
6.5	Command Options.....	10
6.6	Examples of Routing Files.....	12
<b>7</b>	<b>Listing Files.....</b>	<b>14</b>
<b>8</b>	<b>Retrieving Files from Your EDS*ELIT Mailbox.....</b>	<b>16</b>
8.1	Session Flow to Retrieve All Obtainable Files.....	16
8.2	Session Flow to Retrieve a Specific File.....	16
8.3	Examples of Retrieving Files.....	16
<b>9</b>	<b>Command Reference.....</b>	<b>19</b>
9.1	cd command.....	19

9.1.1 cd FROMDATA.....	20
9.1.2 cd MAILBAG.....	21
9.2 dir command.....	21
9.3 get command.....	23
9.4 ls or nlst command.....	23
9.5 mget command.....	24
9.6 put command.....	26
9.7 quote site command (Not Applicable With SSH, <a href="#">See Appendix A</a> ).....	29
<b>10 Using EDS*ELIT to Pull Files from Another FTP Server.....</b>	<b>32</b>
<b>Appendix A - Sending Files to an MVS or ASYNC Application.....</b>	<b>34</b>
Carriage Return/Line Feed Characters – KEEPLF, STRIPLF, & STRIPSTREAM.....	34
<b>Appendix B – Feedback Files .....</b>	<b>36</b>
<b>Appendix C – SSH-Based Secure FTP (SFTP) .....</b>	<b>38</b>
<b>Appendix D– Frequently Asked Questions .....</b>	<b>39</b>

# 1 Document Conventions

## 1.1 Terminology

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.
ANX	Advanced Network eXchange
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.
EDS*ELIT ID	A unique code that identifies a user in the EDS*ELIT network, i.e., an EDS*ELIT User ID. User ID, Mailbox, and Location Code are synonymous.
ELIT	Electronic Information Transfer
ELIT Filename	A name from 1 to 10 characters assigned by the Sender that identifies the transfer operation within EDS*ELIT. 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 EDS*ELIT network, i.e., an EDS*ELIT User ID. User ID, Mailbox, and Location Code are synonymous.
Mailbox	Identifies a holding place for files sent to your EDS*ELIT ID. A mailbox ID is equivalent to an EDS*ELIT ID or Location Code.
MVS	IBM Multiple Virtual Storage Mainframe
Obtainable Files	Files in your mailbox waiting to be retrieved.
ReceiverID	EDS*ELIT ID of the receiver of a file.
RFC	Request for Comments. A document in which an Internet standard or protocol is published.
SenderID	EDS*ELIT ID of the sender of a file.
SSL	Secure Sockets Layer
Switcher	An EDS*ELIT ID mailbox that can send and receive files on behalf of one or more other EDS*ELIT IDs.
TCP/IP	Transmission Control Protocol/Internet Protocol
*THS/THD/TTR	EDS*ELIT proprietary envelope records that contain file routing information.
Trading Partner	The sending and/or receiving parties involved in exchanging electronic data.

Term	Definition
VAN	Value-Added Network
VPN	Virtual Private Network
X12	ANSI standard for EDI.

## 1.2 How to Use This Document

This document describes how to access EDS\*ELIT 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.

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

## 1.3 Feedback

Comments on this document are welcome and encouraged. Please send your feedback to [elit.services@eds.com](mailto:elit.services@eds.com).

## 2 About EDS\*ELIT

EDS\*ELIT (pronounced "E.D.S. ELITE") is the acronym for EDS Electronic Information Transfer, a highly scalable, multi-platform, multi-protocol B2B communication service suite. EDS\*ELIT 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. EDS\*ELIT supports multiple types of data file transfer: EDI, Electronic Funds Transfer (EFT), and proprietary data structures. EDS\*ELIT also supports multiple platforms and network communication protocols [Async, 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).

### 2.1 EDS\*ELIT FTP Server

The EDS\*ELIT FTP Server implements the FTP standard as documented by RFC-959. There are multiple FTP servers within the EDS\*ELIT network that support EDS\*ELIT FTP users. They run on IBM Multiple Virtual Storage (MVS) mainframes.

Certain FTP commands (such as `cd` and `pwd`) have been enhanced in EDS\*ELIT and function differently from other FTP implementations. For example, in EDS\*ELIT, 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 EDS\*ELIT. Although the EDS\*ELIT FTP Server has been enhanced in this way, standard FTP client software may be used to access EDS\*ELIT.

### 2.2 FTP Client Software

FTP client software is required to access EDS\*ELIT. Access over the Internet may also require FTP software that supports Secure Sockets Layer (SSL) RFC 4217. Refer to [Connecting to EDS\\*ELIT](#) 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 EDS\*ELIT is not supported.

### 2.3 ID Configuration Options

EDS\*ELIT FTP IDs can be configured in two different ways.

