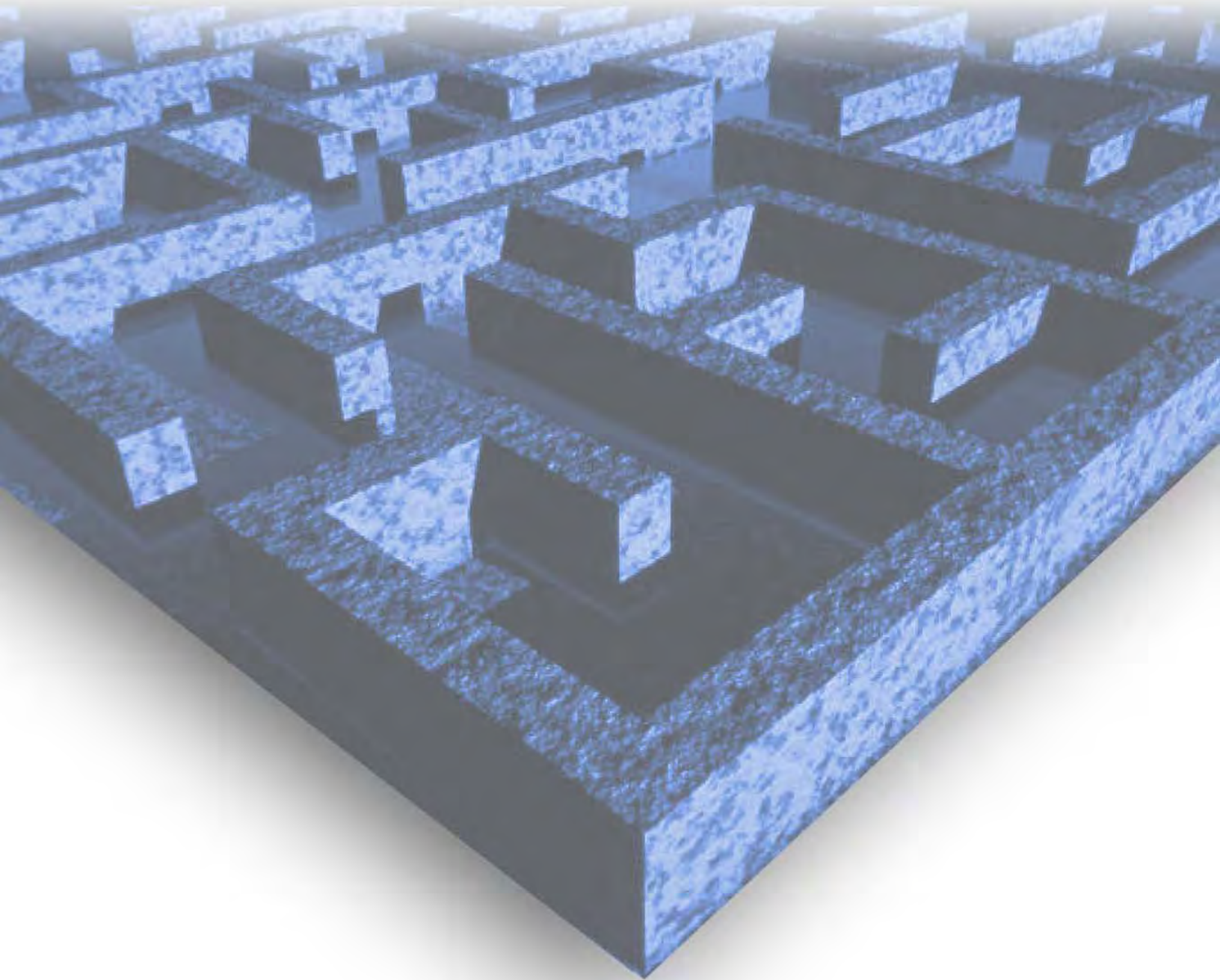


TCP/IP for VSE



CSI

INTERNATIONAL

Copyright © 1996–2013 by CSI International

All Rights Reserved

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

This material contains confidential and proprietary material of Connectivity Systems, Inc. (hereafter referred to as *CSI International* and *CSI*) and may not be used in any way without written authorization from CSI International. This material may not be reproduced, in whole or in part, in any way, without prior written permission from CSI International.

Permission is hereby granted to copy and distribute this document as follows:

- Each copy must be a complete and accurate copy.
- All copyright notices must be retained.
- No modifications may be made.
- Use of each copy is restricted to the evaluation and/or promotion of CSI International's TCP/IP FOR VSE product or in accordance with a license agreement.

TCP/IP FOR VSE User Guide
Version 1 Release 5F
March 2013

Published by CSI International
8120 State Route 138
Williamsport OH 43164
Phone: 800-795-4914
Fax: 740-986-6022

Internet: <http://www.csi-international.com>

E-mail: info@csi-international.com

Technical Support: support@csi-international.com

Documentation comments: documentation@csi-international.com

CSI International Technical Support and Technical Support Escalation

CSI Product Technical Support

During Business Hours Monday through Friday, 9:00 A.M. through 5:00 P.M. EST/EDT

- Telephone

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

- E-mail

Questions: support@csi-international.com

- Web

http://csi-international.com/problemreport_vse.htm

**Emergency Service
(24/7)** After business hours, and 24 hours on Saturday and Sunday

- Telephone¹

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

Escalating Technical Issues

Business Hours Monday through Friday, 9:00 A.M. through 5:00 P.M. EST/EDT

Request support escalation by the telephone numbers or e-mail address listed below.

1. Calls are automatically transferred to a 24/7 answering service that forwards all calls based on severity.

- Telephone

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

- E-mail

Escalation: support@csi-international.com

After Hours (SEV1)

1. Call one of the following telephone numbers.

Toll Free in the USA: 800-795-4914

Worldwide: 740-420-5400

2. Tell the operator that you have a “SEV1” issue.

The answering service will follow the following chain:

- a. Technical Support Rep(s)
- b. Product Developer (if necessary)
- c. Technical Support Manager (if necessary)
- d. Development Manager (if necessary)
- e. Chief Operating Officer (if necessary)
- f. Chief Executive Officer (if necessary)

3. Calls continue until the best person to resolve your issue is contacted.

Updates

The following tables identify updates to the *TCP/IP for VSE User Guide*. These updates are associated with incidents in the CSI support database.

Recent changes on each page are indicated by a change bar in the margin.

Published March 2013			
Change ID	Topic	Description	Page(s)
F007163	FTP	Updated the MAXACTIVE parameter description to include the default value (16)	2.24
F007178	FTP	Updated the SETVAR command description to explain how VSE SETPARM variables must be assigned with SETVAR before they can be used in FTPBATCH commands	2.34 , 2.99
F007172	FTP	Updated the SETVAR command description to specify a variable's maximum length (49 characters)	2.99
F007168	EMAIL	Updated the ATTACH and SEND command descriptions to add autonomous-file support	6.20 , 6.26

Published 7/8/2011			
Change ID	Topic	Description	Page(s)
F007100	FTP	Added note explaining the correct use of the continuation character (-) in batch files	2.22
F007090	FTP	Clarified the SET IGNORERR command description for FTPBATCH	2.30
F007105	FTP, LPR, EMAIL	Updated the variables table for auto event processing in FTP and added similar tables for LPR and EMAIL	2.39 , 4.7 , 6.8
F007092	FTP	Added table to indicate applicable record formats for each autonomous file type	2.48
F007096	FTP	Added note on the LOCAL_DLBL command	2.48
F007093	LPR	Added SET command descriptions related to formatting control	4.17–4.70
F007076	LPR	Updated PRINT command: multiple reports	4.28
F007104	PING	Added note explaining how to select a stack ID when running ping, traceroute, or DISCOVER as a CICS transaction	5.4
F007101	EMAIL	Corrected the SET ATTACHFAIL description	6.29
F007106	EMAIL	Corrected the SET DISP description	6.35

Published 10/27/2010			
Change ID	Topic	Description	Page(s)
F007081	FTP	Added note: use SET commands before LOPEN	2.27
F007087	FTP	Added new section on autonomous file syntax	2.47–2.52
F007082	FTP	Added note on BIM-EPIC-controlled autonomous files: the <i>recfm</i> , <i>lrecl</i> , and <i>blksize</i> values must be omitted from the syntax	2.48

Published 7/13/2010			
Change ID	Topic	Description	Page(s)
F007078	FTP	Updated the FTPBATCH parameter and SET command descriptions	2.23 – 2.33
F007066	FTP	Clarified the SET EXTRADAT description and the related message ID	2.29
F007070	FTP	Added a description of SET TELNTRAN, which sets the control-connection translate table in FTPBATCH jobs	2.33
F007067	LPR	Added descriptions of SET INFOPRINT, SET FORMDEF, and SET PAGEDEF commands	4.47 , 4.49 , 4.54 , 4.68
F007068	EMAIL	Clarified the valid values for QUIET=YES and other EXEC EMAIL parameters	6.13 , 6.13
F007073	EMAIL	Added a note listing disallowed EOD characters in the TEXT command when calling the EMAIL client from within a REXX program	6.28
F007074	REXEC	Added a description of the SET TIMEOUT command, which may be used to increase the connection timeout period	7.7 , 7.9

Contents

CSI International Technical Support and Technical Support Escalation	i
Updates.....	iii
Chapter 1: Getting Started	1.1
Introduction.....	1.1
Chapter 2: FTP	2.1
Protocol Background.....	2.1
File Transfer Concepts	2.3
Ports	2.4
UNIX Emulation Mode	2.5
Entering UNIX Mode	2.6
Data Translation	2.7
Translation Constraints	2.7
VSE Directory Output	2.8
Root Directory	2.9
Library Files	2.11
VSE Sublibraries	2.12
POWER Files	2.13
VSAM Catalogs	2.14
HFS Files	2.15
Mechanics of Transferring a File	2.15
Diagnosing FTP Hangs	2.16
FTP as a Client on VSE	2.17
Internal FTP Client.....	2.18
Starting the Client	2.18
Parameters	2.19
UPSI Switches	2.20

Command Variables	2.20
Return Codes	2.20
External FTPBATCH Client.....	2.22
Starting the Client	2.22
Parameters	2.23
UPSI Switches	2.26
Controlling the External FTPBATCH Client	2.27
SET Commands	2.27
Command Variables	2.34
Return Codes	2.34
Using REXX to Run the External FTPBATCH Client	2.36
Automatic FTP Client	2.37
Setup	2.38
Command Variables	2.39
Return Codes	2.42
Queue Entry Disposition	2.42
Interactive FTP Client.....	2.43
Example 1	2.44
Example 2	2.45
Usage Notes	2.45
VSE FTP Client Command Descriptions.....	2.46
VSE File Names	2.46
Fully Qualified Name	2.46
Partially Qualified Name	2.47
Autonomous Files	2.47
Example 1	2.49
Example 2	2.50
Example 3	2.51
Example 4	2.52
FTP Client Commands	2.52
ACCT	2.53
ACTIVE	2.53
ADAT	2.53
APPEND	2.54
ASCII	2.55
AUTH	2.56
BATCH	2.56
BINARY	2.56
BYE	2.57
CD	2.57
CDUP	2.58
CLOSE	2.58
CONF	2.59
DEBUG	2.60
DELETE	2.60

DIR	2.60
DUMPVAR	2.61
EBCDIC	2.61
ENC	2.62
ERASE	2.62
EVNTABND	2.63
EVNTDEBG	2.63
EVNTINIT	2.63
EVNTKILL	2.63
EVNTREPT	2.63
EXECUTE	2.63
FEAT	2.65
GET, GETX	2.65
HELP	2.66
IMAGE	2.67
LACCT	2.67
LADAT	2.67
LAPPEND	2.68
LAUTH	2.69
LCD	2.69
LCDUP	2.70
LCLOSE	2.70
LCONF	2.71
LDELETE	2.71
LDIR	2.72
LENC	2.73
LERASE	2.74
LFEAT	2.74
LGOTOEOJ	2.74
LMIC	2.74
LMKDIR	2.75
LNLIST, LNLST	2.75
LNOOP, LNOP	2.75
LOPEN	2.75
LPASS	2.76
LPBSZ	2.76
LPROT	2.77
LPWD	2.78
LQUOTE	2.79
LRENAME	2.79
LRMDIR	2.80
LSITE	2.80
LSTATUS	2.80
LSYSTEMS	2.81
LUSER	2.81
MGET	2.82

MIC	2.84
MKDIR	2.84
MODE	2.85
MPUT	2.86
NLIST, NLST	2.88
NOOP, NOP	2.88
NOPAD	2.89
NOUPRMPT	2.89
OPEN	2.89
PAD	2.90
PASS	2.91
PASSIVE	2.91
PBSZ	2.92
PROT	2.92
PUT, PUTX	2.94
PWD	2.95
QUIT	2.95
QUOTE	2.96
RENAME, RENAMEX	2.97
RENX	2.98
RETRIEVE	2.98
RMDIR	2.98
SETVAR, SETVAREX	2.98
SITE	2.100
STATUS	2.100
STORE	2.101
STRUCTURE	2.101
SYSTEMS	2.102
TERSE	2.103
TYPE	2.103
USER	2.103
VERBOSE	2.104
XGOTOEOJ	2.104
FTP as a Server on VSE.....	2.105
FTP Server Commands and Replies on VSE	2.105
Reply Codes	2.106
VSE FTP Server Commands	2.107
VSE Server Command Replies	2.109
SITE Commands	2.112
Entering SITE and LSITE Commands	2.112
Entering QUOTE and LQUOTE Commands	2.112
Categories of SITE Commands	2.113
Session Control	2.114
System Control	2.114
SITE Command Summary	2.114

Command Precedence	2.118
SITE Command Details	2.119
SITE ALLOWABORT	2.119
SITE BLKSIZE, SITE LRECL, SITE RECFM	2.119
SITE CC	2.123
SITE CRLF	2.124
SITE CLOSE	2.124
SITE DIAGNOSE	2.125
SITE DUPKEY	2.125
SITE EJECT	2.125
SITE EXTYPES	2.125
SITE FCB	2.125
SITE FCBPREFIX	2.125
SITE FIX, SITE FIXRECFM	2.126
SITE FTPMODEL	2.126
SITE FULLBUFFER	2.127
SITE HIGHKEY	2.127
SITE JSEP	2.127
SITE LDEST, SITE PDEST, SITE XDEST	2.127
SITE LEADZERO	2.128
SITE LOWKEY, SITE HIGHKEY	2.129
SITE MODELNAME	2.129
SITE NAT	2.129
SITE NOBLANK	2.130
SITE NOEJECT	2.130
SITE NULLRECORD	2.130
SITE PADDING	2.130
SITE PASSWD	2.131
SITE PDF	2.131
SITE power-command	2.131
SITE PROCDATA	2.132
SITE RECCR	2.132
SITE RECLF	2.133
SITE RECORD_CONTINUE	2.133
SITE SOSI	2.133
SITE STRIP	2.135
SITE TERSE	2.135
SITE TRANSLATE	2.135
SITE TRCC	2.136
SITE UNIX	2.137
SITE WTO	2.138
Chapter 3: Telnet and TN3270	3.1
Overview	3.1
Note About Color	3.2

Interacting with a TN3270 Client	3.3
Caution About Closing a Telnet Session	3.3
Using the CICS Telnet Client	3.4
Starting the Telnet Client	3.4
Using the Batch Telnet Client	3.6
Interactions	3.7
Return Codes	3.7
Checking Command Execution	3.8
Chapter 4: Printing Files with LPR	4.1
Overview	4.1
Setting Up the LPR Client	4.2
Running the LPR Client	4.3
Using CICS LPR	4.3
Using AUTOLPR	4.4
DEFINE EVENT	4.4
POWER JECL Statements	4.5
LPR Scripts	4.6
Command Variables	4.7
AUTOLPR Output	4.10
AUTOLPR Example 1	4.10
AUTOLPR Example 2	4.10
AUTOLPR Example 3	4.11
Using Batch LPR	4.11
Parameters	4.12
Commands	4.12
Return Codes	4.12
Generating the Batch LPR Job	4.13
Generating LPR Separator Pages	4.15
Completing an LPR Transaction	4.16
Using LPR Client Commands	4.17
* comment	4.21
/. label	4.21
CD, CDUP	4.22
DIR, LS	4.22
DUMPDATA, DUMPFCB, DUMPINSERTS, DUMPVARS	4.23
EXEC	4.24
GOTO	4.24
IF	4.25
INITVAR	4.26
LONG, SHORT	4.26
NOP	4.27
PRINT	4.28

PWD	4.30
QUERY OPTIONS	4.30
QUIT	4.30
SAY	4.30
SETVAR	4.31
START	4.32
SET ALIGN	4.32
SET ASA	4.32
SET BADPDF	4.33
SET BANNER	4.33
SET BLANK	4.34
SET CC	4.34
SET CHANNELx	4.35
SET CHANNELONLY	4.36
SET CLASS	4.36
SET CONTROL	4.37
SET COPIES	4.37
SET CRLF	4.38
SET DATAFILE	4.39
SET DEBUG	4.40
SET DEFAULT_INSERTS	4.40
SET DELREQ	4.41
SET DISP	4.41
SET DYNCONTROL	4.42
SET EOD	4.43
SET EOF	4.44
SET EOL	4.44
SET EXTRA	4.45
SET FCB	4.45
SET FCBPREFIX	4.46
SET FILENAME	4.47
SET FNO	4.47
SET FORMDEF	4.47
SET HOST, SET IP, SET IPADDR	4.48
SET INFOPRINT	4.49
SET INSERTS	4.50
SET JOBNAME	4.51
SET JSEPCOUNT	4.51
SET LASTCOLUMN	4.52
SET LEAVEBLANK	4.52
SET NOEJECT	4.53
SET OPTION	4.53
SET PAGEDEF	4.54

SET PAUSE	4.54
SET PCL5	4.55
SET PDF	4.55
SET PRINTER	4.56
SET RECFM, SET BLKSIZE, SET LRECL	4.57
SET REPEAT	4.59
SET RPORT, SET LPORT	4.60
SET SENDBLENGTH	4.60
SET SKIP0	4.61
SET SKIP1	4.61
SET SKIP2	4.62
SET SKIP3	4.62
SET SOD	4.62
SET SOSI	4.63
SET TITLE	4.64
SET TOF	4.64
SET TRANSLATE	4.65
SET USER, SET PASSWORD	4.66
SET VALIDATE	4.67
Using LPR with InfoPrint Printers.....	4.68
Using an FCB to Control Printing.....	4.69
Controlling Printers with the INSERTS Phase	4.71
What is PCL?	4.71
Specifying PCL Commands	4.72
INSERTS Macro	4.72
INSERTS Example 1	4.73
INSERTS Example 2	4.74
Chapter 5: Ping, Traceroute, DISCOVER Clients	5.1
Overview	5.1
Ping	5.2
Incoming Ping	5.2
Outgoing Ping	5.2
DISCOVER.....	5.3
Running CICS Clients.....	5.4
Ping	5.4
Traceroute	5.5
DISCOVER	5.6
CICS Command Mode	5.6
Running Batch Clients	5.8
Sample JCL	5.8
Return Codes	5.9

Chapter 6: TCP/IP for VSE EMAIL	6.1
Overview	6.1
Setting Up the EMAIL Client	6.3
Running the EMAIL Client.....	6.4
Using CICS EMAIL	6.4
Using AUTOEMAIL	6.5
DEFINE EVENT	6.5
JECL Statements	6.6
EMAIL Scripts	6.8
Command Variables	6.8
AUTOEMAIL Output	6.11
AUTOEMAIL Example	6.11
Using Batch EMAIL	6.12
VSE Name Definition	6.12
Batch Example	6.12
Parameters	6.13
Commands	6.13
Return Codes	6.14
Sending E-mail.....	6.15
Using EMAIL Client Commands	6.16
ATTACH	6.20
CD	6.21
DIR, LS	6.23
E, EXEC	6.24
HELP	6.24
PWD	6.25
QUERY OPTIONS	6.25
QUIT	6.25
RPORT	6.25
SEND	6.26
TEXT	6.28
SET ATTACHFAIL	6.29
SET AUTH	6.30
SET BINARY	6.30
SET BLINDLIST	6.30
SET BLKSIZE, SET LRECL, SET RECFM	6.31
SET CC	6.33
SET CHECKNAME	6.34
SET COPYLIST	6.34
SET CRLF	6.35
SET DEBUG	6.35
SET DISP	6.35
SET DUPES	6.36
SET ECHO	6.37

SET EOJEXEC	6.37
SET EXTTYPES	6.38
SET FCB	6.39
SET FCBPREFIX	6.40
SET FILETYPE	6.40
SET FROM	6.41
SET HOST, SET IPADDR, SET IP	6.42
SET INSERTS	6.43
SET JOIN	6.44
SET LANGUAGE	6.44
SET LOCK	6.46
SET LPASS	6.46
SET LUSER	6.46
SET MAILSERVER	6.47
SET NEWNAME	6.47
SET NOEJECT	6.48
SET NOTIFY	6.48
SET PASS	6.48
SET PDF	6.49
SET PRIORITY	6.49
SET REPLYTO	6.50
SET REPLYTOLIST	6.50
SET SEPARATOR	6.50
SET SMTP	6.51
SET SNAPDUMP	6.51
SET SOSI	6.51
SET SUBJECT	6.52
SET TLS	6.52
SET TO, SET BLIND, SET COPY	6.53
SET TOLIST	6.53
SET TRANSLATE	6.54
SET TRATT	6.54
SET TRBODY	6.55
SET TRUNC	6.56
SET USER	6.56
SET USERID, SET PASSWORD	6.56
SET VALIDATE	6.57
SET VCARD	6.57
E-mail Examples	6.59
Example 1	6.59
Example 2	6.60

Chapter 7: TCP/IP FOR VSE REXEC Client	7.1
Overview	7.1
Setting Up the REXEC Client	7.2
Running the REXEC Client	7.3
Using CICS REXEC	7.3
Using Batch REXEC	7.4
Parameters	7.4
Commands	7.4
Return Codes	7.5
Sending REXEC Commands	7.6
Using REXEC Client Commands	7.7
SET HOST, SET IPADDR, SET IP	7.8
SET TIMEOUT	7.9
SET TRANSLATE	7.10
SET USERID, SET PASSWORD	7.11
COMMAND	7.12
EXEC	7.13
HELP	7.14
WAIT	7.14
REXEC Examples	7.15
Example 1	7.15
Example 2	7.15
Chapter 8: PDF Conversion Facility	8.1
Overview	8.1
Enabling Conversion	8.3
FTP, EMAIL, LPR	8.3
Carriage Control Formatting	8.4
HTTP Daemon	8.4
Creating the Configuration File	8.7
Calculating the Page Layout	8.12
Example	8.12
Calculating Height	8.12
Calculating Width	8.12
Conversion Examples	8.13
Example 1	8.13
Example 2	8.14
Example 3	8.14

Getting Started

Introduction

This manual explains how to use *TCP/IP FOR VSE* and your VSE mainframe to

- Transfer files
- Log on to VSE VTAM applications from your PC
- Log on to UNIX systems from a VSE CICS session
- Print workstation output on your host printers
- Print z/VSE output on your workstation printer.