#### 2.3.1 Mailbox ID

An EDS\*ELIT 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.

### 2.3.2 Push ID

An EDS\*ELIT FTP "push" ID is one in which data is automatically delivered ("pushed") to the destination FTP server by the ELIT 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 or FTP over a VPN).
- The owner must provide a 24x7 contact so that ELIT 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.

## 2.4 Getting Setup with EDS\*ELIT

Before using the EDS\*ELIT FTP service, you must have an EDS\*ELIT ID configured for FTP use. You can request a new EDS\*ELIT ID or have an existing EDS\*ELIT ID converted by contacting EDS\*ELIT Services via e-mail, [elit.services@eds.com](mailto:elit.services@eds.com) or by filling out an *EDS\*ELIT ID Request* on the EDS\*ELIT Web site, [http://www.business-exchange.canada.eds.com/EDS\\_ELIT/](http://www.business-exchange.canada.eds.com/EDS_ELIT/).

## 2.5 Technical Support

For help implementing FTP with EDS\*ELIT, please contact EDS\*ELIT Services via e-mail, [elit.services@eds.com](mailto:elit.services@eds.com), or by telephone at 1 248 364 5976 (7:30 a.m.– 4 p.m., Monday–Friday, Eastern time).

For support with production problems, please call our 7 X 24 X 365 Help Desk at 1 866 214 0108 (U.S. and Canada) or 1 248 754 2218 (outside U.S. and Canada).

## 2.6 Connecting to EDS\*ELIT

This section describes the ways you can connect to the EDS\*ELIT FTP server.

### 2.6.1 Internet via a Secure Connection

To use the Internet, you have two options.

- The first option is to 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.

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 EDS\*ELIT using Secure FTP software, use one of the Domain Name Server (DNS) names below as instructed by EDS\*ELIT Services.

Secure FTP connections using Explicit Session Security AUTH SSL:

<ftp.elit.eds.com> or <ftp2.elit.eds.com>

Secure FTP connections using Implicit Session Security:

[ftp.elit.eds.com](http://ftp.elit.eds.com) port 990 or [ftp2.elit.eds.com](http://ftp2.elit.eds.com) port 990

- The second option is to establish a Virtual Private Network (VPN) with EDS\*ELIT. A VPN securely connects your site and the EDS\*ELIT FTP Server using firewall software and/or hardware, however, SSL is not required. The EDS\*ELIT Services group initiates the process, but the firewall administrators at each site are involved in predefining the security rules. After the rules are defined, you can access EDS\*ELIT using one of the DNS names below as instructed by EDS\*ELIT Services.

[ftp.elit.eds.com](http://ftp.elit.eds.com) or [ftp2.elit.eds.com](http://ftp2.elit.eds.com)

### 2.6.2 The ANX network

The Advanced Network eXchange (ANX) network is a private, Internet Protocol (IP)-based network dedicated for business use. This connection method requires establishing a VPN with EDS\*ELIT as described in the [Internet Via A Secure Connection](#), as well as a subscription through a third-party ANX provider.

To connect to EDS\*ELIT over the ANX network, use one of the IP addresses below as instructed by EDS\*ELIT Services.

198.132.123.196 or 198.132.123.197

198.132.123.188 or 198.132.123.189

### 2.6.3 Dial-Up TCP/IP

This option allows you to access the EDS\*ELIT FTP Server over a dial-up telephone line. Once connected, you can access the EDS\*ELIT FTP Server from your FTP client using the following DNS name:

[ftp.elit.eds.com](http://ftp.elit.eds.com)

The [Resource Library](#) on the EDS\*ELIT Web site ([http://www.business-exchange.canada.eds.com/EDS\\_ELIT/](http://www.business-exchange.canada.eds.com/EDS_ELIT/)) has instructions and telephone number for configuring a Windows computer for dial-up Transmission Control Protocol/Internet Protocol (TCP/IP).

### 2.6.4 Access from within the EDS Network (EDS Internal Only)

You can connect using standard FTP or FTP that supports SSL within the EDS\*ELIT network. Use one of the following DNS names as instructed by EDS\*ELIT Services:

[ftp.elit.eds.com](http://ftp.elit.eds.com) or [ftp2.elit.eds.com](http://ftp2.elit.eds.com)

## 3 File Types

EDS\*ELIT 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.

### 3.1 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.

### 3.2 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. EDS\*ELIT runs on IBM MVS mainframes and stores data internally in the EBCDIC character set.

Files transferred into EDS\*ELIT in ASCII mode are assumed to be in the ASCII character set, and EDS\*ELIT converts them to EBCDIC for internal use. Likewise, files outbound from EDS\*ELIT are converted from EBCDIC to ASCII.