The following facilities are provided by all TCP/IP implementations, including *TCP/IP FOR VSE*:

- File Transfer Protocol (FTP), which allows files to be freely exchanged between VSE and other hosts.
- Telnet, which is a protocol that allows you to log on to a remote system using the display facilities of the local system.
- Line Printer Requester (LPR) and Line Printer Daemon (LPD). Together, these facilities allow you to print files from one system on any TCP/IP-connected host.

In general, you may have to configure TCP/IP and its applications on each platform that requires its services. For assistance, see the documentation for the specific platform. The remainder of this manual discusses the applications that are provided with *TCP/IP FOR VSE* and how to use them.

Protocol Background

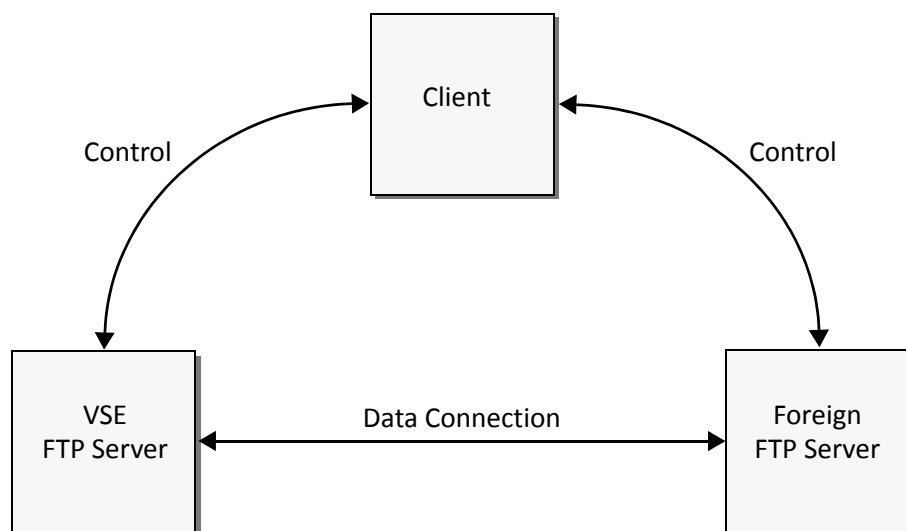
FTP, or File Transfer Protocol, is a widely accepted standard on almost all platforms that have implemented TCP/IP. It is used to transfer files between two systems. RFC959 is the definitive IETF (Internet Engineering Task Force) document that defines the FTP protocol, and it also contains an excellent overview, history, and detailed explanation of the FTP protocol. RFC959 can be viewed or downloaded from www.ietf.org/rfc.html.

The objectives of FTP, according to RFC959, are as follows:

- Promote sharing of files
- Encourage use of remote computers
- Shield users from variations in file storage systems among hosts
- Transfer data reliably and efficiently

Although users can run FTP directly at a terminal, FTP is designed mainly for use by programs. The FTP protocol is really just a set of rules for transferring files between two systems.

The following diagram shows how FTP works.



The client in this diagram controls the transfer of files between the local and foreign systems. The client is also referred to as the control connection. Each system must provide a server that is also known as an FTP daemon. In a nutshell, a single point of control is used to send commands to each FTP server for transferring files between the two systems.

Almost all systems (such as MS Windows, UNIX, VM, Linux, and MVS) come with FTP client software. But, not all systems have an FTP server installed and/or enabled. The local FTP server and the foreign FTP server handle commands from the client, and they usually require a user ID and password before participating in an FTP transfer.

The first thing that must be decided is, “Who is the client for your FTP operation?” Or, phrased another way, “Where do you want to start and control the file transfer?” If you want to start and control the FTP transfer from VSE, then you can use any of the FTP clients provided with TCP/IP. These clients all run on VSE. See the section “[FTP as a Client on VSE](#)” (page 2-17) for detailed information on starting and controlling FTP transfers from VSE.

If you want to start and control the FTP transfer from a foreign system, then you must consult the foreign system’s documentation for information on the commands and the interface that it provides. A common example of a foreign client would be the Microsoft® Windows DOS FTP command. Many PC software companies provide easy-to-use graphical FTP client interfaces.

Before any foreign FTP client can transfer files to or from VSE, an FTP server (daemon) must be running on the VSE system. See the section “[FTP as a Server on VSE](#)” (page 2-105) for information on using VSE as an FTP server. For information on setting up an FTP server on VSE, see the *TCP/IP FOR VSE Installation Guide*, chapter 6, “Configuring FTP Clients and Daemons.”

File Transfer Concepts

All FTP transfers have a single client that is also referred to as the control connection. This FTP client uses the telnet protocol to send commands to and receive replies from a local and foreign FTP server. The FTP client opens connections to

- Local FTP server (daemon), which usually runs on the same system as the client
- Foreign FTP server (daemon).

Both the local and foreign servers usually require a user ID and password.

FTP clients often

- Mask actual FTP server commands (DIR=LIST)
- Issue commands to each server at the same time.

For example, if a batch job on VSE sends a file to a foreign UNIX system, a PUT command in that job would issue these commands:

- RETR command to the VSE FTP server; this command reads the file from VSE
- STOR command to the UNIX FTP server; this command writes the file on UNIX.

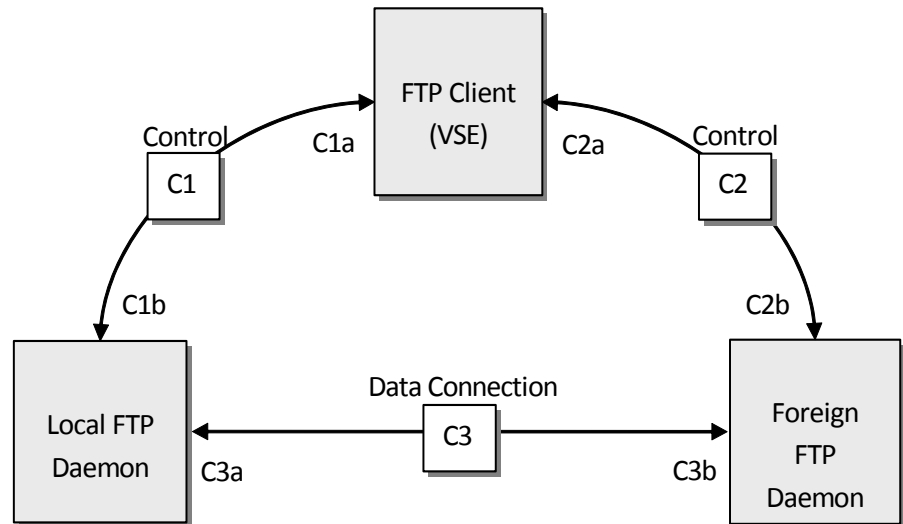
Other FTP server commands would also be issued to control the transfer translation parameters.

Both the foreign and local servers must support the standard set of commands as defined in RFC959. The FTP server commands always consist of four or fewer alpha characters, and the replies are always a three-digit number. The VSE FTP server commands and replies are described in the section "[FTP as a Server on VSE](#)" (page [2-105](#)).

The first issue that must be addressed when using FTP on VSE is where the FTP client should run. You must decide which system is to control the FTP transfer. If the client/control commands will be issued from VSE, then see the section "[FTP as a Client on VSE](#)" (page [2-17](#)). If the client/control commands will be issued from a foreign FTP client to VSE, see the section "[FTP as a Server on VSE](#)" (page [2-105](#)).

Ports

Multiple ports are used during an FTP transfer. It is a good idea to understand what each port is used for and the purpose of the connections. The following diagram identifies the connections



Up to three connections are established during an FTP session. These are shown as C1, C2, and C3 in the above diagram. Each connection requires a pair of ports—one on VSE and one on the foreign system. The following table describes the port connections.

Connection	Description
C1a – C1b	Used for commands from VSE to the locally attached FTP daemon (on VSE). <ul style="list-style-type: none"> • C1a: dynamic next free on VSE (not changeable) • C1b: dynamic next free on VSE (but can be overridden on the LOPEN command)
C2a – C2b	Used for commands from VSE to the foreign FTP daemon. <ul style="list-style-type: none"> • C2a: dynamic next free on VSE (not changeable) • C2b: 21 by default, but can be overridden on the OPEN command
C3a – C3b	Data connection that depends on the active/passive setting. <ul style="list-style-type: none"> • C3a: dynamic or C1b–1 • C3b: dynamic or C2b–1 (or 20, when using the FTP default port 21)

UNIX Emulation Mode

Before you use FTP, you need to understand the concepts involved in transferring files to and from VSE. You also need to understand how these concepts differ from the concepts involved in transferring files between other types of systems.

Most FTP clients assume that the computers they link to use operating systems like MS Windows or UNIX. They expect these machines to have hierarchical file systems, UNIX-style security, and non-record-oriented files. As a result, these clients cannot link to VSE machines directly. You can still use FTP with most of these clients, however, by operating the FTP daemon provided by TCP/IP FOR VSE in UNIX emulation mode. In this mode, the FTP daemon's behavior changes:

- The transfer type specification—ASCII (text) or BINARY—is ignored. Instead, the decision of whether to transfer a file in ASCII mode or binary mode is determined by the file name extension and specifications in the EXTTYPES.L member. The TCP/IP FOR VSE administrator manages this member. (Information on EXTTYPES.L is in the *TCP/IP FOR VSE Installation Guide*.)

Note: True binary transfers can still be enabled by using the following configuration option:

- For an internal FTP server, use the DEFINE FTPD command with UNIX=BIN
- For FTPBATCH, use PARM='UNIX=BIN'

Be aware that even if the UNIX parameter is set to *BIN*, binary transfers may be suppressed if EXTTYPES.L contains a matching file type, in which case the EXTTYPES.L transfer setting is used. To disable using the EXTTYPES transfer settings, use the following option:

- For an internal FTP server, use the DEFINE FTPD command with EXTTYPES=OFF
- For FTPBATCH, use the command SET EXTTYPES OFF

See the section “[Data Translation](#)” (page 2-7) for more information on this topic. For more information on DEFINE FTPD, see the *TCP/IP FOR VSE Installation Guide*.

- The output from the FTP daemon is changed. Directory listings appear as if they had come from a UNIX system. This is desirable because many FTP clients expect this output to be in a rigorously defined format. This is especially true for graphical-mode FTP clients such as WS_FTP (<http://www.ipswitch.com>) and the FTP client that comes with Web browsers like Microsoft's Internet Explorer. TCP/IP FOR VSE works well with these graphical clients when the FTP daemon on VSE runs in UNIX emulation mode.

Entering UNIX Mode

Your system administrator can decide whether an FTP daemon should run initially in UNIX emulation mode. You can change this for each session, and you can even switch back and forth in the middle of a session.

You can enter UNIX emulation mode in two ways:

- Method 1: Use the following FTP CD command:

```
CD /
```

Note the direction of the slash. One characteristic that distinguishes the UNIX file system from the PC file system is the direction of the slash when you display the hierarchy. TCP/IP FOR VSE assumes that if you send commands in UNIX mode (that is, using a forward slash), then you want TCP/IP FOR VSE to respond in UNIX mode.

Entering the CD command with a backslash always terminates UNIX emulation mode. After you issue this command, all interactions and responses are PC compatible.

```
CD \
```

- Method 2: Use the following FTP SITE command:

```
SITE UNIX ON
```

You can terminate UNIX emulation mode at any time by using the OFF option:

```
SITE UNIX OFF
```

Some FTP clients do not support the SITE command. If your FTP client returns an error message stating that the command is not found, use one of the following commands to either enable or disable emulation mode:

```
QUOTE SITE UNIX ON  
QUOTE SITE UNIX OFF
```

To determine whether you are running in UNIX emulation mode, issue the *SYS* or *LSYS* client command, or obtain a directory listing and look at the format.

Data Translation

When you transfer data between different types of systems, compatibility issues arise. To maintain compatibility, some types of systems may require data translation. To determine whether to transfer data using ASCII-to-EBCDIC translation or binary mode (no translation), you need to understand the differences between VSE files and PC or UNIX files:

- VSE text files are in *EBCDIC*. EBCDIC is an 8-bit character set with 256 possible character values.
- PC and UNIX text files are in *ASCII*. ASCII is a 7-bit character set with 128 character values.

This means that 50 percent of the EBCDIC characters have no ASCII equivalents. Since the 7-bit byte was abandoned, PC and UNIX machines now generally support 256 ASCII characters.

VSE files and PC or UNIX files also differ in the following ways:

- VSE binary files are absolute data values. Records within a particular file may have a variable length or a fixed length. Records may be further grouped into blocks.
- PC and UNIX machines have no concept of records, although text files use line control characters to simulate records. An important by-product of this discussion is the fact that when a PC system transfers data to a UNIX system, no translation is required because both systems are ASCII based. In other words, when transferring from PC to UNIX machines, ASCII mode and binary mode mean the same thing.

There are times when you want TCP/IP FOR VSE to perform EBCDIC-to-ASCII translation and times when you don't. In general, the FTP client commands *ASCII* and *BIN* control the translation function:

- The *ASCII* command requests translation.
- The *BIN* command prevents any translation from taking place.

If you are transferring any kind of text file, you want FTP to translate the file if the other side is an ASCII-based system. If you are transferring something that is already in the format required by the other side, as when you are transferring a TCP/IP FOR VSE service pack from a PC to the mainframe, you want to use *BINARY*.

Translation Constraints

We noted that a PC system can transfer a file to a UNIX system without translation. In most cases, the file does not require translation because both systems are ASCII systems. When TCP/IP FOR VSE emulates a UNIX system, the PC system assumes that no matter what the user says, the file must be transferred in binary mode (no translation). Because VSE is really an EBCDIC system, we need some way of overriding this behavior on the part of the PC. This is where the *EXTTYPES.L* configuration file comes in. *EXTTYPES.L* enables you to specify criteria that force ASCII-to-EBCDIC

translation to occur, even when you are running in UNIX emulation mode. It is the only automatic method of causing a non-binary transfer to occur if you are running in UNIX emulation mode.

It is important to understand that the EXTTYPES.L file overrides your selections, regardless of whether you are in UNIX emulation mode, except in the following cases:

- The SITE EXTTYPES OFF command is used to disable EXTTYPES for a session (if allowed by the security exit).
- The SET EXTTYPES OFF command is used in an FTPBATCH job, or this command is added to the FTPBATCH.L member to disable EXTTYPES processing for all FTPBATCH jobs.
- The administrator defined an internal FTP daemon with EXTTYPES=OFF, which requires you to manually enter all of the necessary commands to PUT a file with the desired attributes.

Translation is also controlled by the transfer mode. TCP/IP FOR VSE supports data transfers in three modes:

- ASCII. This mode is used when the target system is an ASCII computer and expects the data to be transmitted in ASCII.
- EBCDIC. This mode is similar to the ASCII mode and is used when the target system is an EBCDIC computer and expects the data to be transmitted in the EBCDIC character set.
- BINARY (or IMAGE). This mode is used when you do not want translation to occur in any way on the data stream.

In general, you do not need to know whether the target computer is an EBCDIC or ASCII computer. TCP/IP FOR VSE transmits the data appropriately, converting it when necessary. For example, you can transmit data to and from VM without specifying EBCDIC. TCP/IP FOR VSE translates the data from EBCDIC to ASCII, and TCP/IP for VM translates the data from ASCII back to EBCDIC.

VSE Directory Output

Directory output varies depending on the type of directory you display. You can obtain directory listings for the following types of VSE directories:

- The root directory
- Library files
- VSE sublibraries
- POWER files
- VSAM catalogs
- User-defined file systems
- HFS (Hierarchical File System).

The following sections describe these directory types and show the fields that are displayed for each type. Remember that the directory displays in this chapter are valid only if you are running in VSE mode (DOS mode). They are not valid if you are running in UNIX compatibility mode.

Root Directory

When the TCP/IP FOR VSE root directory is displayed in non-UNIX mode, the contents are listed as follows:

POWER	<Power Queues>
IJSYSRS	<Library>
PRD1	<Library>
BIMEDLIB	<BIM-Edit>
SAMFILES	<Directory>
VSAMCAT	<VSAM Catalog>
VSAMUCAT	<VSAM Catalog>
HFTST	<HFS>
HFS001	<HFS>
ICCF	<Directory>
ANALYZE	Sequential Access
TSTKSDS	Key Seq VSAM
DFHTEMP	Entry Seq VSAM

The first field contains the public name the system administrator assigned to the file. The second field contains the file type, which may be one of the following values:

File Type	Description
BIM-Edit	Indicates the file is defined with TYPE=BIM-EDIT and therefore represents a BIM-EDIT library. See the BIM-EDIT documentation for more information about this file type.
Data Space	Indicates the file is defined with TYPE=DSPACE. Files of this type are stored directly into virtual memory.
Directory	Indicates a public name with multiple levels in the file system hierarchy below it. To see what file types the individual files represent, you must issue a CD command and then display the directory.
Entry Seq VSAM	Indicates the file is defined with TYPE=ESDS and therefore represents a VSAM ESDS file.
HFS	Indicates the file is defined with TYPE=HFS for a PC-like hierarchical file system that supports long file names

File Type	Description
ICCF Library	Indicates the file is defined with TYPE=ICCF and thus represents the ICCF library structure. Note that you cannot display a directory listing for an ICCF library.
Key Seq VSAM	Indicates the file is defined with TYPE= KSDS and therefore represents a VSAM KSDS file.
Library	Indicates the file is defined with TYPE=LIBRARY and therefore represents a VSE library. If you display a directory using this public name, the output contains the list of sublibraries that are defined in the library.
Power Queues	Indicates the file is defined with TYPE=POWER and therefore represents the POWER queue structure. If you display a directory using this public name, the output shows the POWER queue structure (usually LST, PUN, and RDR).
Sequential Access	Indicates the file is defined with TYPE=SAM and therefore represents a true sequential disk (SD) file or a VSAM-managed SAM file.
VSAM Catalog	Indicates the file is defined with TYPE=VSAMCAT and therefore represents a VSAM catalog. If you display a directory using this public name, the output shows all the entries in the catalog.
VSE VTOC	Indicates the file is defined with TYPE=VTOC and therefore represents a VSE volume. If you display a directory using this public name, the output contains the list of datasets that are defined on the volume.

Library Files When you display a library file directory, you receive output containing a list of sublibraries. The following output is typical:

1	2	3	4	5
CONFIG	<Sub Library>	38	54	11/01/24 10:54
SAVE	<Sub Library>	32	46	11/01/24 10:54
PROD	<Sub Library>	6259	85,306	11/04/01 18:48
SCEEBASE	<Sub Library>	5855	49,460	11/01/24 10:54
CICSR	<Sub Library>	0	1	11/01/24 10:55
DBASE	<Sub Library>	86	6,526	11/01/24 10:55
COMM	<Sub Library>	0	1	11/01/24 10:55
COMM2	<Sub Library>	0	1	11/01/24 10:55
AFP	<Sub Library>	0	1	11/01/24 10:55
GEN1	<Sub Library>	735	29,452	11/01/24 17:40
TCPIP CFG	<Sub Library>	12	17	11/01/24 18:41
HTML	<Sub Library>	102	3,497	11/01/24 18:51
PTCPIP	<Sub Library>	71	2,502	11/03/08 15:55
TCPIP	<Sub Library>	400	5,850	11/03/08 16:36
LSBCFG	<Sub Library>	8	21	11/03/27 00:27
ASN510	<Sub Library>	24	895	11/04/16 18:20
DRSTEST	<Sub Library>	1	7	11/05/05 10:30
LSB130F	<Sub Library>	245	4,747	11/05/18 12:40

The fields are as follows:

Field	Description
1	VSE sublibrary name
2	File type. If you are displaying a directory for a library, the file type is always <i>Sub Library</i> .
3	Members in the sublibrary
4	Used blocks in the sublibrary
5	Date and time the sublibrary was created

VSE Sublibraries When you display a VSE sublibrary directory, you receive output containing a list of members. The following output is typical.

1	2	3	4	5	6	7	8,9
DFHFCTC2.PHASE	2088	3	11/02/23	15:16	11/02/23	15:36	SN
DFHRDTSP.PHASE	396	1	11/01/24	12:04			SN
DFHTCTSP.PHASE	5544	6	11/01/24	11:35	11/01/24	12:04	SN
DTRTERM1.PHASE	140	1	11/01/24	12:04			SN
DTRIIST.PROC	80	1	11/04/16	17:20	11/04/16	17:51	FN
DTRINNW.PROC	80	1	11/04/16	17:18	11/04/16	18:32	FN
DLM1.PROFDIT1	160	1	10/11/05	16:19			FN

The fields are as follows:

Field	Description
1	Member name
2	Records in the file
3	Library blocks in the file
4	Date the file was created
5	Time the file was created
6	Date of the last update
7	Time of the last update
8	File type: <i>F</i> (fixed) or <i>S</i> (string)
9	DATA=YES indicator: <i>N</i> or <i>Y</i> . A <i>Y</i> indicates that this member is a “.PROC.” It was either cataloged with LIBR with “DATA=YES” or added using FTP with the “SITE PROCDATA ON” command.

POWER Files When you display a directory for a POWER file, you receive output containing a list of three entries. The three entries are RDR, LST, and PUN.

When you display a directory for RDR, LST, or PUN, you receive output containing a list of 38 entries. The 38 entries include one for each possible class (A through Z and 0 through 9) and the two special classes (ALL and BIN). The special class ALL includes all classes in the specified queue. The special class BIN includes all classes in the specified queue, but it specifies an automatic transfer type of BINARY on uploads and downloads.

When you display a directory for any class, the displayed information appears as follows:

1	2	3	4	5	6	7
LPRTST.08730.000	59	4	54	3	K	07/08/10 17:48
JRR20526.05663.000	168	6	168	3	D	05/26/11 18:41
JRR20603.36459.000	168	6	168	3	D	06/03/11 15:37
JRR20603.36597.000	168	6	168	3	D	06/03/11 15:40
JRR20603.37579.000	168	6	168	3	D	06/03/11 16:07
LSVCAT8.62313.000	0	0	0	3	D	06/06/11 16:45
LSVCAT.62332.000	143	8	143	3	D	06/06/11 16:46
LSVCAT8.62387.000	143	8	143	3	D	06/06/11 16:47
LSVCAT.41282.001	5	1	3	3	K	06/07/11 15:54
LSVCAT.41282.002	7	1	6	3	K	06/07/11 15:54
LSVCAT.41282.003	44	1	43	3	K	06/07/11 15:54
LSVCAT.41282.004	15	1	14	3	K	06/07/11 15:54
LSVCAT.41282.005	4	1	3	3	K	06/07/11 15:54
LSVCAT.41282.006	2	1	1	3	K	06/07/11 15:54
LSVCAT.41282.007	56	1	55	3	K	06/07/11 15:54
LSVCAT.41282.008	5	1	4	3	K	06/07/11 15:54

The fields are as follows:

Field	Description
1	Job name, job number, and job suffix (segment number) in one string. These subfields are separated by periods.
2	Records in the file
3	Pages in the file
4	Lines in the file
5	Priority of the queue entry
6	Job disposition
7	Date and time the file was created in the queue

VSAM Catalogs When you display a directory for a VSAM catalog, you receive output containing a list of files that are defined in the catalog. The following example is typical:

1	2	3	4	5
VSE.RECORDER.FILE	10/11/24	12:00	1024	S
VSE.HARDCOPY.FILE	10/11/24	12:00	1024	S
VSE.POWER.DATA.FILE	10/02/19	12:00	1024	D
VSE.POWER.ACCOUNT.FILE	10/01/18	12:00	1024	D
VSESP.JOB.MANAGER.FILE	10/02/19	12:00	1024	S
Z9999994.VSAMDSPC.TAF9D957.TC570788	10/11/24	12:00	1024	V
VSE.DUMP.LIBRARY	10/11/24	12:00	1024	S
ICCF.LIBRARY	10/11/24	12:00	1024	D
INFO.ANALYSIS.DUMP.MGNT.FILE	10/11/16	12:00	1024	S
INFO.ANALYSIS.EXT.RTNS.FILE	10/11/24	12:00	1024	S
WORK.HIST.FILE	10/02/19	12:00	1024	S
SYS.NEW.RES	10/02/19	12:00	1024	S

The fields are as follows:

Field	Description
1	File name
2	Date the file was created
3	Time the file was created
4	Logical record length
5	File type: S (Sequential) D (BDAM) V (VSAM) I (ISAM) U (Undefined)

HFS Files When you display an HFS file directory, you receive output containing a list of directories and files that looks very similar to the hierarchical file system on a PC. The following output is typical.

1	2	3	4	5	6	7
17/01/2010	12:11:11	<DIR>				CSIVSEDR
17/01/2010	12:12:12	<DIR>				CSIVSEDW
23/02/2010	20:11:11		333,557	S	4096	B COWBOYCLOSEUP.JPG
23/02/2010	20:11:11		32,323	S	4096	B PRODUCTCOLLAGE2.GIF
23/02/2010	20:13:13		9,276,416	S	2048	B PRODUCTCOLLAGE2.PPF
23/02/2010	20:14:14		203	V	1024	E INDEX.HTML
23/02/2010	20:14:14		176,080	F	80	E FTPBSAM1.TXT
08/02/2010	22:55:55		2,966	V	256	E SEEVSE README.HTM
12/02/2010	23:03:03		2,146,320	F	80	B TCPIP15E.BJB
13/02/2010	18:58:58		2,293,760	S	4096	B TCPIP15E.ZIP
15/02/2010	14:16:16		11,208	V	1024	E CSI-INTERNATIONAL.HTM

The fields are as follows:

Field	Description
1	Date the file was created
2	Time the file was created
3	Bytes in the file
4	Record format: S (string), V (variable), F (fixed)
5	Record length in bytes
6	Data type: B (binary), E (EBCDIC), A (ASCII)
7	File or directory name

Mechanics of Transferring a File

TCP/IP FOR VSE transfers a file to or from VSE in the following manner:

- When you transfer a file from VSE to another system, you read the file from VSE, apply an algorithm to it, and send the file to another platform. The file is stored by that platform's FTP. TCP/IP FOR VSE uses the following algorithm to process the transfer:
 1. TCP/IP FOR VSE reads the data from the DASD owned by VSE.
 2. If TCP/IP FOR VSE is in ASCII mode, it either drops or uses carriage control (CC) data:
 - If CC is OFF, it removes any CC information.
 - If TRCC is ON, it interprets the CC character. The CC character is removed and replaced with the appropriate ASCII characters.
 3. TCP/IP FOR VSE translates the data from EBCDIC to ASCII using the appropriate translate table.

4. If CRLF is on, TCP/IP FOR VSE adds an ASCII carriage return/line feed combination to the end of each record.
 5. TCP/IP FOR VSE sends the data through the network.
- When you transfer a file to VSE from another system, TCP/IP FOR VSE reads the file from the network, applies an algorithm to it, and stores the file on a DASD that is owned by VSE. In general, TCP/IP FOR VSE uses the following algorithm:
 1. TCP/IP FOR VSE reads data from the network.
 2. If TCP/IP FOR VSE is in ASCII mode:
 - It scans for an end-of-record indicator, which is usually a carriage return followed by a linefeed, or a linefeed character by itself (if RECLF is set to on). It removes the character and inserts the EBCDIC equivalent.
 - It translates the data to EBCDIC using the appropriate translate table.
 - If CC is ON, it prefixes the record with a blank.
 3. If the record format is fixed (RECFM=F) and the record itself is too short, TCP/IP FOR VSE fills the record with the pad character (usually blanks).
 4. TCP/IP FOR VSE writes the record.

This explanation is simplistic, but it describes the basic process. Although the results may be slightly different, the algorithm is the same regardless of the direction the data is moving.

Diagnosing FTP Hangs

Occasionally an FTP session hangs, which means that it appears to be making no progress across the network. There are many reasons why this could happen, but one common reason is an inappropriate MTU size on the outgoing connection. TCP/IP FOR VSE's DISCOVER client can help you determine the best MTU size for a particular TCP/IP host. While this utility can determine the optimal setting for one host, it may return a completely different setting for another host using the same link. If you are experiencing FTP hangs, you should always use the lowest MTU value returned by the utility for any host.

For more information on the DISCOVER client, see [Chapter 5, "Ping, Traceroute, DISCOVER Clients."](#)

FTP as a Client on VSE

The following FTP client types are available on VSE:

1. Internal FTP batch client. This client runs in a VSE partition with the // EXEC FTP JCL statement.
2. External FTP batch client. This client runs in a VSE partition with the // EXEC FTPBATCH JCL statement.
3. Automatic FTP client. This client
 - automatically transfers POWER queue members to a foreign FTP server
 - is created with the TCP/IP FOR VSE DEFINE EVENT command
 - uses FTP commands from a user-defined script file that resides in a VSE *library.sublibrary*.
4. Interactive FTP client. This client
 - uses a CICS FTP transaction
 - requires definitions in CICS for the transaction ID and the associated program.
5. Programmable FTP client. A program can be written on VSE in any language (such as REXX, COBOL, C, or Assembler) to create a custom FTP client. For more information, see the *TCP/IP FOR VSE Programmer's Guide*.

You can initiate a file transfer using any of these methods from your VSE system. The basic idea behind each client is that you initiate the file transfer by providing a series of commands. You can provide the commands using the mechanism that is appropriate to the method that you used to invoke the FTP client. You can enter commands in a batch job, automatically in a script, interactively, or programmatically. The FTP client returns responses to you, your program, or your batch job using SYSLST. The commands you can issue differ slightly depending on the client, but they are all similar.

Internal FTP Client

This client is called the *internal* FTP client because the I/O originates in the TCP/IP FOR VSE partition. Most of the processing takes place in that partition, although the job does not run in the TCP/IP FOR VSE partition. The internal FTP client (along with the external FTPBATCH client) is sometimes called a batch FTP client. Your system administrator can provide guidelines for when it is appropriate to use the internal FTP client and when it is appropriate to use the external FTPBATCH client.

If you specified the internal client and it looks like TCP/IP FOR VSE ran the external client anyway, then your system administrator probably used the FTPBATCH_FETCH command to force all FTP batch jobs to use the FTPBATCH client.

Note: If the FTPBATCH_FETCH ON command has been issued, then the // EXEC FTP program will fetch the FTPBATCH program on top of itself. This effectively makes the // EXEC FTP command the same as // EXEC FTPBATCH, and all the rules for FTPBATCH then apply. The // UPSI XX1XXXXX switch causes // EXEC FTP to ignore the FTPBATCH_FETCH setting.

If you are the system administrator, you can find information about the differences between internal and external clients in the *TCP/IP FOR VSE Installation Guide*. For more information about the FTPBATCH_FETCH command, see the *TCP/IP FOR VSE Command Reference*.

Starting the Client

To start the internal FTP client, use the following JCL as a guide:

```
* $$ JNM=FILETRAN,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=H
// JOB FILETRAN
// LIBDEF *,SEARCH=(PRD2.TCPIP)
// EXEC FTP,PARM='IP=ipaddr,PORT=portnum,ID=sysid'
ftp command 1
ftp command 2
ftp command n
. . .
QUIT
/*
/&
* $$ E0J
```

Parameters

The internal FTP client uses the parameters shown in the following table. Because the parameters are keywords, you can use them in any order.

Parameter	Description
CASE=UPPER	Specifies that all messages issued to the console and to SYSLST are translated to upper case. (<i>UPPER</i> is the only valid value.)
ID= <i>sysid</i>	Specifies the system ID of the TCP/IP FOR VSE partition that is to serve as the local host or client. Remember that you can have more than one copy of TCP/IP FOR VSE running at one time (for example, production and test). The default is <i>00</i> . If your installation uses a TCP/IP FOR VSE ID other than <i>00</i> , you must specify the <i>sysid</i> value in all FTP batch jobs that you run.
IP= <i>ipaddr</i>	Specifies the IP address of the foreign host or server you want to transfer the file to or from. Variable <i>ipaddr</i> can be an IP address such as <i>192.168.0.7</i> or a symbolic name such as <i>tcpip4vse.com</i> . If you use this parameter, the client expects the first four lines coming from SYSIPT to contain, in order, the foreign user ID, the foreign password, the local user ID, and the local password. If you do not use this parameter now, you can use the OPEN command later to establish the session with the foreign host.
PORT= <i>portnum</i>	Specifies the TCP port number on the foreign host or server. The default is <i>21</i> .
RETRY= <i>n</i>	Specifies the number of times the FTP client should attempt to open a connection to the foreign host. The default is <i>1</i> .
RETRYTIME= <i>nnn</i>	Specifies how long an FTP client should wait between attempts to open a connection to the foreign host. The default value is <i>600</i> seconds (10 minutes). Note: The RETRY parameter must be set to a value greater than 0. Otherwise, RETRYTIME has no effect.

UPSI Switches

The internal FTP client allows you to use the VSE User Program Switch Indicators (UPSI) facility to specify the parameters shown in the following table:

UPSI Code	Description
1XXXXXXX	Displays all SYSLOG and SYSLST messages in upper case
X1XXXXXX	Suppresses sending messages to the console (SYSLOG)
XX1XXXXX	Causes // EXEC FTP to ignore the FTPBATCH_FETCH setting

Command Variables

The internal FTP client accepts all SETPARM variables from the VSE JCL. The command variable is replaced with the value the variable had on entry to the FTP job.

Return Codes

The internal FTP batch client returns the following codes:

Return Code	Meaning
0	All commands in your FTP job completed normally.
4	<p>Either the foreign FTP daemon or the local FTP daemon reported an FTP error message in the 4xx range. To find the exact problem, study your output. These errors are generally recoverable in that the FTP process may continue with the next command.</p> <p>You can also receive this return code if there is a file definition error and the file is not predefined to the TCP/IP FOR VSE file system. The syntax for these files is explained in the section "VSE File Names" (page 2-46).</p>
8	Either the foreign FTP daemon or the local FTP daemon reported an FTP error message in the 5xx range. To find the exact problem, study your output. These errors are not recoverable, and the TCP/IP FOR VSE FTP client automatically sends the QUIT command to the FTP daemon and terminates the connection.

Return Code	Meaning
16	<p data-bbox="740 243 1187 275">One of the following events occurred:</p> <ul data-bbox="740 296 1425 663" style="list-style-type: none"><li data-bbox="740 296 1425 432">• The TCP/IP FOR VSE FTP client encountered an invalid parameter while processing the PARM= keyword on the // EXEC FTP card. To find the invalid parameter, study your output. Correct the parameter and rerun the job.<li data-bbox="740 443 1425 516">• The TCP/IP FOR VSE FTP client abended. Contact CSI Technical Support for assistance.<li data-bbox="740 527 1425 663">• The TCP/IP FOR VSE FTP client cannot establish a connection with the TCP/IP FOR VSE partition. Make sure that the TCP/IP partition with the identifier specified in the ID= parameter is active.
24	<p data-bbox="740 684 1425 821">The TCP/IP FOR VSE FTP client cannot load its message handler. Check your LIBDEF statements to ensure that you specified <i>LIBDEF *</i> for the TCP/IP FOR VSE installation library (as opposed to <i>LIBDEF PHASE</i>).</p>

External FTPBATCH Client

This client is called the *external* FTPBATCH client because most of the processing for the job, including CPU and I/O processing, takes place outside of the TCP/IP FOR VSE partition. The external and internal batch FTP clients are sometimes referred to as the VSE batch FTP clients.

Your system administrator can provide guidelines for determining when it is appropriate to use an external FTPBATCH client and when it is appropriate to use the internal FTP client. If you are the system administrator, see the *TCP/IP FOR VSE Installation Guide* for more information.

Starting the Client

To start the external FTPBATCH client, use the following JCL as a guide:

```
* $$ JNM=FILETRAN,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=H
// JOB FILETRAN
// LIBDEF *,SEARCH=(PRD2.TCPIP)
// EXEC FTPBATCH,SIZE=FTPBATCH,PARM='ID=sysid'
SET command 1
SET command 2
LOPEN
LUSER local-vse-userid
LPASS local-vse-password
OPEN foreign-ip-addr foreign-port-number
USER foreign-user-id
PASS foreign-password
ftp command 1
ftp command 2
CLOSE
LCLOSE
QUIT
/*
/&
* $$ EOJ
```

Except for POWER JECL or SLI statements (beginning with “* \$\$”), an asterisk in column 1 begins a comment. A command may be continued on the next line by appending a space and a hyphen (-) to the initial line.

Note: Do not end a comment line with a hyphen; otherwise, processing of the next command may fail.

Parameters

The following keyword parameters may be used. Some parameters apply to an FTPBATCH server.

Parameter	Description
ABORT=[<u>YES</u> NO]	Specifies whether the ABORT command is allowed to prematurely terminate an active transfer. The default is <i>NO</i> .
CASE=UPPER	Specifies that all messages issued to CONSOLE and SYSLST are translated to upper case. (<i>UPPER</i> is the only valid value.)
COUNT	Synonym for MAXACT
DATAPORT= <i>num</i>	Data port number to be used for the local VSE data connection. Applies only when VSE is in the passive mode.
DEBUG=[ON <u>OFF</u>]	Sends debugging messages to SYSLST. The default is <i>OFF</i> .
DUMP=[YES <u>NO</u>]	Specify <i>YES</i> to create a complete dump if the FTPBATCH program abends. Use this parameter when directed by CSI technical support. The default is <i>NO</i> .
DYNFILE=[<u>YES</u> NO]	Allows the user to transfer autonomous files. Default is <i>YES</i> . See the section “ Autonomous Files ” (page 2-47) for more information.
FTIMEOUT=[<i>nnnnn</i> <u>36000</u>]	How long FTPBATCH should wait for the OPEN to a foreign FTP server to complete. The value is the number of 1/300 th -second intervals. Note: The DEFINE ROUTE command can shorten this time interval. The default is <i>36000</i> (2 minutes).
FTPD=NO	Suppresses loading and attaching the FTP daemon when using LOPEN with a foreign FTP server. (<i>NO</i> is the only valid value.)
FTPDPORT= <i>nnnn</i>	Can be used to create an external FTPBATCH server. This causes the FTPBATCH partition to stay up and wait for connection requests from foreign FTP clients. Note: You can block autonomous file transfers on this server (port) using the DYNFILE=OFF parameter.

Parameter	Description
ID= <i>sysid</i>	<p>Specifies the system ID of the TCP/IP FOR VSE partition that is to serve as the local host or client. Remember that you can have more than one copy of TCP/IP FOR VSE running at one time (such as production and test). The default is <i>00</i>. If your installation uses a TCP/IP FOR VSE ID other than <i>00</i>, you must specify the <i>sysid</i> value in any FTP batch job that you run. Alternatively, you can specify the ID parameter in a</p> <pre>// OPTION SYSPARM='sysid' job control statement.</pre>
IP= <i>ipaddr</i>	<p>Specifies the IP address of the foreign host or server that you want to transfer the file to or from. The variable <i>ipaddr</i> can be an IP address such as <i>192.168.0.7</i> or a symbolic name such as <i>tcip4vse.com</i>. If you use this parameter, the client expects the first four lines coming from SYSIPT to contain, in order, the foreign user ID, the foreign password, the local user ID, and the local password. If you do not use this parameter now, you can use the OPEN command later to establish the session with the foreign host.</p> <p>Note: It is best to specify the foreign IP address and port number on the OPEN command, as shown in the example. IP= can be used to specify the foreign IP address, but it is provided primarily for compatibility with the internal FTP client.</p>
LIP= <i>nnn.nnn.nnn.nnn</i>	<p>Allows a different local IP address to be used by FTPBATCH for use in VSE systems with multiple local IP addresses.</p>
MAXACT= <i>nn</i>	<p>Specifies the maximum number of FTP sessions that can connect to FTPBATCH in server mode. This number ranges from 1 to 28. The default is 16, which normally is adequate to ensure that sessions can connect. While you can permit up to 28 concurrent sessions, each session, as it becomes active, requires additional system resources. You must plan the size of the FTPBATCH server partition accordingly.</p>
PORT=[<i>portnum</i> 21]	<p>Specifies the TCP port number on the foreign host or server. The default is <i>21</i>.</p> <p>Note: It is best to specify the foreign port on the OPEN command. See the note for IP= above.</p>

Parameter	Description
RETRY=[<i>n</i> 1]	Specifies the number of times the FTP client should attempt to open a connection to the foreign host. The default is 1.
RETRYTIME=[<i>nnn</i> 18000]	Specifies how long an FTP client should wait between attempts to retry a failed command. This value is specified in 1/300 th -second intervals. The default is 18000 (1 minute). If you do not specify the RETRY parameter, the value specified for RETRYTIME has no effect.
SSL={CLIENT SERVER}	Enables SSL (secure encryption) for either the client or the server. Requires activating the SecureFTP optional feature.
SYSLOG=ON	Specifies that all messages are directed to the console and to SYSLST. (ON is the only valid value.)
TAG=[YES NO]	Specify YES to add the task ID and timestamp to all SYSLST output. The default is NO.
TLS	Synonym for SSL
TRAN= <i>translate-table</i>	<p>Specifies a translate table name for ASCII-to-EBCDIC translation on the data connection. This setting does not override the TRANSLATE parameter on the DEFINE FILE command.</p> <p>Note 1: Setting this parameter causes SITE TRANSLATE commands to fail with a “505 Translate value cannot be overridden” error message.</p> <p>Note 2: Use SET TELNTRAN to override the translate table setting on the control connection.</p>

Parameter	Description
UNIX=[YES BIN NO]	<p>Controls operation in UNIX simulation mode. Specify <i>YES</i> to tell the FTP daemon to start each session in UNIX simulation mode. Binary transfers occur in ASCII mode. The default is <i>NO</i>.</p> <p><i>BIN</i> is the same as <i>YES</i>, but it is better because it allows binary transfers to occur in binary mode. This is useful when you want to use the simulated UNIX directory structure of TCP/IP FOR VSE while allowing true image/binary transfers to occur.</p> <p>Forward slashes may be used to separate directory and file names in UNIX mode. For more information on this mode, see the <i>TCP/IP FOR VSE Installation Guide</i>, chapter 1, "Fundamentals of TCP/IP."</p>
WELCOME= <i>member</i>	Name of a VSE library ('.L') member containing site-specific text that is added to the initial 220 message issued by an FTPBATCH server.
ZEROERR=[YES NO]	Specifies whether a GET of an empty file should be considered an error (<i>YES</i>) or ignored (<i>NO</i>). The default is <i>NO</i> .

UPS I Switches

The VSE User Program Switch Indicators (UPS I) facility may be used to specify the parameters shown in the following table. These UPS I switches are supported in FTPBATCH for compatibility with older releases. Where possible, use the equivalent SET commands instead.

UPS I Code	Description	SET Command
1XXXXXXX	Displays all SYSLOG and SYSLST messages in upper case	SET CONSOLE UPPER
X1XXXXXX	Suppresses sending messages to SYSLOG the console (SYSLOG)	SET CONSOLE NONE
XX1XXXXX	Suppresses the version check	(Not available)
XXX1XXXX	Requires shutdown command to terminate	SET SHUTDOWN ON
XXXX1XXX	Activates diagnostic events	SET DIAGNOSE EVENTS

Controlling the External FTPBATCH Client

The external FTPBATCH client provides additional commands that can be used to control its processing and performance.

You can include these commands directly in your external FTP batch job, or you can code them in an FTPBATCH.L library member. The external FTP batch job executes the commands in this member during its initialization before executing any other FTP commands.

SET Commands

The SET commands specify operational parameters. Use the default values for most installations and change them only to handle unique situations.

Note: In batch jobs, all SET commands must be issued before the LOPEN command. (All commands follow the // EXEC FTPBATCH statement.)

SET commands are issued as shown in this example:

```
SET DATAWECB ON
```

The SET commands are described in the following table:

SET Command	Description
SET BUFFCNT	Synonym for SET BUFFMAX
SET BUFFMAX [<i>nn</i> <u>4</u>]	This command is no longer used. If it is specified, it is ignored.
SET BUFFSIZE [<i>nnnnn</i> <u>65536</u>]	Sets the buffer size to use when sending data to the TCP/IP partition. Using a smaller BUFFSIZE than the default causes more sends of smaller buffers to the TCP/IP partition. The default is 65536. The maximum is 131072.
SET CHAPCNT [<i>nn</i> <u>32</u>]	Used when running FTPBATCH as an external FTP server. Each VSE subtask handles an FTP session, and the VSE CHAP command controls when a session is forced to the lowest priority to allow other sessions to have equal priority. In general, higher values of <i>nn</i> lead to fewer changes in priorities (CHAPs). The default is 32.