From an ASCII platform, text files should be transferred to EDS\*ELIT 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 EDS\*ELIT successfully in ASCII mode.

Be aware that when the `fromdata` option is used (discussed in the [FROMDATA Option](#) section), EDS\*ELIT must be able to read the data in the incoming file. Therefore, an ASCII file transferred in BINARY mode will not work.

## 4 Logging on to EDS\*ELIT

### 4.1 IDs and Passwords

Your assigned EDS\*ELIT ID is the logon ID to your EDS\*ELIT mailbox. Your initial password is set to the password you enter the first time you log on. Passwords must be five to eight characters in length and should contain no special characters. EDS\*ELIT IDs and passwords are not case sensitive.

The example below illustrates logging on to the EDS\*ELIT 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 EDS\\*ELIT](#) section of this document).

```
ftp> open ftp.elit.eds.com
Connected to ftp.elit.eds.com.
220-
220-      Welcome to the EDS*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 (ftp.elit.eds.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.

## 4.2 EDS\*ELIT Switchers

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

## 5 Data Elements for EDS\*ELIT Transfers

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

EDS\*ELIT needs the following data elements to route a file to its destination.

- The EDS\*ELIT ID of who sent the file, referred to as the SenderID.
- The EDS\*ELIT ID of who is to receive the file, referred to as the ReceiverID (or destination).
- ELIT 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 EDS\*ELIT 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 EDS\*ELIT to identify a specific transmission. This distinguishes transmissions from others with the same SenderID, ReceiverID, and ELIT Filename.

## 6 Routing a File Through EDS\*ELIT

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

## 6.1 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 Filename. Your trading partner should be able to provide you with the ReceiverID and the required ELIT Filename, if any. The Reference element is optional and can be used to further identify the data.

For files sent to an MVS mainframe destination, you must also provide EDS\*ELIT 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 (`keeplf`, `striplf`, or `stripstream` parameter). See [Appendix A - Sending Files to an MVS or ASYNC Application](#) for more explanation.

## 6.2 Considerations

Using standard FTP commands, there are multiple ways to specify the file routing for EDS\*ELIT. You can instruct EDS\*ELIT 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 EDS\*ELIT ID that logged in. However, if your EDS\*ELIT ID is defined as a switcher (can send and receive files on behalf of other EDS\*ELIT IDs) you may need to specify the SenderID for each file if it is different from the EDS\*ELIT ID that logged in.

## 6.3 FROMDATA Option

When the `fromdata` option is used, EDS\*ELIT extracts the routing information, i.e., SenderID, ReceiverID, and ELIT 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 EDS\*ELIT. 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 EDS\*ELIT proprietary header and trailer records of `*THS/*THD/*TTR`, `*P*`, or `*#F/*#S/*#D`. These records contain the EDS\*ELIT 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 (`blksz`) 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 EDS\*ELIT 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, EDS\*ELIT IDs. In addition, the SenderID must be one of the following:
  - The same EDS\*ELIT ID as the one that logged in
  - Must convert to the EDS\*ELIT ID that logged in, i.e., be defined in EDS\*ELIT as an alias to the EDS\*ELIT ID that logged in
  - Defined as switching through the EDS\*ELIT ID that logged in
- Parameter values extracted from a file override those set with the `cd`, `quote site`, or `put` commands.
- The default is `striplf`. The command `cd fromdata` sets the session to `keeplf`, however, setting `fromdata` with the `quote site rid=fromdata` command does not change the setting.
- Since EDS\*ELIT 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.

## 6.4 Session Flow

This section describes what needs to occur in an FTP session to route a file through EDS\*ELIT 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 EDS\*ELIT.
- 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 EDS\*ELIT Filename) and optional parameters (Feedback Filename, Application Reference) with the appropriate commands:

```
cd command or cd fromdata
quote site command
```

However, the routing parameters can also be placed on the `put` command line. Note that the `pwd` command displays parameters previously set.

- If routing to an MVS 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 an MVS or ASYNC Application](#) 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 EDS\*ELIT, 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.

## 6.5 Command Options

This section illustrates the different command options when routing a file through EDS\*ELIT.

Assume a file is to be sent to the destination of QE5 with the ELIT Filename of DESADVT. QE5 resides on an MVS 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, keeplf
ftp> put asn.txt
ftp> pwd
ftp> quit
    - or -
ftp> type ascii
ftp> cd qe5/desadvt/./keeplf
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,keeplf
ftp> put asn.txt
ftp> pwd
ftp> quit
    - or -
ftp> type ascii
ftp> quote site lrecl=80
ftp> put asn.txt qe5/desadvt/./keeplf/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/desadvt
ftp> quote site lrecl=80, keeplf
ftp> put asn.txt
ftp> pwd
ftp> quit
  - or -
ftp> type ascii
ftp> cd /lam/qe5/desadvt/./keeplf
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=desadvt,keeplf
ftp> put asn.txt
ftp> pwd
ftp> quit
  - or -
ftp> type ascii
ftp> quote site lrecl=80
ftp> put asn.txt /lam/qe5/desadvt/./keeplf/asn.txt
ftp> pwd
ftp> quit
```