SET Command	Description																
SET CONSOLE <i>operand</i>	<p>Manages how messages are displayed on the VSE system console. The <i>operand</i> specifies whether a message type is sent to the console or suppressed. Only one operand from the following list is allowed per command. Each operand's value defaults to the default setting of the TCP/IP partition.</p> <table border="1"> <thead> <tr> <th>Operand</th> <th>Message Type</th> </tr> </thead> <tbody> <tr> <td>ALL NONE</td> <td>All messages</td> </tr> <tr> <td>WARN NOWARN</td> <td>Warning</td> </tr> <tr> <td>INFO NOINFO</td> <td>Informational</td> </tr> <tr> <td>DIAG NODIAG</td> <td>Diagnostic</td> </tr> <tr> <td>RESP NORESP</td> <td>Response</td> </tr> <tr> <td>SECURE NOSECURE</td> <td>Security</td> </tr> <tr> <td>UPPER</td> <td>All messages are sent in upper case</td> </tr> </tbody> </table>	Operand	Message Type	ALL NONE	All messages	WARN NOWARN	Warning	INFO NOINFO	Informational	DIAG NODIAG	Diagnostic	RESP NORESP	Response	SECURE NOSECURE	Security	UPPER	All messages are sent in upper case
Operand	Message Type																
ALL NONE	All messages																
WARN NOWARN	Warning																
INFO NOINFO	Informational																
DIAG NODIAG	Diagnostic																
RESP NORESP	Response																
SECURE NOSECURE	Security																
UPPER	All messages are sent in upper case																
SET DATAWECB [ON OFF]	ON causes a wait-after-send buffer until the buffer is successfully acknowledged by the foreign FTP server. The default is <i>OFF</i> .																
SET DIAGNOSE [ON OFF] EVENTS FILEIO]	<p>Controls FTP diagnostics as follows:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Effect</th> </tr> </thead> <tbody> <tr> <td>ON</td> <td>Enables diagnostics of FTP commands</td> </tr> <tr> <td>OFF</td> <td>Disables diagnostics of FTP commands (the default)</td> </tr> <tr> <td>EVENTS</td> <td>Enables diagnostics of significant events during the FTP processing</td> </tr> <tr> <td>FILEIO</td> <td>Enables dumps during file I/O requests</td> </tr> </tbody> </table>	Value	Effect	ON	Enables diagnostics of FTP commands	OFF	Disables diagnostics of FTP commands (the default)	EVENTS	Enables diagnostics of significant events during the FTP processing	FILEIO	Enables dumps during file I/O requests						
Value	Effect																
ON	Enables diagnostics of FTP commands																
OFF	Disables diagnostics of FTP commands (the default)																
EVENTS	Enables diagnostics of significant events during the FTP processing																
FILEIO	Enables dumps during file I/O requests																

SET Command	Description										
SET EXTRADAT [<u>FAIL</u>] WARN IGNORE ACCEPT]	<p>This option controls how extra data at the end of a received text file is handled. “Extra data” is defined as a character string not delimited by a CR, LF, or other valid delimiter. The settings are as follows:</p> <table border="1" data-bbox="899 432 1421 1152"> <thead> <tr> <th data-bbox="899 432 1036 474">Value</th> <th data-bbox="1036 432 1421 474">Effect</th> </tr> </thead> <tbody> <tr> <td data-bbox="899 474 1036 699">FAIL</td> <td data-bbox="1036 474 1421 699">The transfer fails and is not stored on VSE. A warning message (FTP343W) is generated, and a 5xx failure code is sent to the client (the default).</td> </tr> <tr> <td data-bbox="899 699 1036 884">WARN</td> <td data-bbox="1036 699 1421 884">A warning message is generated, but the file is stored on VSE. A 2xx normal code is sent to the client. The undelimited data is discarded.</td> </tr> <tr> <td data-bbox="899 884 1036 999">IGNORE</td> <td data-bbox="1036 884 1421 999">No messages are generated. The undelimited data is discarded.</td> </tr> <tr> <td data-bbox="899 999 1036 1152">ACCEPT</td> <td data-bbox="1036 999 1421 1152">No messages are generated. The undelimited data is accepted and stored as if it were correctly delimited.</td> </tr> </tbody> </table>	Value	Effect	FAIL	The transfer fails and is not stored on VSE. A warning message (FTP343W) is generated, and a 5xx failure code is sent to the client (the default).	WARN	A warning message is generated, but the file is stored on VSE. A 2xx normal code is sent to the client. The undelimited data is discarded.	IGNORE	No messages are generated. The undelimited data is discarded.	ACCEPT	No messages are generated. The undelimited data is accepted and stored as if it were correctly delimited.
Value	Effect										
FAIL	The transfer fails and is not stored on VSE. A warning message (FTP343W) is generated, and a 5xx failure code is sent to the client (the default).										
WARN	A warning message is generated, but the file is stored on VSE. A 2xx normal code is sent to the client. The undelimited data is discarded.										
IGNORE	No messages are generated. The undelimited data is discarded.										
ACCEPT	No messages are generated. The undelimited data is accepted and stored as if it were correctly delimited.										
SET EXTYPES [<u>ON</u> OFF]	<p>Enables or disables using EXTTYPE.L for file transfer overrides. The default is <i>ON</i> (enabled).</p> <p>See also “SET SITELAST [YES NO]” (page 2-32).</p>										
SET IDLETIME [<i>nnnnnn</i> 0]	<p>Sets the wait time before terminating an inactive FTP session when using FTPBATCH as an external FTP server. The value <i>nm</i> is the number of 1/300th-second intervals. The default is 0.</p>										

SET Command	Description
SET IGNORERR [ON OFF]	<p>An external FTPBATCH client usually terminates when it encounters a serious problem, which is one that generates a 400- or 500-level message. A serious problem may also be indicated by an error message in this format:</p> <p><i>Error: variable text describing the error condition</i></p> <p>This error could be caused by a severed connection with a reset, and TCP/IP FOR VSE cannot recover from it. For example, the following message could be issued when a SEND or a RECEIVE to either the local or foreign FTP server fails:</p> <p><i>Error: Foreign control connection failed S NDFSTAF FFFFFFFF 000001E8</i></p> <p>If you want FTP to continue under these circumstances, specify SET IGNORERR ON. The default is <i>OFF</i>. Some FTP daemons close the connection after they send a 500-level message. In such cases, the external FTPBATCH client tries to continue, but all subsequent commands are rejected.</p>
SET JOURNAL [ON OFF]	Enables or disables journaling for HFS (Hierarchical File System). The default is <i>OFF</i> .
SET LOGCONSL	Synonym for SET CONSOLE
SET [MSGXLEFT MSGXRGHT]	Causes timestamps to be placed to the left or right of messages issued to SYSLST. The default is to the right.

SET Command	Description
SET MSGXLOG [ON OFF]	<p>SET MSGXLOG ON can be used to echo SYSLST output from FTPBATCH into a sequential disk (SD) file. This can be used for applications that want to ASSGN SYSLST to a disk file for separate processing of output from FTP commands such as a directory listing. This, then, causes an open of an SD file for output, and the FTPBATCH job stream must also contain the ASSGN, DLBL, and EXTENT JCL for the MSGXLOG file.</p> <p>The output file is fixed in unblocked, 121-byte records to allow assigning IJSYSLS to a disk file. The first byte of each record contains the print carriage control and is followed by the 120-byte print line. The default is <i>OFF</i>.</p>
SET NEEDFMSG [ON OFF]	<p>Specifies whether you require either a 226 or a 250 reply from the foreign FTP server when a GET or PUT request successfully completes. The default is <i>ON</i>.</p>
SET NULLRECD NOTHING	<p>For ASCII transfers with null records, SET NULLRECD NOTHING suppresses adding a record containing a single blank character when transferring an ASCII file containing the sequence CR-LF-CR-LF with no intervening blanks. (Note: <i>NOTHING</i> is the only valid value.)</p>
SET PASVPORT <i>start-number</i>	<p>Sets the starting port number to be used for the data connection when in passive mode. This number must be greater than 4096 and less than 65536. This setting overrides the values set by the PORTRANGE command in the TCP/IP partition.</p>
SET PULSE [ON OFF]	<p>Specifies whether pulsing is on or off during a GET or a PUT. The default is <i>ON</i>.</p>
SET RETCODE REPLYNUM	<p>For a failing job, this command causes FTPBATCH to use the highest reply number received from either the local or the foreign FTP daemon as the job step return code. It overrides the standard 4, 8, 12, and 16 return codes. The three-digit FTP reply codes are specified by RFC0959. This command may be set in FTPBATCH.L.</p>

SET Command	Description
SET SENDFAST [ON OFF]	Causes SENDs to be issued without waiting. When SENDFAST is set to ON and the number of unacknowledged bytes is four times the BUFFSIZE, a wait is issued to allow the foreign FTP server to acknowledge the sent data. The default is <i>OFF</i> .
SET SENDSNOT	Synonym for SET SENDFAST
SET SENDWACK	Synonym for SET DATAWECB
SET SHUTDOWN [ON OFF]	SET SHUTDOWN ON causes the FTPBATCH job to remain active until a shutdown command is encountered. The default is <i>OFF</i> .
SET SITELAST [YES NO]	Setting this option to <i>YES</i> allows SITE commands to override the following parameters on the DEFINE FILE command: BLKSIZE, CC, CRLF, LRECL, RECFM, TRCC. The default is <i>NO</i> . Note: If SET EXTYPES ON is used and the file type has a matching entry in EXTYPES.L, then this option has no effect and EXTYPES overrides are always used. See the section “ Command Precedence ” (page 2-118) for details.
SET SSL	Enables SSL. Requires activating TCP/IP FOR VSE’s SecureFTP optional feature.
SET STALLMAX [<i>n</i> 3]	Specifies the number of consecutive stall periods that the FTPBATCH client is to wait for before it terminates. The default is 3.
SET STALLTIM [<i>nnnnn</i> 18000]	Sets how long FTPBATCH waits for bytes to be sent or received. If no bytes are detected within this time, then a stall condition exists. The time is specified in 1/300 th -second intervals. The default is 18000 (1 minute). This setting also controls how messages FTP310 and FTP311 are issued. These messages show how many bytes have been sent or received in the current transmission. If the transmission is stalled, then the external FTPBATCH client produces message FTP313W.

SET Command	Description								
SET STAMP [NONE LEFT RIGHT]	<p>Specifies how to place timestamps in messages sent to SYSLST:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Effect</th> </tr> </thead> <tbody> <tr> <td>NONE</td> <td>Turns off timestamps</td> </tr> <tr> <td>LEFT</td> <td>Puts timestamps to the left of messages</td> </tr> <tr> <td>RIGHT</td> <td>Puts timestamps to the right of messages (the default)</td> </tr> </tbody> </table>	Value	Effect	NONE	Turns off timestamps	LEFT	Puts timestamps to the left of messages	RIGHT	Puts timestamps to the right of messages (the default)
Value	Effect								
NONE	Turns off timestamps								
LEFT	Puts timestamps to the left of messages								
RIGHT	Puts timestamps to the right of messages (the default)								
SET TELNTRAN <i>table-name</i>	Sets the translation table name to be used for the control connection. This value overrides the SET TELNET_TRANSLATE setting for the TCP/IP partition. (For more information on SET TELNET_TRANSLATE, see the <i>TCP/IP FOR VSE Command Reference</i> .)								
SET TERSE [ON OFF]	ON causes shortened (one-line) 150 and 226 messages to be used for GET and PUT requests. The default is OFF.								
SET TIMERAMT	Synonym for SET STALLTIM								
SET TLS	Synonym for SET SSL								
SET VARMAX [<i>nnnnn</i> 80]	Sets the maximum number of variables. This must be specified in the FTPBATCH.L book. The default is 80.								

The anticipated response is that the following message is written to the SYSLST associated with the FTPBATCH job.

SETCDONE

Command Variables

The external FTPBATCH client accepts the following command variables:

Command Variable	Description
All VSE SETPARM variables from VSE JCL	The command variable is replaced with the value the variable had on entry to the FTPBATCH job. Note: You must use the SETVAR command to assign a VSE variable to another variable before you can use the VSE variable in a command. See the SETVAR command description (page 2.99) for an example.
&CURDATE	The current date in the form <i>mddycc</i> (or <i>ddmmycc</i> when VSE is configured with <i>STDOPT=DMY</i>) where <i>mm</i> is the month, <i>dd</i> is the day, <i>yy</i> is the year, and <i>cc</i> is the century. You must use the SETVAR client command to access this variable.
&CURTIME	The current time in the form <i>hhmmss</i> , where <i>hh</i> is the hour (using a 24-hour clock), <i>mm</i> is the minute, and <i>ss</i> is the second. You must use the SETVAR client command to access this variable.

Return Codes

The external FTP batch client uses the return codes shown in the following table. If you need to contact CSI Technical Support because of an error in the external FTP batch client, please include the full FTPBATCH SYSLST output.

Note: The SET RETCODE REPLYNUM command causes FTPBATCH to override the return codes in this table and use the highest reply number received from the local or the foreign FTP daemon as the job step return code. The three-digit FTP reply codes are specified by RFC0959.

The external FTP batch client return codes are described in the following table.

Return Code	Meaning
0	All commands in your FTP job completed normally.
4	<p>Either the foreign FTP daemon or the local FTP daemon reported an FTP error message in the 4xx range. To determine the exact problem, examine your output. These types of return codes are generally recoverable in that the FTP process may continue with the next command.</p> <p>You may also receive a return code of 4 if you have an error in a file definition that is not predefined to the TCP/IP FOR VSE file system. The syntax for these types of files is explained in the section "VSE File Names" (page 2-46).</p>
8	<p>Either the foreign FTP daemon or the local FTP daemon reported an FTP error message in the 5xx range. To determine the exact problem, examine your output. These types of return codes are not recoverable, and your FTP job has terminated. The TCP/IP FOR VSE FTP client automatically sends the QUIT command to the FTP daemon and terminates the connection.</p>
12	<p>An error occurred. You must examine the text of the accompanying FTP301E message to determine the nature of the error.</p>
16	<p>One of the following events occurred:</p> <ul style="list-style-type: none"> • The TCP/IP FOR VSE FTP client encountered an invalid parameter while processing the PARM= keyword on the // EXEC FTPBATCH card. To find the invalid parameter, examine your output. Correct the parameter and rerun the job. • The TCP/IP FOR VSE FTPBATCH client abended. In this case, contact CSI Technical Support for assistance. • The TCP/IP FOR VSE FTP client could not establish a connection with the TCP/IP FOR VSE partition. Make sure that the TCP/IP partition with the identifier specified in the ID= parameter is active.

Using REXX to Run the External FTPBATCH Client

You can run the external FTPBATCH client from a REXX program using the standard VSE REXX ADDRESS LINK command. The program in the following box shows how to initiate the external FTPBATCH client from a REXX program.

```
/* REXX */
in.0 = 9
in.1 = 'lopen'
in.2 = 'luser robert'
in.3 = 'lpass robert123'
in.4 = 'open 192.168.0.47'
in.5 = 'user vseuser'
in.6 = 'pass vseuser'
in.7 = 'lcd prd2.robert'
in.8 = 'get profile.exec vmprof.proc'
in.9 = 'quit'
call assgn 'STDOUT','SYSLST'
rc = rexxipt(in.)
say 'rc from rexxipt was' rc
address link 'FTPBATCH ID=00'
say 'rc from ftpbatch was' rc
exit
```

In this example, the input lines are queued in a REXX stem variable named “*in.*” The name of the REXX stem variable is passed to FTPBATCH using the standard REXXIPT function. Parameters are passed to FTPBATCH using the ADDRESS LINK command. The response from the FTPBATCH job is passed to SYSLST in the partition running the REXX program. You cannot obtain responses to individual FTP commands. If you need the responses, consider using the SOCKET function from your REXX program as documented in the *TCP/IP FOR VSE Programmer's Guide*. The return codes from the ADDRESS LINK command are the same as the return codes from the external FTPBATCH client that are documented in this chapter.

Automatic FTP Client

This client is called the *automatic* FTP client because it is invoked automatically in response to the occurrence of an event. The event is specified in the DEFINE EVENT command. Currently, the only supported event is the occurrence of an entry in the POWER LST queue or the POWER PUN queue in a specified class. When you use the automatic FTP client, the following events occur:

- You issue a DEFINE EVENT command, which tells TCP/IP FOR VSE to watch for POWER LST queue entries or POWER PUN queue entries in the specified class. You use an option on this command to indicate which field within the entry will contain the name of the AutoFTP script that will be used. See the section “[Setup](#)” (page [2-38](#)) for more information.
- TCP/IP FOR VSE obtains the attributes of the POWER queue entry. These attributes are used for setting SETVAR variables that can be used in the AutoFTP script.
- The initial input is checked. If you do not provide the appropriate POWER JECL statements, they are built for you. For example, if there is no “JNM” parameter in the JECL, the job name is based on the actual name of the file. If there is no “CLASS” parameter, the class of the current subdirectory is used, with the exception of the “BIN” or “ALL” POWER areas. For security reasons, if there is a CLASS parameter in the JOB card, it is replaced with the CLASS value of the partition that the file is being written to. Again, this is ignored if a class of “ALL” or “BIN” is being used.
- The script name is obtained. If it is not defined, then the member is put on hold and will not be transferred. If the script cannot be found, a warning message occurs, and the entry is put on hold and is not transferred. A file is written to the POWER LST queue in the class that TCP/IP FOR VSE is monitoring. TCP/IP FOR VSE examines the * \$\$ LST statement for the name of the FTP script that it is to execute.

Note: The script’s name is contained in the DEST= parameter on the * \$\$ LST statement.

- TCP/IP FOR VSE executes the FTP script. The script processing takes place within the TCP/IP FOR VSE partition. The script consists of FTP client commands, which are described later in this chapter.

Setup

Use the following steps to set up the automatic FTP client:

1. Catalog a script that the automated process will use.
2. Define the script name to TCP/IP FOR VSE by issuing the DEFINE NAME command. You can issue this command interactively or as part of the initialization member for the stack.
3. Issue a DEFINE EVENT command to monitor a specific POWER class for automatic FTP processing. The system administrator can issue this command through the console interface or as part of the initializing member for TCP/IP FOR VSE.

Before you begin using the automatic FTP client, your system administrator must tell TCP/IP FOR VSE to monitor the class. The system administrator must also catalog the script to be executed. The execution of the script is performed only for queue entries that are either disposition *D* or *K* of the monitored POWER class.

The syntax of the DEFINE EVENT command that the system administrator uses for FTP is as follows. Only the parameters related to the automatic FTP client are shown. See the *TCP/IP FOR VSE Command Reference* for more information about this command.

```
DEFINE EVENT, ACTION=FTP, CLASS=class, ID=id, QUEUE=queuename
[ , POWERSYSID=powersysid]
```

The parameters in this syntax statement have the following meanings:

Parameter	Description
CLASS	The POWER LST or POWER PUN queue that TCP/IP FOR VSE is to monitor
ID	An arbitrary identifier. You use this identifier to delete the event with the DELETE EVENT command.
QUEUE	Either <i>LST</i> if TCP/IP FOR VSE is to monitor the POWER LST queue, or <i>PUN</i> if TCP/IP FOR VSE is to monitor the POWER PUN queue
POWERSYSID	The POWER system ID to monitor. This ID should be specified only when sharing POWER queues among multiple VSE systems.

Command Variables

The following command variables are available when creating AutoFTP scripts. These variables are always generated. If there are user-defined POWER field names, and these field names are included in the LST card (for example) that created them, then those variables will also be available. For details on defining user variables, see the IBM manual *VSE/POWER Administration and Operation*.

Variable	Description	Length
&CPUID	Numeric CPU identifier	6
&CPUSECS	Current time in 1/300 th -second units. This numeric value may be used as a random number for generating a unique file ID.	9
&CURDATE	Current date in the form <i>mmdyycc</i> , where <i>mm</i> is the month, <i>dd</i> is the day, <i>yy</i> is the year, and <i>cc</i> is a century of 19 or 20. Unlike with the external FTP batch client, you do not need to issue the SETVAR command to access this variable.	8
&CURTIME	Current time in the form <i>hhmmss</i> , where <i>hh</i> is the hour (using a 24-hour clock), <i>mm</i> is the minute, and <i>ss</i> is the second. Unlike with the external FTP batch client, you do not need to issue the SETVAR command to access this variable.	6
&ERROR	One of the following values, as set in the Auto FTP script: <ul style="list-style-type: none"> • <i>EXIT</i>, which indicates that an FTP script encountering an error should send a QUIT command to the foreign host and terminate immediately • <i>IGNORE</i>, which indicates that an FTP script encountering an error should ignore the error and continue processing with the next FTP command <p>CSI International recommends that you leave &ERROR set to <i>EXIT</i> unless you have a good reason to change it. In general, once an FTP command fails, the state of the FTP session is in doubt, and continuing with the script is potentially hazardous.</p>	6
&PWRBLDG	Building name assigned to the queue entry from the BLDG= parameter on the * \$\$ JOB statement	8
&PWRCDAT	Synonym for &PWRXDAT	

Variable	Description	Length
&PWRCLAS	LST or PUN class for the listing that is being transferred, from the * \$\$ LST or the * \$\$ PUN statement	1
&PWRCOPY	Number of requested copies as set by the COPY= parameter on the * \$\$ LST statement	2
&PWRCTIM	Synonym for &PWRXTIM	
&PWRDEPT	Department name assigned to the queue entry from the DEPT= parameter on the * \$\$ LST statement.	8
&PWRDEST	Name of the destination on the * \$\$ LST statement. This can also be the name of the “.L” book from which the automatic FTP client pulls FTP client commands (when HOST= is set to DEST).	8
&PWRDISP	Disposition for the LST or PUN queue entry from the DISP= parameter on the * \$\$ LST or * \$\$ PUN statement	1
&PWRDIST	Distribution code associated with the POWER queue entry from the DIST= parameter of the * \$\$ LST statement.	8
&PWRFLSH	FLASH field identifier. If not specified, it is blank. It is used for laser printers.	4
&PWRFLSN	Number of FLASH copies for laser printers. The default for this numeric field is 000.	3
&PWRFORM	Form identifier from the FNO= parameter on the * \$\$ LST statement.	4
&PWRFRMD	Special FORMDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRLCNT	Number of lines in the file	8
&PWRNAME	Name of the job that sent the output to the POWER LST or PUN class that TCP/IP FOR VSE is monitoring	8
&PWRNUMB	Number that POWER assigned to the job	5
&PWROFCB	Name of the Forms Control Buffer (FCB) specified on the FCB= parameter on the * \$\$ LST statement	8
&PWROUCB	Name of the character set specified on the UCS= parameter of the * \$\$ LST statement	8

Variable	Description	Length
&PWRPAGD	Special PAGEDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRPCNT	Number of pages in the file in numeric format. This is for LST data only.	8
&PWRPRGN	Name of the programmer from the PROGR= parameter on the * \$\$ JOB statement.	20
&PWRPRTY	Priority of the queue entry from the PRI= parameter on the * \$\$ JOB or * \$\$ LST statement	1
&PWRQDAT	Synonym for &PWRXDAT	
&PWRQTIM	Synonym for &PWRXTIM	
&PWRQUE	Either <i>PUN</i> or <i>LST</i> to specify the queue the listing came from	3
&PWRROOM	Room number assigned to the queue entry from the ROOM= parameter on the * \$\$ JOB statement	8
&PWRSID	System ID that is to process the queue entry, from the SYSID parameter on the * \$\$ LST statement	1
&PWRSUFF	Suffix that POWER assigned to the listing	3
&PWRTNOD	Target node name for the queue entry in a PNET environment, as specified on the LDEST parameter of the * \$\$ JOB statement for the LST queue and the PDEST parameter of the * \$\$ JOB statement for the PUN queue	8
&PWRUINF	User information field specified on the USER= parameter from the * \$\$ JOB statement or the * \$\$ LST statement. This field is arbitrary and may be changed with the PALTER command.	16
&PWRUSRI	VSE user ID associated with the job	8
&PWRXDAT	Date of the POWER queue entry in either <i>ddmmyyyy</i> or <i>mmddyyyy</i> format, depending on the IPL option of your VSE system. This value reflects when the entry was initially created or opened.	8

Variable	Description	Length
&PWRXTIM	Time of the POWER queue entry in <i>hhmmss</i> , where <i>hh</i> is the hour using a 24-hour clock. This value reflects when the entry was initially created or opened.	6
&QNUM	Unique POWER queue member identifier. (This is not the POWER job number.)	5

Return Codes

Return codes are not defined for the automatic FTP client. You see confirmation messages on the VSE console if the transfer is successful and failure messages if it is not.

If any FTP command in the script fails, error messages are displayed on the VSE system console. If the job fails to transfer the file, the disposition of the list queue entry is changed to *DISP=Y*. TCP/IP FOR VSE does not attempt to transfer any output that is set to *DISP=Y*. If you want to transfer a POWER LST or PUN queue entry that was set to *DISP=Y*, you can use the PALTER command to change the disposition back to *DISP=K* or *DISP=D* and AutoFTP will try the transfer again.

If you need help determining exactly which FTP command failed, use the SET DIAG=AUTO command, which outputs diagnostics to SYSLST that can be used in troubleshooting. SET DIAG=FTP can be used to determine what part of the FTP process failed, while the AUTO option helps in debugging the automated part of that process.

Queue Entry Disposition

The following table describes what happens to the queue entry after an automatic FTP is attempted.

Original Disposition	Disposition after FTP Succeeds	Disposition after FTP Fails
DISP=K	The entry remains on the queue and the disposition is changed to <i>DISP=L</i> .	The entry remains on the queue and the disposition is changed to <i>DISP=Y</i> .
DISP=D	The entry is deleted from the queue.	The entry remains on the queue and the disposition is changed to <i>DISP=Y</i> .

Interactive FTP Client

This client is called the *interactive* FTP client because it is invoked interactively as a CICS transaction. The interactive FTP client provides the CICS user with all of the flexibility of the batch FTP clients, but allows for manual (non-scripted) control over the FTP session.

To initiate a file transfer, you enter the CICS transaction *FTP*. If you use the VSE Interactive Interface, you should press PF9 (escape, mixed case) first. This clears your screen and allows you to enter the FTP transaction. The syntax of the FTP transaction is as follows:

```
FTP ipaddress,portnumber[,ID=idnumber]
```

The variables have the following meanings:

- *ipaddress* is the IP address of the foreign system. You can specify the IP address in dotted decimal notation (such as *192.168.0.7*) or as a symbolic name (such as *tcpip4vse.com*). There is no default. If you do not specify an IP address, TCP/IP FOR VSE opens a connection to the local TCP/IP FOR VSE partition and expects that, at some point in the FTP session, you will enter an OPEN command to establish a connection with the foreign TCP/IP. Note that the *ipaddress* parameter is positional.
- *portnumber* is the port number of the foreign system. The port number is specified as a decimal number. The default is *21*, and you should not change it unless you are certain that the foreign host has an FTP daemon listening at some port other than 21. The *portnumber* parameter is positional.
- *idnumber* is the ID number of the specific TCP/IP FOR VSE session that you want the interactive FTP client to connect to. The default is *00*. Do not specify this parameter unless you are running multiple versions of TCP/IP FOR VSE in the same VSE image and want to connect to a TCP/IP FOR VSE session with an ID number other than *00*. If you specify *idnumber*, make sure that it is the last parameter on the FTP command line.

Example 1

When you enter the IP address on the FTP command line, you receive a series of startup messages similar to those shown in the following box. Information entered by the user is in bold text.

```
FTP 100.50.90.90,21,ID=00
FTP200I FTP Client -- Startup -- Version 01.04.00(A)
FTP219I Copyright (c) 1995-2006 Connectivity Systems Incorporated
FTP211I Connecting to Port: 000021 at IP: 100.050.090.090 Id:00
FTP209I Establishing connection to TCP/IP partition
FTP212I Connection has been established
F: 220-FTPSERVE IBM VM V2R3 at SYS1, 12:26:12 EST SUNDAY 5/30/11
F: 220 Connection will close if idle for more than 5 minutes.
Enter Foreign User ID or "LOGOFF":
```

The first messages indicate that the FTP client is attempting to start the requested session. After the connection is established, message FTP212I is displayed along with other sign-on messages. In our example, we are connecting to a VM system. The FTP client prefaces all non-prompting messages from the foreign system with the tag *F:*. This contrasts with messages from the local host, which are prefaced with the tag *L:*.

You are now prompted for the following sign-on information:

- Foreign user ID
- Foreign password
- Local user ID
- Local password

Each host prompts you separately for the appropriate information. If the user ID and password required by the local host are identical to those required by the foreign host, you can simply press the enter key when prompted for the local values.

Example 2

In this example, we enter the FTP command with no parameters and see messages similar to the ones shown in the following box. Information entered by the user is shown in bold text.

```
FTP
FTP209I Establishing connection to TCP/IP partition
FTP212I Connection has been established
Ready:
L: 220-TCP/IP for VSE -- Version 01.04.00A -- FTP Daemon
   Copyright (c) 1995,2006 Connectivity Systems Incorporated
220 Service ready for new user.
Enter Local User ID, null, or "LOGOFF":
vseuser
L: USER vseuser
L: 331 User name okay, need password.
Enter Local Password, null, or "LOGOFF"
xxxxxxx
L: PASS xxxxxxxx
L: 230 User logged in, proceed.
Local host connection established.
Ready:
```

Because the FTP command does not contain an IP address, TCP/IP FOR VSE prompts you for a local user ID and password and opens a local connection. At this point, TCP/IP FOR VSE expects you to enter FTP client commands. One of the first commands you should enter is the OPEN command, which establishes a session with the foreign IP address.

Usage Notes

In both examples, note the FTP212I message. This message reflects the connection to the local TCP/IP FOR VSE partition, not the foreign host. If you do not see this message, make sure that the TCP/IP FOR VSE partition is running. If it is, make sure that enough FTP daemons are defined.

Another important thing to notice about your CICS display is the *More* indicator at the bottom. This tells you that there are more messages to be displayed and that you need to press the enter key to see them. When no more messages are waiting to be displayed, TCP/IP FOR VSE prompts you to enter another command. The prompt message of *Ready* is shown at the end of our second example box above. At the Ready prompt, you can enter the FTP client commands that are described in the remainder of this chapter.

VSE FTP Client Command Descriptions

This section covers the following topics:

- VSE file names, including information about how to enter a local-file specification. When commands require you to enter a variable called *local-file-specification*, this section explains the syntax you can use. The characteristics of the variable *foreign-file-specification* are not explained here because they are dependent on the foreign session. When a command requires you to specify information about the foreign file, you need to see the remote system's documentation for information on the syntax required by the foreign FTP daemon.
- FTP command descriptions, including information about syntax and usage. The commands are listed in alphabetic order. Command synonyms are listed together. When a command is not available to a client, that information is noted.

VSE File Names

Many FTP client commands require you to enter a local file name. When a command's syntax contains a variable called *local-file-name*, you can use a fully qualified file name, a partially qualified file name, or the name of an autonomous file.

Fully Qualified Name

A fully qualified public name is valid in the TCP/IP FOR VSE file system. To use a fully qualified name, begin by issuing a change directory command (or a local change directory command) to point to the root directory. To do this, issue one of the following commands:

```
CD \  
LCD \
```

Next, use the following format to specify the fully qualified name:

```
Qualifier-1.qualifier-2...qualifier-n
```

Here are some examples:

```
PUT POWER.LST.A.ROBERTJOB.01234  
PUT PRD1.MACLIB.WTO.A  
PUT VSAMCAT.CICS.CSD
```

Notice that *POWER*, *PRD1*, and *VSAMCAT* are all at the root level of the TCP/IP FOR VSE file system.

Partially Qualified Name

A partially qualified public name is relative to the local current working directory and is valid in the TCP/IP FOR VSE file system. To use a partially qualified public name, begin by issuing a change directory command to point to the directory where your file resides. The following examples use partially qualified directory names. The LCD/CD command is used to specify the directory where the filename resides before the PUT/GET is issued. In these examples, the VSE host is assumed to be the local host.

```
LCD POWER.LST.A
PUT ROBERTJOB.01234

LCD PRD1.MACLIB
PUT WTO.A

LCD VSAMCAT
PUT CICS.CSD
```

Autonomous Files

If a file has not been assigned a public name in the TCP/IP FOR VSE file system, you can still specify it using autonomous file syntax. *Autonomous files* are files that are not defined to TCP/IP FOR VSE using the DEFINE FILE command. They often are specified in batch programs that run external to the TCP/IP FOR VSE partition and that process a disk or tape file. To specify an autonomous file in a PUT or GET statement, use the following syntax:

```
%filename,type,recfm,lrecl[,blksize]
```

The variables have the following meanings:

- *filename* is a one- to seven-character DLBL (disk) or TLBL (tape) filename (also referred to as the DDNAME). It can also be the name of a virtual data space in an FTPBATCH partition. A data space name can be up to eight characters long. A percent sign (%) must precede *filename* and indicates that this is an autonomous file.

The %*filename* must have a matching entry in the partition label area (PARSTD), the system standard label area (STDLABEL), or the batch JCL that is invoking the request. For EXEC FTP and SOCKET FTP, the label used is from the partition label area for the main TCP/IP partition or, if not found there, the system standard label area. For FTPBATCH, the label used is from the partition where the job executes (the partition's PARTSTD or its temporary label area) or, if not found there, the system standard label area.

FTP issues a file OPEN with this string, so the DLBL/TLBL must be accessible to the partition in which FTP is running. If you transfer a tape file, the tape must be assigned to *SYS007* unless it is assigned dynamically by a tape manager such as BIM-EPIC.

- *type* is the type of file being transferred. The valid values are *SAM*, *ESDS*, or *KSDS* (for disk files); *TAPE* (for tape files); *EPIC* (for BIM-EPIC-controlled files); or *DSPACE* (for virtual data space files created with the FTPBATCH DEFINE DSPACE command—see the *TCP/IP FOR VSE Installation Guide* for more information).

Note: *EPIC* applies only to files that are cataloged by BIM-EPIC. If you specify *EPIC*, omit the *recfm*, *lrecl*, and *blksize* values. These values are retrieved automatically from the BIM-EPIC file catalog.

- *recfm* is the record format of the file and must be one of the following:

Value	Format	SAM	ESDS	KSDS	TAPE	DSPACE
F	Fixed unblocked	Yes	Yes	Yes	Yes	
FB	Fixed blocked	Yes	Yes	Yes	Yes	
V	Variable	Yes	Yes	Yes	Yes	
VB	Variable blocked	Yes	Yes	Yes	Yes	
SU	Spanned unblocked	Yes				
SB	Spanned blocked	Yes				
UN	Undefined (see text)	Yes			Yes	
UX	Undefined extended	Yes			Yes	
RAW	Raw data					Yes

- *lrecl* is the maximum logical record length to be used for the file being stored or retrieved.
- *blksize* is the maximum block size of the file being transferred and must be equal to or greater than the record length. It applies only to the blocked record formats. If *blksize* is omitted, TCP/IP FOR VSE uses the value of the maximum logical record length.

Note: When transferring to VSAM areas where a DEFINE CLUSTER is in effect, the *recfm*, *lrecl*, and *blksize* settings cannot be used to change the cluster attributes.

By default, remote FTP users are allowed to bypass the TCP/IP FOR VSE file system and specify VSE files using a local DLBL and the syntax shown above. The DYNFILE=OFF parameter in FTPBATCH (external server) and DEFINE FTPD (internal server) restricts remote FTP users to only accessing files through the TCP/IP FOR VSE file system by using a public name. The “LOCAL_DLBL OFF” command provides the same restriction for all internal FTP servers and is supported for earlier releases. (This command overrides the default DYNFILE=ON parameter in DEFINE FTPD.) See the *TCP/IP FOR VSE Installation Guide* for more information on FTP security.

Sequential disk files and tape files can be specified in FTPBATCH as autonomous files with an undefined record format (*UN* or *UX*). This permits transferring files like Dr. D backup files to a PC or a UNIX FTP server. Specifying *UN* or *UX* preserves the original physical record formats of the files when they are restored with a GET command from FTPBATCH.

Four examples of autonomous file transfers are shown below. The first two describe how the *UN/UX* record formats are used. These examples can be applied when transferring any sequential disk file or tape file.

Example 1

This example shows PUT and GET jobs that are used to transfer a DR. D backup disk file. This file has a *record structure* as defined by IETF RFC959. (The file consists of internally defined sequential records.) The record format specified in the autonomous file name syntax is *UN*.

```
// JOB BACKUP SD file to a PC FTP server
// DLBL DRDBUPV,'DRD.BACKUP',,SD
// EXEC FTPBATCH,SIZE=FTPBATCH
BINARY
LQUOTE STRU R
PUT %DRDBUPV,SAM,UN,32768 FTPBPTUN.BJB
QUIT
/*
/ &
```

```
// JOB RESTORE SD file from a PC FTP server
// DLBL DRDBUPV,'DRD.RESTORE',,SD
// EXEC FTPBATCH,SIZE=FTPBATCH
BINARY
LQUOTE STRU R
GET FTPBPTUN.BJB %DRDREST,SAM,UN,32768
QUIT
/*
/ &
```

The “LQUOTE STRU R” command in both jobs causes the local FTP daemon to include the record structure information defined in RFC959. Without this command, the record structure would be lost and the transferred backup file would be unusable. The LQUOTE is needed in the backup job to tell the local VSE FTP daemon to add the record structure information to the data stream being sent to the foreign FTP daemon.

The foreign FTP daemon is in the default “STRU F” (file structure) mode and receives the binary string of bytes that includes the record structure information. Later, in the restore job where the GET command is used to retrieve the backup file with the same physical data records on the disk, the LQUOTE STRU R tells the local FTP daemon on VSE to expect that the record structure is in the binary data being sent from the foreign FTP daemon.

The UX record format may be used in place of UN, and it is more efficient to do so. With the UX format, however, the “STRU R” command is not needed and must not be used. The same record format (UN or UX) must be specified in both the PUT and GET FTPBATCH jobs.

Example 2 This example shows a PUT job that sends a DR. D backup tape to a remote PC or a UNIX FTP server. The record format specified in the autonomous file name is *UN*.

```
// JOB FTPBPTUT
// OPTION LOG
// OPTION SYSPARM='00'
// TLBL DRDBUPV,'DRDVTAPE'
// LIBDEF *,SEARCH=lib.sublib
// EXEC FTPBATCH,SIZE=FTPBATCH
LOPEN
LUSER xxxxxxxx
LPASS xxxxxxxx
OPEN ip-addr-remote-ftp-server
USER yyyyyyyy
PASS yyyyyyyy
BINARY
LQUOTE STRU R
PUT %DRDBUPV,TAPE,UN,65535 FTPBPTUT.BJB
QUIT
/*
/ &
```

The following GET job can then be used at the DR site to recreate the DR. D backup tape from the file on the PC or the UNIX FTP server. This tape can be used as input to the Dr. D file restore utility.

```
// JOB FTPBGTUT
// OPTION LOG
// OPTION SYSPARM='00'
// TLBL DRDREST,'DRDVTAPE.GTUT'
// LIBDEF *,SEARCH=lib.sublib
// EXEC FTPBATCH,SIZE=FTPBATCH
LOPEN
LUSER xxxxxxxx
LPASS xxxxxxxx
OPEN ip-addr-remote-ftp-server
USER yyyyyyyy
PASS yyyyyyyy
BINARY
LQUOTE STRU R
GET FTPBPTUT.BJB %DRDREST,TAPE,UN,65535
QUIT
/*
/ &
```

Example 3 This example shows a PUT of an ESDS file. The record format in the autonomous file name is *F* (fixed unblocked).

```
* $$ JOB JNM=BATFTPV,CLASS=0,DISP=D,LDEST=(*,LJR)
* $$ LST CLASS=Z,DISP=H,DEST=(*,LJR)
// JOB BATFTPV
// DLBL LJRESDS, 'LJR.ESDS.TEST66.FILE',,VSAM,CAT=VSESPUC
// LIBDEF *,SEARCH=(BIMLIB.CSICNFG,BIMLIB.CSIPROD)
// EXEC FTPBATCH,SIZE=FTPBATCH
  LOPEN
  LUSER LJR
  LPASS abcd
  OPEN aaa.bbb.ccc.ddd
  USER xyz
  PASS testps
  CD remotedir
  LCD \
  PUT %LJRESDS,ESDS,F,175 LJRTST
QUIT
/*
/&
* $$ EOJ
```

This JCL includes the following DLBL statement:

```
// DLBL LSBESDS, 'LSB.ESDS.TEST66.FILE',,VSAM,CAT=VSESPUC
```

If this statement were not in the JCL, then it would need to be in the VSE system standard label area or the permanent partition label area for the FTPBATCH partition (BG in this example).

Example 4 This example shows a GET into an SD file on VSE. The record format in the autonomous file name is *VB* (variable blocked).

```
* $$ JOB JNM=BATFTPV,CLASS=0,DISP=D,LDEST=(*,LJR)
* $$ LST CLASS=Z,DISP=H,DEST=(*,LJR)
// JOB BATFTPV
// ASSGN SYS010,DISK,VOL=SYSWK1,SHR
// DLBL LSBSD8,'LSB.SD8.FILE',,SD
// EXTENT SYS010,SYSWK1,1,0,16680,10
// LIBDEF *,SEARCH=(BIMLIB.CSICNFG,BIMLIB.CSIPROD)
// EXEC FTPBATCH,SIZE=FTPBATCH
LOPEN
LUSER LSB
LPASS abcd
OPEN aaa.bbb.ccc.ddd
USER xyz
PASS testxxx
CD testvcat
LCD \
GET LSB.ESDS.TEST66.FILE %LSBSD8,SAM,VB,256
QUIT
/*
/&
* $$ E0J
```

FTP Client Commands

After you establish a session with an FTP client, you (or your batch jobs, your automatic FTP, or your FTP programs) use FTP commands to communicate with the local and foreign FTP servers. FTP commands can open connections, place either the local side or the foreign side of the session in a certain mode, or move data. In most cases, FTP commands may be used in all client modes. When this is not true, we document the FTP clients that can issue the command.

Each FTP client command can result in commands that are issued to, and responses received from, both local sessions and foreign sessions. The response to an FTP command is set by the local and foreign FTP daemons. All command responses are prefixed with a three-digit number as defined in RFC959.

ACCT The syntax is

```
ACCT account-information
```

The ACCT command sends account information to the foreign FTP daemon. Different FTP daemons process the ACCT command in different ways. For example, the VM/ESA FTP daemon expects a minidisk password with the ACCT command, while the MS Windows FTP daemon does not even support it.

Variable *account-information* contains account information in a format that is meaningful to the foreign FTP daemon.

This command is valid when you have established a connection with a foreign FTP daemon and need to send account information.

The anticipated response to the ACCT command depends on the foreign FTP daemon.

ACTIVE The syntax is

```
ACTIVE
```

This command causes the PASV command to be sent to the foreign FTP server, which returns the IP address and port on which it issues a passive (listen). The IP address and port number is returned to the PASV command in a 227 reply message. VSE then issues an active open to the foreign FTP server to establish the data connection. This command is useful in establishing FTP data connections through firewalls. It is valid when you have established local and foreign connections.

The anticipated response to the ACTIVE command is follows:

```
ACTIVE  
Ready:
```

ADAT The syntax is

```
ADAT data
```

The ADAT command sends authentication and security data to the foreign FTP daemon. This command is part of RFC2228 FTP Security Extensions.

The operand *data* is a Telnet string representing base-64-encoded security data. If a reply code indicating success is returned, the server may also use a string of the form “ADAT=base64data” as the text part of the reply if it wishes to convey security data back to the client.

The ADAT command and the associated replies allow the client and server to conduct an arbitrary security protocol. The security data exchange must include enough information for both peers to be aware of which optional features are available.

The ADAT command must be preceded by a successful AUTH command, and it cannot be issued once a security data exchange completes (successfully or unsuccessfully), unless it is preceded by an AUTH command to reset the security state.

The following table lists the recommended foreign server reply codes for various conditions.

Condition	Reply Code
Either the server has not yet received an AUTH command, or a prior security data exchange completed but the security state has not been reset with an AUTH command.	503
The server cannot base-64 decode the argument.	501
The server rejects the security data (a checksum fails, for example).	535
The server accepts the security data and requires additional data.	335
The server accepts the security data, but it does not require any additional data (that is, the security data exchange has completed successfully).	235

If the server is responding with a 235 or 335 reply code, then it may include security data in the text part of the reply as specified above.

If the ADAT command returns an error, the security data exchange fails and the client must reset its internal security state. If the client becomes unsynchronized with the server (for example, the server sends a 234 reply code to an AUTH command, but the client has more data to transmit), then the client must reset the server's security state.

APPEND

The syntax is

```
APPEND local-file-name foreign-file-name
```

The APPEND command appends records to a foreign file. If the foreign file exists, the local file is appended to it. If the foreign file does not exist, it is created. This is valid only for ESDS files accessed through VSAMCAT services or a LIBR text file.

The variables have the following meanings:

- *local-file-name* is a local file name that meets the qualifications outlined in the section “VSE File Names.”
- *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides. If you do not specify this variable, the FTP daemon uses the *local-file-name* to determine what the *foreign-file-name* should be. Each FTP daemon has its own method of determining what the *foreign-file-name* is, so the results vary.

This command is valid when you have established local and foreign connections.

The anticipated response to APPEND depends on the local and foreign FTP daemons. In general, you receive various messages from both daemons that are similar to the ones shown in this example:

```
APPEND PRD1.MACLIB.WTO.A WTO.A
L: PASV
L: 227 Entering Passive Mode (192,168,000,009,016,013).
F: PORT 192,168,000,009,016,013
F: 200 PORT command successful.
F: APPE WTO.A
F: 150 Opening ASCII mode data connection for WTO.A.
L: RETR PRD1.MACLIB.WTO.A
L: 150-File: PRD1.MACLIB.WTO.A
   Type: ASCII Recfm: FB Lrecl:   80 Blksize:   80
   CC=ON  UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
   Translate with US_ENG_03
150 File status okay; about to open data connection
F: 226 Transfer complete.
L: 226-Bytes sent:      236,078
   Records sent:      2,879
   Transfer Seconds:   5.34 (   46K/Sec)
   File I/O Seconds:  1.23 (   230K/Sec)
226 Closing data connection.
Ready:
```

ASCII The syntax is

```
ASCII
```

The ASCII command places both the local FTP session and the foreign FTP session in ASCII mode. This mode tells TCP/IP FOR VSE to perform the following functions:

- To translate outbound data to ASCII using the specified translation table
- To translate incoming data from ASCII to EBCDIC.