When `fromdata` is used, the routing (SenderID, ReceiverID, ELIT Filename) is extracted from the data. In addition, if EDS\*ELIT \*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
ftp> quote site keeplf
ftp> put asn.txt fromdata/asn.txt
ftp> pwd
ftp> quit
  - or -
ftp> type ascii
ftp> quote site rid=fromdata,keeplf
ftp> put asn.txt
ftp> pwd
ftp> quit
```

## 6.6 Examples of Routing Files

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

```

ftp> cd pol/release/./keeplf                               (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,keeplf (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 EDS*ELIT to get the routing from the data)
200 Site Command Accepted
ftp> pwd                                                  (display the parameters, note STRIPLF)
257 SenderId: "LAM" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo: "TCPIP" STRIPLF
ftp> cd fromdata                                          (get the routing from the data, note KEEPLF)
250 SenderId: "LAM" ReceiverId: "FROMDATA" FileName: "$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 EDS\*ELIT ID that logged in, POL, must be set-up by EDS\*ELIT 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 (206.122.41.170:(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/tdl/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.
```

## 7 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 EDS\*ELIT. This is to distinguish a particular transmission from others with the same SenderID, ReceiverID, and ELIT 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 EDS*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 (dygdoqe.sys.eds.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
```

## 8 Retrieving Files from Your EDS\*ELIT Mailbox

You can retrieve files from your EDS\*ELIT 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.

EDS\*ELIT 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.

### 8.1 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 EDS\*ELIT.
- 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 EDS\*ELIT, 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.

### 8.2 Session Flow to Retrieve a Specific File

EDS\*ELIT assigns a unique control number to each file transfer. This distinguishes a specific file from others with the same SenderID, ReceiverID, and ELIT 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 EDS\*ELIT.
- Display files with the `dir` command to retrieve the EDS\*ELIT 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 EDS\*ELIT, 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.

### 8.3 Examples of Retrieving Files

The following example session illustrates various ways to retrieve files from EDS\*ELIT. More detailed information is available in the [Command Reference](#) section.

```

220-
220-      Welcome to the EDS*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 (dygdoqe.sys.eds.com:(none)): tdl          (log in)
331 Username "tdl" needs password
Password:
230 User "TDL" 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 Aaked.
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
```

## 9 Command Reference

This section describes the commands that can be used with the EDS\*ELIT 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 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 EDS\*ELIT (to the right of the equals sign) are not case sensitive.

### 9.1 cd command

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

`cd /SenderID/ReceiverID/ELITFilename/Reference/keeplf|striplf|stripstream`

<i>SenderID</i>	Optional. 3 to 17 characters. The EDS*ELIT ID that is sending files. This value defaults to the EDS*ELIT ID that logged on. This is mostly used by EDS*ELIT IDs defined as switchers.
<i>ReceiverID</i>	Required; 3 to 17 characters. The EDS*ELIT ID to receive the files being sent or the <code>fromdata</code> or <code>mailbag</code> verb (see <code>cd fromdata</code> and <code>cd mailbag</code> ). If a <i>SenderID</i> is not specified, a "/" should not prefix the <i>ReceiverID</i> .
<i>ELITFilename</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 <code>\$default</code> .
<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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 an MVS or ASYNC Application](#) section for more information.
- If `fromdata` is used, the values extracted from the data (e.g. ReceiverID) override any value set with this command.