For information about data types and when it is appropriate to use ASCII transfers, binary transfers, and EBCDIC transfers, see the section “[Data Translation](#)” (page [2-7](#)).

The anticipated response is as follows:

```
ASCII
F: TYPE A N
F: 200 Type set to A.
L: TYPE A N
L: 200 Command okay.
```

AUTH The syntax is

```
AUTH {SSL|TLS}
```

The AUTH command proposes an authentication mechanism to the foreign FTP daemon. It is part of the RFC2228 FTP Security Extensions. *SSL* and *TLS* are the only valid operands. The foreign server must reply with a 234 message if it supports the SSL or TLS protocols.

Some servers allow the AUTH command to be reissued to establish new authentication. The AUTH command, if accepted, removes any state associated with prior FTP security commands. In this case the server must also require that the user reauthorize (that is, it must reissue some or all of the USER, PASS, and ACCT commands).

BATCH The syntax is

```
BATCH
```

The BATCH command allows using the MPUT and MGET commands from a batch environment.

BINARY The syntax is

```
BIN
BINARY
IMAGE
```

The BINARY command places both the local FTP session and the foreign FTP session in binary mode. This mode tells TCP/IP FOR VSE not to translate data from ASCII to EBCDIC. For information about data types and when it is appropriate to use ASCII transfers, binary transfers, and EBCDIC transfers, see the section “[Data Translation](#)” (page [2-7](#)).

This command is valid when you have established local and foreign connections.

The anticipated response is as follows:

```
BIN
F: TYPE I
F: 200 Type set to I.
L: TYPE I
L: 200 Command okay.
```

BYE The syntax is

```
BYE
QUIT
```

The BYE command closes both the local FTP session and the foreign FTP session and exits the client. If you want to terminate only the foreign session, use the CLOSE command. If you want to terminate only the local session, use the LCLOSE command.

This command is valid at any time.

The anticipated response is as follows:

```
BYE
F: QUIT
F: 221 termination message from Foreign Server if open
L: QUIT
L: 221 Service closing control connection.
FTP201I FTP Client -- Shutdown --
```

CD The syntax is

```
CD directory-name
```

The CD command performs the following functions:

- It changes the active directory in your foreign FTP daemon so that you can navigate through the file system of that daemon.
- It issues a Print Working Directory (PWD) command to display the new working directory.

Because you are working with the foreign FTP daemon and using its file system, the syntax of the CD command can vary depending on the active operating system.

Variable *directory-name* is the name of the directory that you want to be the active directory.

This command is valid when you have established a connection to the foreign FTP daemon.

The anticipated response is as follows:

```
CD /tools
F: CWD /tools
F: 250 CWD command successful.
F: PWD
F: 257 "/tools" is current directory.
Ready:
```

CDUP The syntax is

```
CDUP
```

The CDUP command is used to navigate through the file system of the foreign FTP daemon. If the foreign file system is a hierarchical file system, it changes the current working directory to the next highest level in the hierarchy. It is the same as the following command:

```
CD ..
```

This command is valid when you have established a connection to the foreign FTP daemon.

The anticipated response is as follows:

```
CDUP
F: CDUP
F: 250 CWD command successful.
F: PWD
F: 257 "/tools" is current directory.
Ready:
```

CLOSE The syntax is

```
CLOSE
```

The CLOSE command terminates a session with the foreign FTP daemon. After you issue a CLOSE command, you can use the OPEN command to initiate a session with another FTP daemon.

The CLOSE command is valid when you have established a connection to the foreign FTP daemon.

The anticipated response is as follows:

```
CLOSE
Foreign host connection closed.
Ready:
```

CONF The syntax is

```
CONF message
```

The CONF (confidentiality protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “confidential” message produced by a security-mechanism-specific confidentiality procedure. The foreign server decodes and verifies the encoded message. This command must be preceded by a successful security data exchange.

The following table lists the recommended foreign server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command because a checksum fails, for example.	535

Normally, the command is interpreted as an FTP command. An end-of-line code is not needed, but if one is included, it must be a Telnet end-of-line code, not a local end-of-line code.

DEBUG The syntax is

```
DEBUG [ON|OFF]
```

This command causes the VSE FTP client to issue debugging messages. The default is OFF.

DELETE The syntax is

```
DEL foreign-file-name  
DELETE foreign-file-name  
ERASE foreign-file-name
```

The DELETE command requests the foreign FTP daemon to delete a specific file. Each foreign FTP daemon responds in its own way, so it may or may not delete the file.

Variable *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides.

This command is valid when you have established a connection with the foreign host. You do not need an active connection with the local host. You probably need authorization from the foreign FTP daemon before you can use this command and have your request honored.

This command may be used with all FTP client types.

The anticipated response depends on the specific foreign FTP daemon. The following sample responses are from an MS Windows system:

```
ERASE wto.a  
F: DELE wto.a  
F: 250 DELE command successful.  
Ready:
```

DIR The syntax is

```
DIR directory-name  
NLIST directory-name  
NLST directory-name
```

The DIR command lists the requested directory on the foreign FTP system. The DIR, NLIST, and NLST commands are identical except that the DIR command returns directory information while the NLIST and NLST commands return a list of files without directory information.

Variable *directory-name* is the name of the directory that you want to list. The directory name that you specify may or may not contain wild cards, depending on the foreign FTP daemon and its capabilities. For VSE systems, the [LDIR](#) documentation (page [2.72](#)) contains a description of supported wild cards.

This command is valid when you have established a connection to the foreign FTP daemon.

The anticipated response depends on the specific foreign FTP daemon. The following sample responses are from an MS Windows system:

```
DIR
F: PORT 192,168,000,009,017,183
F: 200 PORT command successful.
F: LIST
F: 150 Opening ASCII mode data connection for /bin/ls.
03-16-10 03:42PM <DIR> annedoc
03-16-10 03:43PM <DIR> doc140
05-31-10 03:38PM 236078 f.a
05-31-10 03:38PM 236078 prd1.maclib.wto.a
08-06-10 07:49PM 461 RXT.JOB
04-29-10 06:27PM 266752 WhatsNew.ppt
05-31-10 03:52PM 236078 WTO.A
F: 226 Transfer complete.
Ready:
```

DUMPVAR The syntax is

```
DUMPVAR
```

DUMPVAR is a diagnostic command that allows the dumping of a variable's contents.

EBCDIC The syntax is

```
EBCDIC
EBC
```

The EBCDIC command tells the FTP client that files are to be transferred in EBCDIC. If you are transferring text files between EBCDIC-based mainframe hosts (MVS, OS/390, VM, VSE, and AS/400), this eliminates the translation of data to and from ASCII.

This command is valid when you have established a connection with a foreign host that understands EBCDIC. If the foreign host does not understand EBCDIC, it is likely to reject the command. If this happens, one side of your session is probably set to EBCDIC and the other side is probably set to ASCII. To avoid problems, you must quit and restart your session.

You must have established both a foreign connection and a local connection before you can use the EBCDIC command.

The anticipated response is as follows:

```
EBCDIC
F: TYPE E N
F: 200 Representation type is EBCDIC.
L: TYPE E N
L: 200 Command okay.
Ready:
```

ENC The syntax is

```
ENC message
ENCX message
```

The ENC (Privacy Protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “private” message produced by a security-mechanism-specific message integrity and confidentiality procedure.

The following table lists recommended foreign server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command because a checksum fails, for example.	535

Normally, the command is interpreted as an FTP command. An end-of-line (EOL) code is not needed, but if one is included, it must be a Telnet EOL code, not a local EOL code.

ERASE See DELETE command.

EVNTABND The syntax is

```
EVNTABND
```

The EVNTABND command causes the client to abend immediately.

This command is intended for diagnostic use.

EVNTDEBG The syntax is

```
EVNTDEBG
```

The EVNTDEBG command activates event-diagnostic debugging dumps.

EVNTINIT The syntax is

```
EVNTINIT
```

The EVNTINIT command initializes event debugging.

EVNTKILL The syntax is

```
EVNTKILL
```

The EVNTKILL command terminates event debugging.

EVNTREPT The syntax is

```
EVNTREPT
```

The EVNTREPT command creates a report of events.

EXECUTE The syntax is

```
EXECUTE membername  
EXEC membername
```

The EXECUTE command tells the FTP client to read the specified library member and to execute the FTP commands that are found in that member. When the EXECUTE command processor reaches the end of the member, it returns control to the environment that issued the EXECUTE command and continues processing. If you EXECUTE a member containing a QUIT command, the session is terminated and control does not return to the environment that issued the EXECUTE command.

The variable *membername* is the name of a library member located somewhere in the acceptable search sequence. If *membername* does not include a member type, a type of *.L* is assumed.

You can use the EXECUTE command in many situations. For example, you could invoke a series of commands from numerous batch jobs without having to repeat the commands in every batch job.

You can use command variables to pass information to a member that you are executing. For example, you could set command variable *&FILE* to *WTO.A* and put the following contents into the member that you are executing:

```
ERASE &FILE
PUT &FILE
```

If FTP runs in an environment where *&FILE* is defined as a command variable, symbolic substitution occurs and the FTP daemon receives the following commands:

```
ERASE WTO.A
PUT WTO.A
```

This command is valid at any time.

This command may not be used with the automatic FTP client.

The anticipated response is as follows:

```
EXEC STATCMDS.L
Commands will be taken from STATCMDS
Command:STATUS
F: STAT
F: 211-csiserver Microsoft Windows FTP Server status:
  Version 3.0
  Connected to 192.168.0.9
  Logged in as robert
  TYPE: ASCII, FORM: Nonprint; STRUcture: File; transfer M
  No data connection
211 End of status.
Command:LSTATUS
L: STAT
L: 212-Currently selected values:
  Type: ASCII Recfm: FB Lrecl:...80.Blksize:...80.
  CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
  Translate with US_ENG_03
212 End of status
Command:EXECEOF
Ready:
```

FEAT The syntax is

```
FEAT
```

The FEAT command is a request for the features supported by the foreign FTP server.

GET, GETX The syntax is

```
GET foreign-file-name (local-file-name)  
GETX !foreign-file-name! (!local-file-name!)  
RETRIEVE !foreign-file-name! (!local-file-name!)
```

The GET command retrieves a file from the foreign system and stores it in the local VSE file system.

The variables have the following meanings:

- *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides.
- *local-file-name* is the name you want to use when you write the file to your local system. The name must meet the qualifications outlined in the section [“VSE File Names”](#) (page 2-46). The variable has the same meaning for the GET and GETX commands, but GETX requires the name to be preceded and followed by a delimiting character. This restriction enables you to specify a local file name that contains embedded blanks. The delimiting character cannot be a period.

The *local-file-name* does not have to use the same delimiting character that *foreign-file-name* uses. If you do not specify a *local-file-name*, the file is stored in the VSE file system under its original name. Note that both the *foreign-file-name* and *local-file-name* variables are specified in the context of the foreign and local current working directories. The file name is appended to the current working directory on both sides of the connection to obtain the real file names. Most (but not all) FTP daemons allow you to specify a fully qualified file name starting with the ‘/’ (or ‘\’ for PC file systems). TCP/IP FOR VSE does not allow you to specify a fully qualified file name. In every case, it appends the *local-file-name* to the current working directory.

This command is valid when you have established local and foreign connections. In addition, you probably need to be authorized to read the file on the foreign system, and you definitely need to be authorized to write the file on the local VSE system.

The following example contains a GETX command. The example assumes that you have an MS Windows file named *TCP/IP What's New Presentation*. To upload your file to VSE, you can issue the following command from the FTP client:

```
GETX ?TCP/IP What's New Presentation.txt? ?Whatsnew.txt?
```

The anticipated response depends on the specific foreign FTP daemon. The following sample responses are from an MS Windows FTP daemon:

```
GET ntdata.dat ntfile.text
L: PASV
L: 227 Entering Passive Mode (192,168,000,009,021,176).
F: PORT 192,168,000,009,021,176
F: 200 PORT command successful.
L: STOR ntfile.text
L: 150-File: PRD2/ROBERT/NTFILE.TEXT
   Type: Ascii  Recfm: FB  Lrecl:   80.Blksize:   80
   CC=ON  UNIX=ON  RECLF=OFF TRCC=OFF CRLF=ON
   Translate with US_ENG_03
150 File status okay; about to open data connection
F: RETR ntdata.dat
F: 150 Opening ASCII mode data connection for ntdata.dat(236078 bytes)
L: 226-Bytes sent:      236,078
   Records sent:      2,879
   Transfer Seconds:   8.49 (   28K/Sec)
   File I/O Seconds:   1.83 (  230K/Sec)
226 Closing data connection.
F: 226 Transfer complete.
Ready:
```

HELP The syntax is

```
HELP
HELP command-name
```

The HELP command provides the following types of information:

- With no argument, you receive a list of commands recognized by the FTP client.
- If you specify *command-name*, you receive information about that command.

You can customize the information returned by the HELP command. The HELP command with no operands displays the contents of member *FTPC.K* in the TCP/IP FOR VSE distribution library. The HELP command with an operand displays the contents of the operand as a K book in the TCP/IP FOR VSE distribution library. For example, *HELP MGET* displays the contents of the *MGET.K* member in the TCP/IP FOR VSE distribution library.

IMAGE See BINARY command.

LACCT The syntax is

LACCT <i>account-information</i>

The LACCT command sends account information to the local FTP server.

The variable *account-information* contains account information in a format that is meaningful to the local FTP daemon.

LADAT The syntax is

LADAT <i>data</i>

The LADAT command sends authentication and security data to the local FTP daemon. It is part of RFC2228 FTP Security Extensions.

The operand *data* is a Telnet string representing base-64-encoded security data. If a reply code indicating success is returned, the server may also use a string of the form “LADAT=base64data” as the text part of the reply if it wishes to convey security data back to the client.

The LADAT command, and the associated replies, allows the client and server to conduct an arbitrary security protocol. The security data exchange must include enough information for both peers to be aware of which optional features are available.

The LADAT command must be preceded by a successful LAUTH command, and it cannot be issued once a security data exchange completes (successfully or unsuccessfully), unless it is preceded by an LAUTH command to reset the security state.

The following table lists the local server reply codes for various conditions.

Condition	Reply Code
Either the server has not yet received an LAUTH command, or a prior security data exchange completed but the security state has not been reset with an LAUTH command.	503
The server cannot base-64 decode the argument.	501
The server rejects the security data (a checksum fails, for example).	535

Condition	Reply Code
The server accepts the security data and requires additional data.	335
The server accepts the security data, but it does not require any additional data (that is, the security data exchange has completed successfully).	235

If the server is responding with a 235 or 335 reply code, then it may include security data in the text part of the reply as specified above.

If the LADAT command returns an error, the security data exchange fails and the client must reset its internal security state. If the client becomes unsynchronized with the server (for example, the server sends a 234 reply code to an LAUTH command, but the client has more data to transmit), then the client must reset the server's security state.

LAPPEND

The syntax is

```
LAPPEND foreign-file-name [local-file-name]
```

The LAPPEND command retrieves a file from the foreign system and appends it to an existing file in the local VSE file system. You can use LAPPEND to append data to VSAM ESDS files or VSE libraries.

The variables have the following meanings:

- *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides.
- *local-file-name* is the name of the file you want to append to in your local system. The name must meet the qualifications outlined in the section "[VSE File Names](#)" (page 2-46). The variable has the same meaning for the GET command. If you do not specify a *local-file-name*, the file is appended in the VSE file system under its original name.

Note: Both the *foreign-file-name* and *local-file-name* variables are specified in the context of the foreign and local current working directories. The file name is appended to the current working directory on both sides of the connection to obtain the real file names. Most (but not all) FTP daemons allow you to specify a fully qualified file name starting with the '/' (or '\ for PC file systems). TCP/IP FOR VSE does not allow you to specify a fully qualified file name. In every case, it appends the *local-file-name* to the current working directory.

This command is valid when you have established local and foreign connections. In addition, you probably need to be authorized to read the file on the foreign system and you definitely need to be authorized to write the file on the local VSE system.

The following example shows how the LAPPEND command is used:

```
LAPPEND updates.txt libmem.txt
```

LAUTH The syntax is

```
LAUTH {SSL|TLS}
```

The LAUTH command proposes an authentication mechanism to the local FTP daemon. It is part of RFC2228 FTP Security Extensions

SSL and *TLS* are the only valid operands.

The local server must reply with a 234 message if it supports the SSL or TLS protocols.

The server allows the LAUTH command to be reissued to establish new authentication. The LAUTH command, if accepted, removes any state associated with prior FTP security commands. The server must also require that the user reauthorize (that is, reissue some or all of the LUSER, LPASS, and LACCT commands) in this case.

LCD The syntax is

```
LCD directory-name
```

The LCD command performs the following functions:

- It changes the active directory in your local FTP daemon so that you can navigate through the file system of that daemon. In general, the local file system is the VSE system that runs FTP, so this command navigates through VSE directories and VSE files.
- It issues a Print Working Directory (PWD) command to display the new working directory.

The variable *directory-name* is the name of the directory that you want to be the active directory. If *directory-name* is a '/', the local session is placed in UNIX emulation mode. For more information about UNIX emulation mode, see the section [“UNIX Emulation Mode”](#) (page 2-5).

This command is valid when you have established a connection to the local FTP daemon.

The anticipated response is as follows.

```
Ready:
LCD \ijsysrs
L: CWD \ijsysrs
L: 250 Requested file action okay, completed.
L: PWD
L: 257 "IJSYSRS"
Ready:
```

LCDUP The syntax is

```
LCDUP
```

The LCDUP command is used to navigate through the file system of the local FTP daemon. It changes the current working directory to the next highest level in the hierarchy. It is the same as the following command:

```
LCD ..
```

This command is valid when you have established a connection to the local FTP daemon.

The anticipated response is as follows:

```
LCDUP
L: CDUP
L: 200 Command okay.
L: PWD
L: 257 ""
Ready:
```

LCLOSE The syntax is

```
LCLOSE
```

The LCLOSE command terminates a session with the local FTP daemon. Use the LCLOSE command followed by the LOPEN/LUSER/LPASS combination to change the local user ID before you initiate a request for data access.

The LCLOSE command is valid when you have established a connection to the local FTP daemon.

Use the LCLOSE command followed by the LOPEN/LUSER/LPASS combination to change the local user ID before you initiate a request for data access.

The anticipated response is as follows:

```
LCLOSE
Local host connection closed.
Ready:
```

LCONF The syntax is

```
LCONF message
```

The LCONF (Local Confidentiality Protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “confidential” message produced by a security-mechanism-specific confidentiality procedure. The local server decodes and verifies the encoded message. This command must be preceded by a successful security data exchange.

The following table lists the local server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command (a checksum fails, for example).	535

Normally, the command is interpreted as an FTP command. An end-of-line (EOL) code need not be included, but if one is included, it must be a Telnet EOL code, not a local EOL code.

LDELETE The syntax is

```
LDELETE local-file-name
LERASE local-file-name
```

The LDELETE command requests the local FTP daemon to delete a specific file. In general, the local FTP daemon calls the TCP/IP FOR VSE security exit to determine whether you are authorized to delete the file.

Variable *local-file-name* is a local file name that meets the qualifications outlined in the section [“VSE File Names”](#) (page [2-46](#)).

This command is valid when you have established a connection with the local host. You do not need an active connection with the foreign host.

The anticipated response is as follows:

```
LERASE wto.a
L: DELE wto.a
L: 200 Command okay.
Ready:
LERASE wto.a
L: DELE wto.a
L: 550 Action not taken: Entry not in sub-directory.
Ready:
```

In our example, note that the first LERASE command is successful. The second LERASE command is not successful because the entry is already deleted. The second LERASE command receives message number 550 from the local daemon. Under normal circumstances, this is enough to terminate the local FTP session. Depending on which FTP client you are using, you can tell TCP/IP FOR VSE not to terminate the FTP session for a 550 level message. To do this, use the SET IGNORERR command or the &ERROR variable.

LDIR The syntax is

```
LDIR directory-name
LNLIST directory-name
LNLST directory-name
```

The LDIR command lists the specified directory on the local VSE system. The LDIR, LNLIST, and LNLST commands are identical except that LDIR returns directory information, while LNLIST and LNLST return a list of file names in the current local directory without directory information.

Variable *directory-name* is the name of a directory you want to list. The directory name may or may not contain wild cards, depending on the local FTP daemon and its capabilities. In general, the following example shows how the supported wild cards work. For our example, assume that you have a VSE sublibrary containing the following five members:

```
file1.text
file2.text
file3.ppt
pat.doc
nyjetsaregreat.jpg
```

The following table shows specific LDIR commands and the effects of using wild cards:

Sample Command	Files Contained in Output Listing
LDIR	file1.text, file2.text, file3.ppt, pat.doc, nyjetsaregreat.jpg
LDIR F*	file1.text, file2.text, file3.ppt
LDIR F*.PPT	file3.ppt
LDIR *.jpg	nyjetsaregreat.jpg
LDIR F*.T*	file1.text, file2.text

This command is valid when you have established a connection to the local FTP daemon. The anticipated response depends on the type of VSE directory you list.

LENC The syntax is

LENC <i>message</i>
LENCX <i>message</i>

The LENC (Local Privacy Protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “private” message produced by a security-mechanism-specific message integrity and confidentiality procedure.

The following table lists the local server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command because a checksum fails, for example.	535

An end-of-line (EOL) code is not needed, but if included, it must be a Telnet (not a local) EOL code.

LERASE See LDELETE command.

LFEAT The syntax is

LFEAT

The LFEAT command requests the features supported by the local FTP server.

LGOTEOJ The syntax is

LGOTEOJ

This command tells a locally attached FTP daemon to terminate. It is used by FTPBATCH to terminate the partition.

LMIC The syntax is

LMIC <i>message</i> LMICX <i>message</i>

The LMIC (Local Integrity Protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “safe” message produced by a security-mechanism-specific message integrity procedure. The local server decodes and verifies the encoded message. This command must be preceded by a successful security data exchange.

The following table lists the local server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command because a checksum fails, for example.	535

Normally, the command is interpreted as an FTP command. An end-of-line (EOL) code need not be included, but if one is included, it must be a Telnet EOL code, not a local EOL code.

LMKDIR The syntax is

```
LMKDIR
```

The LMKDIR command requests that a directory be created by the local FTP server.

LNLIST, LNLST See LDIR command.

LNOOP, LNOP The syntax is

```
LNOOP
LNOP
```

These commands send a no-operation request to the local FTP server to test the connections to that server. They do not generate any action.