## Examples

Assume that EDS\*ELIT 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
```

### 9.1.1 cd FROMDATA

**cd fromdata/ELITFilename/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 EDS\*ELIT 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 EDS\*ELIT proprietary headers, such as \*THS/\*THD/\*TTR headers. The routing and certain other fields (ELIT Filename, Record Length, feedback, etc.) are taken from the headers. The SenderID and ReceiverID in the data must be predefined in EDS\*ELIT.

## 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.
- Note that this command changes the session to `keep1f` (however, setting `fromdata` with the `quote site rid=fromdata` command keeps the session in the default mode of `striplf`).

## Example

Assume that EDS\*ELIT ID TDL has issued the following command.

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

### 9.1.2 cd MAILBAG

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

**cd mailbag/ELITFilename/Reference/keep1f|striplf|stripstream**

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

## Example

Assume that EDS\*ELIT ID TDL has issued the following command.

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

## 9.2 dir command

The `dir` command displays files in your mailbox waiting to be picked up, as indicated by the OBTAINABLE status.

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

<i>no parameter</i>	This is the default display of <i>File Permissions, SenderID, ELITFilename, File Size, Deliver Date, and DataSetName</i> for files sent to you ( <i>ReceiverID</i> ).
<i>s=SenderID</i>	Optional. Displays files sent to you with the <i>SenderID</i> specified. A wildcard value can be used within the <i>SenderID</i> value.
<i>cc=ReceiverID</i>	Optional. This is used by EDS*ELIT IDs that are defined as switchers. It displays files for only the <i>ReceiverID</i> specified. <i>sn</i> can be used with this parameter, i.e., <code>dir sn,cc=ReceiverID</code> . A wildcard value can be used within the <i>ReceiverID</i> value.

<i>f=ELITFilename</i>	Optional. Displays files sent to you with the <i>ELITFilename</i> specified. A wildcard value can be used within the <i>ELITFilename</i> value.
<i>r=Reference</i>	Optional. Displays files sent to you with the <i>Reference</i> specified. A wildcard value can be used within the <i>Reference</i> 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 <i>DataSetName</i> specified. A wildcard value can be used within the <i>DataSetName</i> value.
<i>sn</i>	Optional. Displays <i>ELITFilename</i> , <i>Status</i> , <i>SenderID</i> , <i>ReceiverID</i> , <i>Control#</i> , <i>Reference (last 5 characters)</i> , and <i>Deliver Date</i> for files sent to you.
<i>fd</i>	Optional. Displays the last file sent from the current session with the destination of <i>FROMDATA</i> .

## Notes

- Files sent to yourself (loopback files) will not display unless you issue a "cd *EDS\*ELIT ID*" command, where *EDS\*ELIT 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) Acked.
ftp: 127 bytes received in 0.00Seconds 127000.00Kbytes/sec.
```

### 9.3 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.
```

### 9.4 ls or nlst command

This command displays a brief list of files waiting to be picked up. Only the *DataSetName* is displayed.

`ls s=SenderID, f=ELITFilename, 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.

*f=ELITFilename* Optional. Displays files sent to you with `OBTAINABLE` status and the given *ELITFilename* value. A wildcard value can be used within the *ELITFilename* value.

*c=Control#* 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.
```

## 9.5 mget command

The `mget` command retrieves all files from your mailbox.

EDS\*ELIT 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=ELITFilename, r=Reference, d=DataSetName
```

*	Required if no other parameters are specified. Retrieves all files waiting to be picked up. When issued by an EDS*ELIT ID defined as a switcher, files with the status of OBTAINABLE for all subordinate EDS*ELIT IDs are retrieved.
sn	Optional. Create <i>DataSetNames</i> of <i>ELITFilename</i> + <i>SenderID</i> + <i>Reference</i> + <i>DataSetName</i> + <i>Control#</i> . If not specified, the default <i>DataSetName</i> is <i>DataSetName</i> + <i>Control#</i> .
s= <i>SenderID</i>	Optional. Retrieve only those files with the specified <i>SenderID</i> . A wildcard value can be used within the <i>SenderID</i> value.
cc= <i>ReceiverID</i>	Optional. Retrieves files with the OBTAINABLE status for that <i>ReceiverID</i> , only. This is used by EDS*ELIT IDs that are defined as switchers. A wildcard value can be specified within the <i>ReceiverID</i> value.
f= <i>ELITFilename</i>	Optional. Retrieve only those files with the given <i>ELITFilename</i> . A wildcard value can be specified within the <i>ELITFilename</i> value.
r= <i>Reference</i>	Optional. Retrieve only those files with the specified <i>Reference</i> . A wildcard value can be used within the <i>Reference</i> value.
d= <i>DataSetName</i>	Optional. Retrieve only the file with the given <i>DataSetName</i> . A wildcard value can be used within the <i>DataSetName</i> 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 A
cked.
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 A
```

```

cked.
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 TCPIP 11/29/2005 20:41
WKLYORDERS OBTAINABLE JAZ LAM 16621 02029 11/29/2005 20:24
226 Transfer completed successfully. Data connection closed.
ftp: 218 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.
ftp: 70 bytes received in 0.01Seconds 7.00Kbytes/sec.

```

## 9.6 put command

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

```

put Localfile /SenderID/ReceiverID/ELITFilename/Reference/keep|strip|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 EDS*ELIT ID that is sending files. This value defaults to the EDS*ELIT ID that logged on. This is mostly used for EDS*ELIT IDs defined as switchers.
<i>ReceiverID</i>	Optional. 3 to 17 characters. The EDS*ELIT ID to receive the files being sent or the fromdata or mailbag verb (see <i>cd FROMDATA</i> , <i>cd MAILBAG</i> ). If a <i>SenderID</i> is not specified, a "/" should <u>not</u> prefix the <i>ReceiverID</i> .
<i>ELITFilename</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 <code>\$default</code> . If this parameter is specified, the <i>ReceiverID</i> must also be specified.
<i>Reference</i>	Optional. 1 to 35 alphanumeric characters. The <i>Reference</i> 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 <i>ELITFilename</i> must also be specified.
<code>keeplf</code>	Optional. This qualifier keeps carriage return/line feed characters in your files. Please refer to the <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <i>Notes</i> below). This is the name to assign the file at its destination.

## Notes

- 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`, `ELITFilename`, `Reference`, `keeplf|striplf|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.
- Unlike the `cd` command, the `put` command does not change the session to `keeplf`, but maintains the existing mode.
- When `fromdata` is used, the values most commonly extracted from the data (`SenderID`, `ReceiverID`, `ELIT filename`) override the parameters on the `put` command line.
- 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
```

EDS\*ELIT 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 Filename is DESADVT; DataSetName is asn.dat.

A fully qualified `put` command is shown below. It routes the file DATA.DAT to EDS\*ELIT using the *SenderID* of TD1, the *ReceiverID* of LAM, the *ELITFilename* 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
```

EDS\*ELIT 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 EDS\*ELIT ID of TDL, assigning it the EDS\*ELIT Filename of SALES. KEEPLF 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 td1/sales/./keep1f/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: "." KEEPLF SALES.
TDL...novsales.dat Acked.
```

```
ftp> quote site rid=td1,fnm=sales,keep1f                (set parameters for the transfer)
200 Site Command Accepted
ftp> pwd                                                  (display parameters)
257 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "." KEEPLF
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: "." KEEPLF SALES.
TDL...novsales.dat Acked.

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

ftp> cd tdl/sales                             (set parameters for the transfer)
250 SenderId: "POL" ReceiverId: "TDL" FileName: "SALES" RefNo: "TCPIP" STRIPLF
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" KEEPLF SA
LES.TDL.TCPIP.novsales.dat Acked.

```

## 9.7 quote site command (Not Applicable With SSH, [See Appendix A](#))

Specifies EDS\*ELIT parameters for subsequent transfers.

```

quote site sid=SenderID,rid=ReceiverID,fnm=ELITFilename,ref=Reference,
          lrecl=RecordLength,keeplf|striplf|stripstream,fb=ELITFilename,
          nat=IPAddress,prefix=DataSetName,suffix=gdg|Suffix,mindsn,
          chunk=RecordLength

```

<i>sid=SenderID</i>	Optional. 3 to 17 characters. The EDS*ELIT ID that is sending files. This value defaults to the EDS*ELIT ID that logged on. This is mostly used by EDS*ELIT IDs defined as switchers.
<i>rid=ReceiverID</i>	Optional. 3 to 17 characters. The EDS*ELIT ID to receive the files being sent or the fromdata or mailbag verbs (see <i>cd fromdata</i> , <i>cd mailbag</i> ). If a <i>SenderID</i> is not specified, a "/" should not prefix the <i>ReceiverID</i> .
<i>fnm=ELITFilename</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>ref=Reference</i>	Optional. 1 to 35 alphanumeric characters. The reference string to be assigned to any files sent. This value is optional unless <i>keeplf</i> , <i>striplf</i> , or <i>stripstream</i> is specified, in which case a "." can be used here as a placeholder
<i>lrecl=RecordLength</i>	Optional. This specifies the logical record length of the file, which is a value from 1 to 32760. This value is needed when the

Receiver is on an MVS mainframe and is expecting a file with fixed-length records.

<code>keeplf</code>	Optional. This qualifier keeps carriage return/line feed characters in your files. Please refer to the <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> 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 <a href="#">Appendix A - Sending Files to an MVS or ASYNC Application</a> section for more information. The default mode is <code>striplf</code> .
<code>fb=ELITFilename</code>	Optional. Requests a feedback file indicating the status of the file sent. See <a href="#">Appendix B – Feedback Files</a> for more information on Feedback Files.
<code>nat=IP-Address</code>	Optional. This is the Network Address Translation (NAT) address of the EDS*ELIT 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> <i>DataSetName</i>	Optional. Dataset name prefix to assign to files retrieved with the <code>mget</code> command. The <i>DataSetName</i> is formed as <i>prefix.suffix&lt;control#&gt;</i> ( <i>control#</i> is 5 digits). If no <i>suffix</i> is given, the default is <code>fst</code> . If <code>suffix=gdg</code> is given, the <i>DataSetName</i> is formed as <i>"prefix(+1)"</i> . The <code>."</code> character is allowed in the <i>DataSetName</i> . See the <i>Examples</i> at the end of this section.
<code>suffix=Suffix</code> or <code>GDG</code>	Optional. Keyword of <code>gdg</code> or the string to use as the suffix of the dataset name. The <i>DataSetName</i> is formed as <i>prefix.suffix&lt;control#&gt;</i> ( <i>control#</i> is 5 digits). If no <i>suffix</i> is given, it defaults to <code>fst</code> . If <code>suffix=gdg</code> , is given, <code>(+1)</code> is used as the suffix, and the <i>DataSetName</i> is formed as <i>"prefix(+1)"</i> . See the <i>Examples</i> at the end of this section.
<code>mindsn</code>	Optional. Requests a minimal dataset name as a reply to <code>ls</code> or <code>mget</code> . The original dataset name is replaced by <code>fstxxxxx</code> , where <code>xxxxx</code> is the control number of the entry. This is used for platforms such as the AS400.
<code>chunk=RecordLength</code>	Optional. EDS*ELIT inserts carriage return/line feed characters at the record length given for files retrieved with the <code>get</code> and <code>mget</code> commands. Used for platforms such as the AS400.