The LNOOP and LNOP commands are valid when a connection is established with the local server.

The anticipated response to the LNOP command is as follows:

```
LNOP
L: NOOP
L: 200 Command okay.
Ready:
```

LOPEN The syntax is

```
LOPEN ipaddress portnumber
```

The LOPEN command establishes a session with the local FTP daemon. For more information about when an explicit LOPEN command is required, see the documentation for each individual FTP client.

The variables have the following meanings:

- *ipaddress* is the IP address of the system you are connecting with. You can specify the IP address in dotted decimal notation or as a symbolic name. The name is resolved by the Domain Name Server or the DEFINE NAME command. The *ipaddress* parameter is positional.
- *portnumber* is the port number of the local system. The port number is specified as a decimal number. The default is to leave it blank, which causes it to use the next available free port, and you should not change it. The *portnumber* parameter is positional.

This command must be the first command in an internal or external FTPBATCH client session in which you did not specify the IP parameter in the parameter string. The primary reason for using the LOPEN command is to reestablish a connection with the local FTP daemon following the execution of an LCLOSE command.

For internal and external FTPBATCH clients in which you omit the IP= parameter at startup, the LOPEN command must be the first command in the input stream.

For the interactive FTP client, LOPEN is accepted only when the startup transaction omits the IP address.

For internal, external, and programmable clients, the next command in the input stream must be a LUSER command specifying a valid user ID on the local system, and the command after that must be an LPASS command specifying a valid password for that user ID.

For the interactive client, the anticipated response is a prompt for a user ID.

LPASS The syntax is

```
LPASS password
```

The LPASS command supplies a password to the local FTP daemon. The variable *password* is a valid password for the user ID specified in the preceding LUSER command.

This command is valid only when it follows an LUSER command. It must immediately follow the LUSER command for all clients except the interactive FTP client. The interactive client responds to LOPEN by prompting the CICS terminal user for user ID information, and responds to the user ID information with a password prompt.

The anticipated response is feedback from the local FTP daemon indicating whether the local user ID and password combination is accepted.

LPBSZ The syntax is

```
LPBSZ buffsize
```

The LPBSZ command is a local protection buffer size request. It is part of RFC2228 FTP Security Extensions.

The argument *buffsize* is a decimal integer representing the maximum size, in bytes, of the encoded data blocks to be sent or received during file transfer. This number shall be no greater than can be represented in a 32-bit unsigned integer.

This command allows the FTP client and server to negotiate a maximum protected buffer size for the connection. There is no default size; the client must issue a LPBSZ command before it can issue the first LPROT command.

The LPBSZ command must be preceded by a successful security data exchange.

The following reply codes should be issued by the local server:

- If the server cannot parse the argument, or if the argument does not fit in 32 bits, the server should respond with a 501 reply code.
- If the server has not completed a security data exchange with the client, it should respond with a 503 reply code.

Otherwise, the server must reply with a 200 reply code. If the size provided by the client is too large for the server, it must use a string of the form “PBSZ=*number*” in the text part of the reply to indicate a smaller buffer size. The client and the server must use the smaller of the two buffer sizes if both buffer sizes are specified.

LPROT

The syntax is

LPROT <i>level</i>

The LPROT command is a local data connection protection level request. It is part of RFC2228 FTP Security Extensions.

The argument *level* is a single Telnet character code specifying the data channel protection level.

This command indicates to the local server what type of data channel protection the client and server are using. The following codes are assigned:

- C (Clear)
- S (Safe)
- E (Confidential)
- P (Private)

The default protection level, if no other level is specified, is Clear. The Clear protection level indicates that the data channel carries the raw data of the file transfer with no security applied. The Safe protection level indicates that the data is integrity protected. The Confidential protection level indicates that the data is confidentiality protected. The Private protection level indicates that the data is both integrity and confidentiality protected.

It is reasonable for a security mechanism to not provide all data-channel protection levels. It is also reasonable for a mechanism to provide more protection at a level than is required (for example, a mechanism might provide Confidential protection, but include integrity-protection in that encoding, due to API or other considerations).

The LPROT command must be preceded by a successful protection buffer size negotiation.

The following tables lists the local server reply codes for various conditions.

Condition	Reply Code
The server does not understand the specified protection level.	504
The current security mechanism does not support the specified protection level.	536
The server has not completed a protection buffer-size negotiation with the client.	503
No previous LPBSZ command was issued, and the LPROT command is rejected.	503
The server is not willing to accept the specified protection level.	534
The server is not able to accept the specified protection level, such as if a required resource is unavailable	431

Otherwise, the local server must reply with a 200 reply code to indicate that the specified protection level is accepted.

LPWD The syntax is

```
LPWD
```

The LPWD command displays the current working directory of the local FTP daemon.

This command is valid when you have established a connection with the local host.

The anticipated response is as follows:

```
LPWD
L: PWD
L: 257 "/ijsysrs/syslib"
Ready:
```

LQUOTE The syntax is

```
LQUOTE local-ftp-server-command
```

The LQUOTE command issues a command directly to the local FTP daemon. Normally, FTP client commands are translated to commands that FTP daemons understand. The commands that FTP daemons understand are standard. The LQUOTE command bypasses this translation mechanism and allows you to send commands directly to the local FTP daemon.

It is usually hazardous to issue the LQUOTE command instead of commands such as SITE, GET, PUT, and so forth. When you send commands directly to the local daemon, you bypass any preliminary command setup that may be required for the commands to function properly.

Variable *local-ftp-server-command* is any command that the local FTP daemon understands. For a list of commands supported by the local FTP daemon, you can issue the LQUOTE HELP command.

This command is valid when you have established a connection with the local host. The anticipated response varies depending on the command.

LRENAME The syntax is

```
LRENAME old-file-name new-file-name  
LRENAMEX old-file-name new-file-name  
LRENX old-file-name new-file-name
```

The LRENAME command requests the local FTP daemon to rename a specific file on the local VSE system. Some VSE file types support this command and some do not. See the *TCP/IP FOR VSE Installation Guide* for a list of file types that do support the LRENAME function. The local FTP daemon calls the TCP/IP FOR VSE security exit to determine whether you are authorized to rename the file. LRENAMEX and LRENX are synonyms for LRENAME.

The variables have the following meanings:

- *old-file-name* is the name of the file that you want to rename
- *new-file-name* is the name that you want to assign to the file.

If the local FTP daemon cannot locate the file in the current working directory, the LRENAME command is not successful and the local daemon issues message number 550. Under normal circumstances, this is enough to terminate the local FTP session. Depending on which FTP client you are using, you can tell TCP/IP FOR VSE not to terminate the FTP session for a 550 level message. To do this, use the SET IGNORERR command or the &ERROR variable.

This command is valid when you have established a connection with the local host and are authorized to rename the file.

The anticipated response is as follows. Note that message number 350 is always returned by TCP/IP FOR VSE whether the file exists or not.

```
LRENAME robert.a wto.a
L: RNFR robert.a
L: 350 File exists, ready for destination name
L: RNT0 wto.a
L: 250 Requested file action okay, completed.
Ready:
```

LRMDIR The syntax is

```
LRMDIR dirname
```

This LRMDIR command is a request that a local directory be removed.

The variable *dirname* is the directory name on the local FTP server.

LSITE The syntax is

```
LSITE local-ftp-site-command
```

The LSITE command issues a site-specific command directly to the local FTP daemon.

The variable *local-ftp-site-command* is any SITE command that the local FTP daemon understands. For a list of commands that are supported by the local FTP daemon, you can issue the LSITE HELP command. This command is valid when you have established a connection with the local server.

The anticipated response varies depending on the command.

LSTATUS The syntax is

```
LSTATUS
```

The LSTATUS command displays the status of the local VSE FTP daemon. The information that is displayed includes the parameters that a file transfer command like GET or PUT would use if it executed at that time.

This command is valid when you have opened a connection with the local server.

The anticipated response is shown in the following example:

```
LSTATUS
L: STAT
L: 212-Currently selected values:
    Type: ASCII Recfm: FB Lrecl:      80Blksize:      80
    CC=ON  UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
    Translate with US_ENG_03
212 End of status
Ready:
```

For information on the UNIX, CC, RECFM, TRCC, and CRLF values, see the section [“Categories of SITE Commands”](#) (page [2-113](#)).

LSYSTEMS

The syntax is

```
LSYSTEMS
LSYS
```

The LSYSTEMS command displays the system type of the local FTP daemon. VSE systems can return one of two system types, which are *VSE* for VSE mode and *UNIX* for UNIX mode. The FTP standard requires the client to look at the first word following message number 215 in the response. This command is valid when you have established a connection with the local host.

The anticipated response is as follows. You receive one of the two responses shown, depending on whether your local session is running in VSE or UNIX mode.

```
LSYSTEMS
L: SYSTEMS
L: 215 VSE system type
Ready:

LSYSTEMS
L: 215 UNIX Simulation on VSE system.
```

LUSER

The syntax is

```
LUSER userid
```

The LUSER command supplies a user ID to the local FTP daemon. This command is required after the execution of an LOPEN command for all FTP clients except the interactive client. The interactive client prompts you for a user ID.

Variable *userid* is a user ID that is valid to the local TCP/IP FOR VSE session.

The interactive client responds to `LUSER` by prompting the CICS user for a password. TCP/IP FOR VSE then checks the user ID and password for validity. For all other FTP clients, the next command in the input stream should be the `LPASS` command.

If you specify a local user name of *anonymous*, TCP/IP FOR VSE forces UNIX mode on. Do not specify a local user name of *anonymous* if you do not want to run in UNIX compatibility mode.

MGET The syntax is

```
MGET foreign-file-name
```

The MGET command enables you to transfer multiple files from a foreign host with a single operation. You can use this command only if the local current working directory points to a VSE Librarian subdirectory or to a POWER RDR, PUN, or LST queue. The local file names are derived from the foreign file names.

The MGET command examines every file in the current working directory of the foreign FTP server looking for matches to the foreign file name that you specify. To predict which files a particular MGET command is going to transfer, you can use the NLST or DIR commands. In general, the MGET command lists a directory on the foreign system, copies the output, and transfers every file listed in the output.

Variable *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides. You must consult the documentation for the foreign host to see what files and directories are eligible for MGET and how the wild card operations might work on the foreign system.

If you use the interactive FTP client to issue the MGET command, you can press the enter key at any time during the file transfer to interrupt it. When you attempt the interrupt, you are prompted with the following message:

```
MGET INTERRUPTED. ENTER "ABORT" or "CONTINUE"
```

ABORT terminates the transfer, and *CONTINUE* allows it to proceed.

If you use the internal, external, automatic, or programmable clients to issue the MGET command, you cannot interrupt the file transfer unless you cancel the job.

This command is valid when you have established local and foreign connections. In addition, you probably need to be authorized to read the files on the foreign system and you definitely need to be authorized to write the files on the local VSE system.

The anticipated response depends on the specific foreign FTP daemon.

The following sample responses are from an FTP daemon running on TCP/IP for VM. In the first part of this example, an LCD command is issued to change the directory:

```
LCD PRD2.ROBERT
L: CWD PRD2.ROBERT
L: 250 Requested file action okay, completed.
L: PWD
L: 257 "PRD2.ROBERT"
Ready:
```

An MGET command is then issued:

```
MGET *.TEXT
F: PORT 192,168,000,009,016,002
F: 200 Port request OK.
F: NLST *.TEXT
F: 125 List started OK
F: 250 List completed successfully.
L: PASV
L: 227 Entering Passive Mode (192,168,000,009,016,003).
F: PORT 192,168,000,009,016,003
F: 200 Port request OK.
L: STOR A06MS.TEXT
L: 150-File: PRD2.ROBERT.A06MS.TEXT
  Type: Ascii  Recfm: FB  Lrecl:   80  Blksize:   80
  CC=ON  UNIX=OFF  RECLF=OFF  TRCC=OFF  CRLF=ON
  Translate with US_ENG_03
150 File status okay; about to open data connection
F: RETR A06MS.TEXT
F: 150 Sending file 'A06MS.TEXT'  FIXrecfm   80
L: 226-Bytes sent:           820
  Records sent:              10
  Transfer Seconds:         2.05 (      K/Sec)
  File I/O Seconds:        .32 (      0K/Sec)
226 Closing data connection.
F: 250 Transfer completed successfully.
MGET Requirements completed
Ready:
```

MIC The syntax is

MIC <i>message</i> MICX <i>message</i>

The MIC (Integrity Protected) command is part of RFC2228 FTP Security Extensions.

The operand *message* is a Telnet string consisting of a base-64 encoded “safe” message produced by a security-mechanism-specific message integrity procedure. The remote server decodes and verifies the encoded message. This command must be preceded by a successful security data exchange.

The following table lists recommended foreign server reply codes for various conditions.

Condition	Reply Code
The server has not completed a security data exchange with the client.	503
The server cannot base-64 decode the argument.	501
The server rejects the command because it is not supported by the current security mechanism.	537
The server rejects the command because a checksum fails, for example.	535

Normally, the command is interpreted as an FTP command. An end-of-line (EOL) code does not need to be included, but if one is included, it must be a Telnet EOL code and not a local EOL code.

MKDIR The syntax is

MKDIR <i>dirname</i>

The MKDIR command requests that a foreign directory be created.

The variable *dirname* is the name of the directory to be created on the foreign server.

MODE The syntax is

```
MODE [S|B]
```

TCP/IP FOR VSE can transfer data using one of the following two modes:

- Stream mode, which transfers data as a series of bytes
- Block mode, which transfers data as a series of records.

The MODE command enables you to specify whether a data transfer is to use stream mode or block mode. You specify *S* for stream mode or *B* for block mode. The default is *S*.

All FTP daemons support stream mode. There are only a few FTP daemons that support block mode, including OS/390 and VM/ESA.

Block mode tends to be a more reliable data-transfer mechanism because it understands and uses end-of-record and end-of-file concepts in its processing. Stream mode, however, assumes that it has reached the end of the file when the sending side closes the connection. Unfortunately, the receiving side cannot really know whether the connection closed because there was no more data to send or whether there was some error condition. You should always use block mode to send record-oriented data if the foreign FTP daemon supports it.

This command is valid when you have established local and foreign connections. In addition, the foreign FTP daemon must support the requested mode.

The anticipated response is as follows:

```
MODE B
F: MODE B
F: 200 Data transfer mode is Block.
L: MODE B
L: 200 Command okay.
Ready:
```

MPUT The syntax is

```
MPUT local-file-name
```

The MPUT command enables you to transfer multiple files from a local host with a single operation. You can use this command only if the local current working directory points to a VSE Librarian subdirectory or to a POWER RDR, PUN, or LST queue. The foreign file names are derived from the local file names in a way that is unique to each foreign FTP daemon and foreign file system.

If you need to control the names assigned by the foreign FTP daemon, consider using the programmable FTP client. Using this client, you can issue an LNLIST command, trap the output, and use the data in the output to issue a series of PUT commands.

The MPUT command examines every file in the local current working directory looking for matches to the local file name that you specify. To predict which files a particular MPUT command is going to transfer, you can use the LNLIST or LDIR commands. In general, the MPUT command lists a directory on the local system, copies the output, and transfers every file listed in the output.

The variable *local-file-name* is a local file name that meets the qualifications outlined in the section [“VSE File Names”](#) (page 2-46). You can select multiple files by including the asterisk (*) wild-card character in the name.

The example below shows how this wild card works. For this example, assume that you have the following five files in the current working directory of your VSE sublibrary:

```
file1.text
file2.text
file3.ppt
pat.doc
nyjetsaregreat.jpg
```

The following table shows the effect of specific MPUT commands that use the asterisk wild card. No other wild-card characters can be used.

Command	Files Selected and Transferred
MPUT	file1.text, file2.text, file3.ppt, pat.doc, nyjetsaregreat.jpg
MPUT F*	file1.text, file2.text, file3.ppt
MPUT F*.PPT	file3.ppt
MPUT *.jpg	nyjetsaregreat.jpg
MPUT F*.T*	file1.text, file2.text

If you use the interactive FTP client to issue the MPUT command, you can press the Enter key at any time during the file transfer to interrupt it.

When you attempt the interrupt, you are prompted with the following message:

```
MPUT INTERRUPTED. ENTER "ABORT" or "CONTINUE"
```

ABORT terminates the transfer, and *CONTINUE* allows it to proceed.

If you use the internal, external, automatic, or programmable clients to issue the MPUT command, you cannot interrupt the file transfer unless you cancel the job.

This command is valid when you have established local and foreign connections. In addition, you probably need to be authorized to read the files on the foreign system, and you definitely need to be authorized to write the files on the local VSE system.

The anticipated response depends on the specific foreign FTP daemon. The following responses are from an FTP daemon running on MS Windows during the transfer of two files:

```
MPUT DFHP*.JOB
L: PORT 192,168,000,009,016,254
L: 200 Command okay.
L: NLST DFHP*.JOB
L: 150 File status okay; about to open data connection
L: 226 Closing data connection.
L: PASV
L: 227 Entering Passive Mode (192,168,000,009,016,255).
F: PORT 192,168,000,009,016,255
F: 200 PORT command successful.
F: STOR DFHPCTIP.JOB
F: 150 Opening ASCII mode data connection for DFHPCTIP.JOB.
L: RETR DFHPCTIP.JOB
L: 150-File: PRD2.PAT.DFHPCTIP.JOB
   Type: ASCII  Recfm: FB  Lrecl:   80  Blksize:   80
   CC=ON  UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
   Translate with US_ENG_03
150 File status okay; about to open data connection
F: 226 Transfer complete.
L: 226-Bytes sent:           6,970
   Records sent:           85
   Transfer Seconds:       1.41 (   6K/Sec)
   File I/O Seconds:      .54 (   0K/Sec)
226 Closing data connection.
```

```

L: PASV
L: 227 Entering Passive Mode (192,168,000,009,017,000).
F: PORT 192,168,000,009,017,000
F: 200 PORT command successful.
F: STOR DFHPPTIP.JOB
F: 150 Opening ASCII mode data connection for DFHPPTIP.JOB.
L: RETR DFHPPTIP.JOB
L: 150-File: PRD2.PAT.DFHPPTIP.JOB
    Type: ASCII  Recfm: FB  Lrecl:   80  Blksize:   80
    CC=ON  UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
    Translate with US_ENG_03
150 File status okay; about to open data connection
F: 226 Transfer complete.
L: 226-Bytes sent:          2,378
    Records sent:          29
    Transfer Seconds:      1.21 (    2K/Sec)
    File I/O Seconds:     .22 (    0K/Sec)
226 Closing data connection.
MPUT Requirements completed

```

NLIST, NLST See DIR command.

NOOP, NOP Command Syntax:

```

NOOP
NOP

```

These commands test the connections to the foreign FTP daemon. They do not generate any action.

The NOOP and NOP commands are valid when there is a connection with the foreign host.

The anticipated response to the NOOP command is as follows:

```

NOOP
F: NOOP
F: 200 NOOP command successful.

```

NOPAD The syntax is

```
NOPAD
```

NOPAD has the opposite effect of PAD. It causes leading zeros to be suppressed on the PORT command when establishing data connections. NOPAD is the system default to match the FTP daemon. It is often necessary to suppress padding because some common firewalls are incorrectly coded and deal with leading zeros by corrupting datagrams.

This command is valid when you have established local and foreign connections. The anticipated response to the NOPAD command is as follows:

```
NOPAD  
Ready:
```

NOUPRMPT The syntax is

```
NOUPRMPT
```

This command (No User Prompt) causes FTPBATCH (// EXEC FTPBATCH) to require the LUSER and LPASS prefixes to be used.

OPEN The syntax is

```
OPEN ipaddress portnumber
```

The OPEN command establishes a session with the foreign FTP daemon. For more information about when an explicit OPEN command is required, see the documentation for each individual FTP client.

The variables have the following meanings:

- *ipaddress* is the IP address of the foreign system that you are connecting with. You can specify the IP address in dotted decimal notation or as a symbolic name. The name is resolved by the Domain Name Server or the DEFINE NAME command. There is no default. The *ipaddress* parameter is positional.
- *portnumber* is the port number of the foreign system. The port number is specified as a decimal number. The default is 21, and you should not change it unless you are certain that the foreign host has an FTP daemon listening at some port other than 21. The *portnumber* parameter is positional.

The OPEN command is valid at any time during the FTP session. If a session is already active, it is closed in favor of the new session.

For the interactive client, the anticipated response is a prompt for a foreign user ID and foreign password. For other clients, the anticipated response is that the foreign daemon expects OPEN to be followed immediately by the USER and PASS commands.

A sample anticipated response from a CICS interactive FTP client is as follows:

```
OPEN 192.168.0.7
F: 220-FTPSERVE IBM VM Level at SYS1.YOUR.DOMAIN.NAME, 12:36:28
220 Connection will close if idle for more than 5 minutes.
Enter Foreign User ID or "LOGOFF":
vseuser
F: USER vseuser
F: 331 Send password please.
Enter Foreign Password or "LOGOFF"
xxxxxxx
F: PASS XXXXXXXX
F: 230 VSEUSER logged in; working directory = VSEUSER 191
Foreign host connection established.
Ready:
```

PAD The syntax is

```
PAD
```

This command enables using lead zeros to pad IP addresses and PORT numbers to three-character groupings. It causes leading zeros to be included on the PORT command when establishing data connections. The default behavior is NOPAD to match the FTP daemon.

This command is valid when you have established local and foreign connections.

The anticipated response to the PAD command is as follows:

```
PAD
Ready:
```

PASS The syntax is

```
PASS password
```

The PASS command supplies a password to the foreign FTP daemon.

Variable *password* is a valid password for the user ID specified in the preceding USER command.

This command is valid only when it follows a USER command. It must immediately follow the USER command for all clients except the interactive FTP client. The interactive client responds to the USER command by prompting the CICS terminal user for user ID information, and responds to the user ID information with a password prompt.

The anticipated response is feedback from the foreign FTP daemon indicating whether the foreign user ID and password combination is accepted.

PASSIVE The syntax is

```
PASSIVE
```

This command causes the data connection on VSE to be passive. This means that a PASV command is sent to the VSE FTP daemon, which issues a passive open (listen) on the data connection. This daemon then sends a PORT command to the foreign connection with the IP address and port on which VSE is listening. The foreign FTP daemon then issues an active open to the passively listening VSE FTP daemon. The default behavior is PASSIVE.

This command is valid when you have established local and foreign connections.

The anticipated response to the PASSIVE command is as follows:

```
PASSIVE  
227: ENTERING PASSIVE MODE
```

PBSZ The syntax is

```
PBSZ buffsize
```

The PBSZ command is a foreign protection buffer size request. It is part of RFC2228 FTP Security Extensions.

The argument *buffsize* is a decimal integer representing the maximum size, in bytes, of the encoded data blocks to be sent or received during file transfer. This number shall be no greater than can be represented in a 32-bit unsigned integer.