## Notes

- Any values extracted with the `fromdata` option override those specified with this command.

## 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 Acked.

ftp> quote site rid=fromdata
200 Site Command Accepted

ftp> pwd
257 SenderId: "TDL" ReceiverId: "FROMDATA" FileName: "$DEFAULT" RefNo: "TCPIP" S
TRIPLF
```

## 10 Using EDS\*ELIT to Pull Files from Another FTP Server

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

An EDS\*ELIT ID has to be defined by EDS\*ELIT Services for the FTP server you want to access. The pull function is initiated by sending a file to that EDS\*ELIT ID with an ELIT 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 nodelete, lrecl=RecordLength, as1.5
site dirempty ERROR 550 Mailbox Empty
cd RemoteDirectory
mget *
get RemoteFileName
```

Minimally, the trigger file must contain a USER command and either a get or mget command; the other commands are optional. 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*, *ELITFilename*, *Reference*, *striplf*, *keep1f*, and *lrecl*.

USER <i>Userid/Password</i>	Specifies the logon <i>UserId</i> and <i>Password</i> on the remote system.
binary	This parameter will set the transfer mode to binary. If not specified, the transfer mode is ASCII.
nodelete	By default, EDS*ELIT 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 EDS*ELIT "delete" command results in error. This parameter removes the delete command from the dialogue.
failmail= <i>eMailId</i>	Specifies the <i>eMailId</i> to receive notice of a problem in the "pull" operation.
emptymail= <i>eMailId</i>	Specifies the <i>eMailId</i> to receive notice of a "pull" attempt that did not find any files on the remote FTP server.
dirempty <error string>	Used to specify the <error string> exception message returned by the remote FTP server when a "pull" attempt did not find any files. If the remote server exception message matches the beginning of the <error string>, then EDS*ELIT will positively acknowledge the FTTPULL file, otherwise the FTTPULL file will be

NAK'ed and a retry will be attempted in 5 minutes (i.e. 'dirempty ERROR 550').

<code>cd <i>RemoteDirectory</i></code>	Specifies the remote directory from which to "pull" files. If not specified, files are pulled from the default directory of the logon <i>UserId</i> .
<code>mget *</code>	Indicates that all files in the remote directory should be retrieved.
<code>get <i>RemoteFileName</i></code>	Indicates to retrieve the single file specified by <i>RemoteFileName</i> .
<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. EDS*ELIT will decrypt the data in a staging area and verify the signature prior to "pulling" the data into the EDS*ELIT network. The <code>requeue</code> function described below is designed to recover files that experience certain errors in the as1.5 processing.
<code>requeue</code>	This function is used in conjunction with the as1.5 protocol and directs EDS*ELIT 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 EDS\*ELIT, ...). 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.

## Appendix A - Sending Files to an MVS or ASYNC Application

Because of differences in computer platforms, you may need to specify parameters that allow EDS\*ELIT to provide a native text file to a receiving MVS (mainframe) or ASYNC application. This is done by indicating the logical record length required by the receiving application, and how EDS\*ELIT should handle carriage return/line feed characters in your data. The latter depends on the type of text file you are sending to EDS\*ELIT, 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 EDS\*ELIT 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 EDS\*ELIT, not files you retrieve from EDS\*ELIT.

`keeplf`, `striplf`, and `stripstream` work in conjunction with the `quote site lrecl=n` command, where *n* is the record length required by the receiving MVS 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`.

#### 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 EDS\*ELIT 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.

```

ABCDEF<cr><lf>
HIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnop<cr><lf>
mnopqrstuvwxyz

```

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

```

ABCDEFGHIJKLMN<cr><lf>
OPQRSTUVWXYZ0123456789abcdefghijklmnop<cr><lf>
qrstuvwxyz^

```

## Appendix B - Feedback Files

Feedback files are files sent to you containing delivery status information on specific files you sent to EDS\*ELIT 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=ELITFilename` command, where *ELITFilename* is the name of the feedback file to be sent to your EDS\*ELIT 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:	EDS*ELIT 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:	EDS*ELIT 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 HOLD
003-FILE CANCELED BY OPERATOR
004-FILE PURGED BY OPERATOR
005-FILE DELIVERED ON TIME
006-FILE DELIVERED LATE
007-FILE NOT DELIVERED, PURGED
008-FILE UNLOADED INTO DD DUMMY(DISCARDED)
009-FILE UNLOADED INTO "LOST & FOUND" FNT ENTRY
010-FILE MAILBOXED, WAITING FOR PICKUP
011-FILE NOT DELIVERED
012-FILE DELIVERED OFFNET TO location
013-FILE PICKED UP BY USER

```

The most common messages are described below.

```

001-FILE UNLOADED INTO DSN- Dataset name
The file reached its destination on an MVS 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 EDS\*ELIT ID

The file was sent to another VAN or was retrieved by an EDS\*ELIT ASYNC user.

013-FILE PICKED UP BY USER

The file was destined to an EDS\*ELIT "pull" mailbox and was retrieved.

## Appendix C - SSH-Based Secure FTP (SFTP)

FTP commands for the EDS\*ELIT SSH-based secure FTP server are the same as those described in this document with the exception of the `site` (`quote site`) command. This command is not supported by the SSH Internet-Draft. Therefore, all routing and special qualifiers must be specified with the `cd` and `put` commands. For this reason, the `cd` command has been enhanced to support the logical record length (`lrecl`) qualifier, a qualifier primarily used for files destined to mainframe applications.

### **cd** command

The SFTP `cd` command indicates the `lrecl` qualifier in addition to *SenderID*, *ReceiverID*, *ELITFilename*, *Reference*, and a data format qualifier of `keeplf` or `striplf`. Otherwise, the `cd` command behaves the same as described in the [cd Command](#) section. The particular command syntax for SFTP is:

```
cd /SenderID/ReceiverID/ELITFilename/Reference/keeplf|striplf/lrecl=RecordLength
```

See the [Appendix A – Sending Files to an MVS or ASYNC Application](#) for descriptions of the `keeplf` and `striplf` qualifiers, and the [cd Command](#) section for descriptions of the other qualifiers. Note that *ReceiverID* is the only required parameter, and `striplf` is the default mode (`stripstream` is not supported).

### Examples

- |   |  |
|---|--|
| <pre>cd pol/pay/./striplf/lrecl=120</pre> | <p>The next <code>put</code> command will deliver data from the EDS*ELIT ID that logged in to <code>pol</code>, with an <i>ELITFilename</i> of <code>PAY</code>; the <i>Reference</i> of <code>."</code> is used as a placeholder. The data will be formatted into fixed-length records of 120 bytes and in accordance with the rules of the <code>striplf</code> qualifier.</p> |
| <pre>cd /tdl/pol/aaadata</pre>            | <p>The next <code>put</code> command will deliver data from <code>tdl</code> to <code>pol</code>, with an <i>ELITFilename</i> of <code>aaadata</code>, a <i>Reference</i> of <code>TCP/IP</code>, and will use the data format qualifier of <code>striplf</code>. Because <i>SenderID</i> is specified, it is not the EDS*ELIT ID that logged in.</p>                            |
| <pre>cd fromdata</pre>                    | <p>The next <code>put</code> command will deliver data using the <code>fromdata</code> option, which extracts routing information directly from the data. <code>fromdata</code> defaults to <code>keeplf</code>.</p>   |

## Appendix D- Frequently Asked Questions

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

A: When you send a file with 80 byte records to an MVS mainframe application, 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 application 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 EDS\*ELIT ID and password.
  - If you are connecting over the Internet and a VPN is not established, ensure you are using secure software.
- 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 document are valid and predefined in the EDS\*ELIT Network. If `fromdata` is not used, ensure a valid EDS\*ELIT `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`, `striplf`, `stripstream`, `lrecl`) or with a different EDS\*ELIT 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 EDS\*ELIT cannot exchange non-EDI data files.

*Q: Does EDS\*ELIT support the EBCDIC or ASCII character set?*

A: EDS\*ELIT stores data internally in the EBCDIC character set. Text files inbound to EDS\*ELIT are assumed to be ASCII files and are converted to EBCDIC for the EDS\*ELIT mainframe. Likewise, files outbound from EDS\*ELIT are converted from EBCDIC to ASCII. Any file that should not be converted, such as compressed or encrypted files, should be transferred to EDS\*ELIT 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 EBCDIC files.

*Q: Does EDS\*ELIT support non-EDI data?*

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

*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`.