This command allows the FTP client and server to negotiate a maximum protected buffer size for the connection. There is no default size; the client must issue a PBSZ command before it can issue the first PROT command.

The PBSZ command must be preceded by a successful security data exchange.

The following reply codes should be issued by the server:

- If the server cannot parse the argument, or if the argument does not fit in 32 bits, the server should respond with a 501 reply code.
- If the server has not completed a security data exchange with the client, it should respond with a 503 reply code.

Otherwise, the server must reply with a 200 reply code. If the size provided by the client is too large for the server, it must use a string of the form `PBSZ=number` in the text part of the reply to indicate a smaller buffer size. The client and the server must use the smaller of the two buffer sizes if both buffer sizes are specified.

PROT The syntax is

```
PROT Level
```

The PROT command is a foreign data connection protection-level request. It is part of RFC2228 FTP Security Extensions.

The argument *level* is a single Telnet character code specifying the data-channel protection level. This command indicates to the foreign server what type of data-channel protection the client and server are using. The following codes are assigned:

- C (Clear)
- S (Safe)
- E (Confidential)
- P (Private)

The default protection level, if no other level is specified, is Clear. The Clear protection level indicates that the data channel carries the raw data of the file transfer with no security applied. The Safe protection level indicates that the data is integrity protected. The Confidential protection level indicates that the data is confidentiality protected. The Private protection level indicates that the data is both integrity and confidentiality protected.

It is reasonable for a security mechanism to not provide all data-channel protection levels. It is also reasonable for a mechanism to provide more protection at a level than is required. For example, a mechanism might provide Confidential protection but include integrity protection in that encoding because of API or other considerations.

The PROT command must be preceded by a successful protection buffer size negotiation.

The following table lists the foreign server reply codes for various conditions.

Condition	Reply Code
The server does not understand the specified protection level.	504
The current security mechanism does not support the specified protection level.	536
The server has not completed a protection buffer-size negotiation with the client.	503
No previous PBSZ command was issued, and the PROT command is rejected.	503
The server is not willing to accept the specified protection level.	534
The server is not able to accept the specified protection level, such as if a required resource is unavailable	431

Otherwise, the foreign server must issue a 200 reply code to indicate that the specified protection level is accepted.

PUT, PUTX

The syntax is

```
PUT local-file-name (foreign-file-name)  
PUTX !local-file-name! (!foreign-file-name!)  
STORE local-file-name (foreign-file-name)
```

The PUT, PUTX, and STORE commands retrieve a file from the local FTP daemon on VSE and store it in the foreign FTP file system.

The variables have the following meanings:

- *local-file-name* is the name of a local file that you want to transfer to your foreign system. The name must meet the qualifications outlined in the section [“VSE File Names”](#) (page 2-46). The variable has the same meaning for the PUT and PUTX commands, but PUTX requires the name to be preceded and followed by a delimiting character. This restriction enables you to specify a local file name that contains embedded blanks. The delimiting character cannot be a period. The *local-file-name* does not have to use the same delimiting character that *foreign-file-name* uses. If you do not specify a *foreign-file-name*, the file is stored on the foreign system under its original name.
- *foreign-file-name* is a foreign file name that is dependent on the foreign FTP daemon and the system on which it resides.

Note: The *foreign-file-name* and *local-file-name* variables are specified in the context of the foreign and local current working directories. The file name is appended to the current working directory on both sides of the connection to obtain the real file names. Most (but not all) FTP daemons allow you to specify a fully qualified file name starting with the ‘/’ (or ‘\’ for PC file systems). TCP/IP FOR VSE does not allow you to specify a fully qualified file name. In every case, it appends the *local-file-name* to the current working directory.

This command is valid when you have established local and foreign connections. In addition, you probably need to be authorized to write the file on the foreign system, and you definitely need to be authorized to read the file on the local VSE system.

The following box contains a sample PUTX command. Our example assumes that you have a file in a VSE library named *REXX.PROC*. To store the file in your Microsoft Windows server as *REXX IS A GREAT LANGUAGE.CMD*, you can use the following command:

```
PUTX !REXX.PROC! !REXX IS A GREAT LANGUAGE.CMD!
```

The anticipated response depends on the specific foreign FTP daemon. The following responses are from an MS Windows FTP daemon:

```
PUT wto.a foreign.a
L: PASV
L: 227 Entering Passive Mode (192,168,000,009,016,247).
F: PORT 192,168,000,009,016,247
F: 200 PORT command successful.
F: STOR foreign.a
F: 150 Opening ASCII mode data connection for foreign.a.
L: RETR wto.a
L: 150-File: PRD2.PAT.WTO.A
   Type: ASCII  Recfm: FB  Lrecl:   80  Blksize:   80
   CC=ON  UNIX=OFF  RECLF=OFF  TRCC=OFF  CRLF=ON
   Translate with US_ENG_03
150 File status okay; about to open data connection
F: 226 Transfer complete.
L: 226-Bytes sent:      236,078
   Records sent:      2,879
   Transfer Seconds:   5.36 (   46K/Sec)
   File I/O Seconds:  1.29 (  230K/Sec)
226 Closing data connection.
Ready:
```

PWD The syntax is

```
PWD
```

The PWD command displays the current working directory of the foreign FTP daemon. This command is valid when you have established a connection to the foreign FTP daemon.

The anticipated response is as follows:

```
PWD
F: PWD
F: 257 "/logs" is current directory.
Ready:
```

QUIT See BYE command.

QUOTE The syntax is

```
QUOTE foreign-ftp-server-command
```

The QUOTE command issues a command directly to a foreign FTP daemon. FTP client commands normally are translated to commands that FTP daemons understand. The commands that FTP daemons understand are standard. The QUOTE command bypasses this translation mechanism and allows you to send commands directly to the foreign FTP daemon.

It often is hazardous to issue the QUOTE command instead of commands such as SITE, GET, PUT, and so forth. When you send commands directly to the foreign daemon, you bypass any preliminary command setup that may be required for the commands to function properly.

Variable *foreign-ftp-server-command* is any command that the foreign FTP daemon understands. For a list of commands that are supported by the foreign FTP daemon, you can issue the QUOTE HELP command.

This command is valid when you have established a connection with the foreign host.

The anticipated response is depends on the foreign FTP system. The following response is from IBM's VM/ESA TCP/IP FL310:

```
QUOTE HELP
F: help
F: 214-The server-FTP commands are:
214-ABOR, ACCT,*ALLO, APPE, CWD, DELE, HELP, LIST, MKD, MODE, RMD
214-NLST, NOOP, PASS, PASV, PORT, PWD, QUIT, REIN,*REST, RETR, CDUP
214-RNFR, RNT0, SITE, SYST, STAT, STOR, STOU, STRU, TYPE, USER
214-The commands preceded by '*' are unimplemented
214-The data representation type may be ASCII, EBCDIC or IMAGE. The data
214-structure must be File; the mode may be Stream or Block.
214-If the connection to this server is inactive for more than
214-300 seconds, the connection will be closed.
214-File identifiers have two components: the name and the type.
214-These components are separated by a period.
214-For information about a particular command, type
214 HELP SERVER command.
```

RENAME, RENAMEx

The syntax is

```
RENAME old-file-name new-file-name  
RENAMEx !old-file-name! new-file-name  
RENx !old-file-name! new-file-name
```

The RENAME command asks the foreign FTP daemon to rename a specific file on the foreign system. Some FTP daemons support this command and some do not. In general, the foreign FTP daemon tries to determine whether you are authorized to rename the file. The RENAMEx command is the same as the RENAME command, except that the RENAMEx command uses delimiting characters for the old file name.

The variables have the following meanings:

- *old-file-name* is the name of the file that you want to rename. This name must be meaningful to the foreign file system. The variable has the same meaning for the RENAME and RENAMEx commands, but RENAMEx requires the name to be preceded and followed by a delimiting character. This restriction enables you to specify a foreign file name that contains embedded blanks. The delimiting character cannot be a period.
- *new-file-name* is the name that you want the file to have. This name must be meaningful to the foreign file system.

If the foreign FTP daemon cannot locate the file in the current working directory, the RENAME command is not successful and the foreign daemon issues message number 550. Under normal circumstances, this is enough to terminate the foreign FTP session. Depending on which FTP client you are using, you can tell TCP/IP FOR VSE not to terminate the FTP session for a 550-level message. To do this, use the SET IGNORERR command or the &ERROR variable.

This command is valid when you have established a connection with the foreign host and have the authorization to rename the file.

The anticipated response depends on the specific foreign FTP daemon. The following sample responses are from an MS Windows FTP daemon:

```
RENAME wto.a patwto.a  
F: RNFR wto.a  
F: 350 File exists, ready for destination name  
F: RNT0 patwto.a  
F: 250 RNT0 command successful.  
Ready:
```

RENX See RENAME command.

RETRIEVE See GET command.

RMDIR The syntax is

```
RMDIR directory-name
```

The RMDIR command removes (deletes) the specified directory on the foreign FTP daemon.

Variable *directory-name* is the name of a directory that is meaningful to the foreign FTP file system.

Some FTP daemons do not support the RMDIR command. When VSE is the foreign FTP daemon, it does not support the RMDIR command. Also, VM/ESA does not support the RMDIR command unless you are using the shared file system.

This command is valid when you have established a foreign connection. In addition, you probably need to be authorized to delete the directory on the foreign system.

The anticipated response depends on the specific foreign FTP daemon. The following sample responses are from an MS Windows FTP daemon:

```
RMDIR pat
F: RMD pat
F: 250 RMD command successful.
Ready:
```

SETVAR, SETVAREX The syntax is

```
SETVAR &variablename = value
SETVAREX &variablename = value
```

The SETVAR command sets the value of a variable in the current FTP client environment. Dynamic variables are used primarily to name files when they are stored on the foreign file system. Variable names can be up to 8 bytes long. The first character is an ampersand (&) and is followed by up to seven alphanumeric characters.

A set of predefined variables is available for each FTP client. See the section “[Command Variables](#)” (page [2-20](#)) for a list of these variables.

The command operands are as follows:

- *variablename* is the name of an existing variable
- *value* is the value you want to assign.

The following rules apply when setting variables:

- You can assign a literal value to a variable by enclosing the literal value in quotes. For example, to set variable *&V1* to *ABC*, use the following command:

```
SETVAR &V1 = "ABC"
```

- You can set a variable to any other variable. For example, to set variable *&V2* to the value of variable *&V1*, use the following command. If *&V1* is *ABC*, as shown in the previous example, then this command also sets *&V2* to *ABC*.

```
SETVAR &V2 = &V1
```

- You can set a variable to a substring of another variable, using the SUBSTR command, as shown in the following example. After the command executes, variable *&V3* is set to *BC*.

```
SETVAR &V3 = SUBSTR(&V2,2,2)
```

The syntax of the SUBSTR command is

```
SUBSTR(&varname,decimal_starting_pos,decimal_length)
```

- You can concatenate values in a variable by using the plus sign (+). For example, after the following command executes, variable *&V4* is set to *ABCDEF*.

```
SETVAR &V4 = SUBSTR(&V1,1,1) + &V3 + "DEF"
```

- The maximum length of the resolved value is 49 characters.
- You can use variables anywhere within the script file or a batch job and they are resolved at execution time.
- You can specify a VSE SETPARM variable in commands such as PUT or SUBSTR if you first set another variable to the VSE variable. This requirement applies to automatic FTP and FTPBATCH. For example, assume that the VSE variable *VSEVAR* is defined as follows:

```
// SETPARM VSEVAR='20121113'
```

You could use the following assignments in an FTPBATCH job:

```
SETVAR &TMP = &VSEVAR          /* Set &TMP to VSEVAR
SETVAR &MDY = SUBSTR(&VSEVAR,5,4) /* Now use VSEVAR or TMP
```

The SETVAR command is valid at any time.

If the SETVAR command is successful, there is no response. If the command is invalid or refers to a variable that does not exist, the response is an error message.

SITE The syntax is

```
SITE foreign-ftp-site-command
```

The SITE command issues a site-specific command directly to the foreign FTP daemon. The SITE commands supported by foreign FTP daemons vary. For more information about a specific FTP daemon and its supported SITE commands, see the documentation for that daemon.

Variable *foreign-ftp-site-command* is any SITE command that the foreign FTP daemon understands. For a list of commands that are supported by the foreign FTP daemon, you can usually issue the SITE HELP command.

This command is valid when you have established a connection with the foreign FTP daemon.

The anticipated response varies depending on the command and the foreign FTP daemon.

STATUS The syntax is

```
STATUS
```

The STATUS command displays status information about the foreign FTP daemon.

This command is valid when you have established a connection with the foreign FTP daemon.

The anticipated response depends on the foreign FTP daemon. The MS Windows FTP daemon returns the following response:

```
STATUS
F: STAT
F: 211-csiserver Microsoft Windows FTP Server status:
  Version 3.0
  Connected to 192.168.0.9
  Logged in as pat
  TYPE: ASCII, FORM: Nonprint; STRUCTure: File; transfer MODE: STREAM
  No data connection
211 End of status.
Ready:
```

The VM/ESA FL310 FTP daemon returns the following response:

```
STATUS
F: STAT
F: 211-Server FTP talking to host 192.168.0.9, port 4277
211-User: VSEUSER   Working directory: VSEUSER 191
211-The control connection has transferred 830 bytes.
211-There is no current data connection.
211-The next data connection will be actively opened
211-to host 192.168.0.9, port 4277, using
211-mode Stream, structure File, type ASCII, byte-size 8.
211 record format is V
Ready:
```

STORE See PUT command.

STRUCTURE The syntax is

```
STRUCT [E|R]
STRUCTURE [E|R]
```

TCP/IP FOR VSE can transfer files using one of the following two structures:

- File structure, which transfers the file as a series of bytes.
- Record structure, which transfers the file as a series of records, complete with end-of-record indicators.

The STRUCTURE command enables you to specify whether a file transfer is to use file structure or record structure. Specify *F* for file structure or *R* for record structure. The default is *F*.

All FTP daemons support the file structure mode. The only FTP daemon we know of that supports the record structure mode is in OS/390.

When used in conjunction with the MODE BLOCK command, STRUCTURE R provides FTP with both end-of-record and end-of-file indicators. In general, this provides a more reliable transfer.

This command is valid when you have established local and foreign connections. In addition, the foreign FTP daemon must support the requested structure. The anticipated response is as follows:

```
STRUCT F
F: STRU F
F: 200 STRU F ok.
L: STRU F
L: 200 Command okay.
Ready:
```

SYSTEMS

The syntax is

```
SYST
SYSTEMS
```

The SYSTEMS command displays the system type of the foreign FTP daemon. The FTP standard requires the system type to be the first word following message number 215 in the response.

This command is valid when you have established a connection to the foreign FTP daemon. The anticipated response is as follows. The first word of the response following message number 215 identifies the operating system. The text after the first word is arbitrary.

The following example shows the response from an MS Windows system.

```
SYST
F: SYST
F: 215 _NT version 4.0
Ready:
```

The next example shows the response from a VM/ESA FL310 system.

```
SYST
F: SYST
F: 215-VM/ESA Version 2 Release 3.0, service level 9801
   VM/CMS Level 14, Service Level 801
215 VM is the operating system of this server.
Ready:
```

TERSE The syntax is

```
TERSE
```

The TERSE command causes abbreviated (one-line) no-continuation messages to be issued for the 150 and 226 messages issued during a GET or PUT operation.

TYPE The syntax is

```
TYPE [BINARY|IMAGE|EBCDIC|ASCII]
```

The TYPE command specifies the type of data transfer you want to perform. The command synchronizes the transfer type of the local FTP daemon and the foreign FTP daemon.

The parameters have the following meanings:

- BINARY or IMAGE requests a binary mode transfer and is the same as issuing the BINARY command.
- EBCDIC forces an EBCDIC-mode transfer and is the same as issuing the EBCDIC command.
- ASCII forces an ASCII-mode transfer and is the same as issuing the ASCII command.

This command is valid when you have established local and foreign connections.

The anticipated response is as follows:

```
TYPE A
F: TYPE A N
F: 200 Type set to A.
L: TYPE A N
L: 200 Command okay.
Ready:
```

USER The syntax is

```
USER userid
```

The USER command supplies a user ID to the foreign FTP daemon.

Variable *userid* is a user ID that is valid to the foreign FTP daemon.

This command is valid only when it follows an OPEN command. It must immediately follow the OPEN command for all clients except the interactive FTP client. The interactive client responds to OPEN by prompting the CICS terminal user for user ID information. The interactive client must provide the user ID by responding to the prompt and cannot provide the user ID with the USER command. The interactive client uses the USER command only to change a user ID in an already active local session, and not to supply a user ID in a new session.

You do not need the USER command if you specified the IPADDR parameter in the parameter string of the internal or external batch clients. See the sections for these clients earlier in this chapter for more information about when the USER command is required.

For all FTP clients except the interactive client, the USER command must immediately follow the OPEN command in the input stream.

For the interactive FTP client, the response to the USER command is a prompt for a password by the foreign FTP daemon. The user ID and password are then validated by the foreign FTP.

You do not need to use the USER command if you specified the IPADDR parameter in the parameter string of the INTERNAL or EXTERNAL batch clients. See the description of these clients earlier in this chapter for more information about when the USER command is required.

VERBOSE

The syntax is

```
VERBOSE
```

The VERBOSE command has the opposite effect of the TERSE command. It enables the full text of the 150 and 226 messages to be issued during a GET or PUT operation.

XGOTOEOJ

The syntax is

```
XGOTOEOJ
```

The XGOTOEOJ command sends a GEOJ request to the foreign FTP server.

This functionality is supported only when using the FTPBATCH program as an external FTP server.

FTP as a Server on VSE

An FTP server (daemon) on VSE is defined using one of the following commands:

- DEFINE FTPD (which is a TCP/IP FOR VSE command)
- // EXEC FTPBATCH

An FTP server on VSE must be defined before a foreign FTP client on a PC, UNIX, VM, z/OS, OS/400, or other operating system can transfer files to and from the VSE system.

Note: If you only want to use the external batch FTP client (FTPBATCH) to send or receive files from a remote system, you do not need to configure an FTP server on VSE because the batch program attaches and executes an FTP daemon automatically during the batch job's execution. This daemon is terminated when the FTPBATCH job step is completed.

For details on creating internal and external FTP servers on VSE, see the *TCP/IP FOR VSE Installation Guide*, chapter 6, "Configuring FTP Clients and Daemons."

FTP Server Commands and Replies on VSE

The VSE FTP server includes support for the FTP commands listed later in this section. Normally, the FTP client masks these commands, but most clients allow the use of a QUOTE command to directly pass any of these commands to the VSE FTP server.

For example, a PUT command generates all the following commands:

```

PUT filename
F: PASV
F: 227 Entering Passive Mode (66,193,91,153,210,10)
L: PORT 66,193,91,153,210,10
L: 200 Command okay
L: RETR %SAM0,SAM,FB,80,800
L: 150-About to open data connection
   File:Local.File.Definition
   Type:ASCII Recfm:FB Lrecl:   80 Blksize:   800
   CC=ON UNIX=ON RECLF=OFF TRCC=OFF CRLF=ON NAT=OFF CONT=OFF
   Translate with OS_02
150 File status okay; about to open data connection
F: STOR FTPBSAM0.TXT
F: 150 "/AAAJUNK/FTPBSAM0.TXT" file ready to receive in ASCII mode
L: 226-Bytes sent: 328
   Records sent: 4
   Transfer Seconds:      .33 ( 994-bytes per second)
   File I/O Seconds:     .05 ( 6K per second)
226 Closing data connection
F: 226 Transfer finished successfully.
```

In this example,

- lines beginning with “F:” are associated with commands sent to the foreign FTP server
- lines beginning with “L:” are associated with commands sent to the local FTP server.

Reply Codes

All commands require a three-digit numeric reply to indicate the success or failure of the command. These three-digit codes have specific meanings as defined in IETF FTP Protocol standard (RFC0959).

There are six possible values for first digit of the reply code. These values are described in the following table.

Reply Code	Description
1yz	Positive Preliminary reply. The requested action is being initiated; expect another reply before proceeding with a new command. (The user process sending another command before the completion reply would be in violation of protocol; but server-FTP processes should queue any commands that arrive while a preceding command is in progress.) This type of reply can be used to indicate that the command was accepted and the user process may now pay attention to the data connections, for implementations where simultaneous monitoring is difficult. The server-FTP process may send, at most, one 1yz reply per command.
2yz	Positive Completion reply. The requested action has been successfully completed. A new request may be initiated.
3yz	Positive Intermediate reply. The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The user should send another command specifying this information. This reply is used in command sequence groups.

Reply Code	Description
4yz	<p>Transient Negative Completion reply.</p> <p>The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again. The user should return to the beginning of the command sequence, if any. It is difficult to assign a meaning to “transient,” particularly when two distinct sites (Server and User processes) have to agree on the interpretation. Each reply in the 4yz category might have a slightly different time value, but the intent is that the user process is encouraged to try again. A rule of thumb in determining if a reply fits into the 4yz or the 5yz (Permanent Negative) category is that replies are 4yz if the commands can be repeated without any change in command form or in properties of the User or Server (for example, the command is spelled the same with the same arguments used; the user does not change his file access or user name; the server does not put up a new implementation.)</p>
5yz	<p>Permanent Negative Completion reply.</p> <p>The command was not accepted and the requested action did not take place. The user process is discouraged from repeating the exact request (in the same sequence). Even some “permanent” error conditions can be corrected, so the human user may want to direct his user process to reinitiate the command sequence by direct action at some point in the future (for example, after the spelling has been changed, or the user has altered his directory status.)</p>
6yz	<p>Protected reply.</p> <p>There are three reply codes of this type. The first, reply code 631, indicates an integrity-protected reply. The second, reply code 632, indicates a confidentiality- and integrity-protected reply. The third, reply code 633, indicates a confidentiality-protected reply.</p>

VSE FTP Server Commands

The VSE FTP server commands are summarized in the following table.

Command	Description
ABOR	Aborts an active file transfer
ACCT	Processes account information
ADAT	Authenticates data
ALLO	Allocates storage space
APPE	Appends data to a file being stored
AUTH	Sets authentication mechanism

Command	Description
CDUP	Changes to parent directory
CONF	Processes confidentiality-protected command
CWD	Changes working directory
DELE	Deletes a file
ENC	Processes privacy-protected command
EVNT	Enables event debugging
FEAT	Displays features available
GEOJ	Goes to EOJ
HELP	Gets help for a command
LIST	Lists the contents of a directory
MIC	Processes integrity-protected command
MKD	Creates a directory
MODE	Specifies the mode in which data is to be transferred
NLST	Generates list of file names in a directory
NOOP	Specifies no operation
PASS	Specifies password
PASV	Specifies passive for data connection
PBSZ	Sets protection buffer size
PORT	Specifies a port for data connection
PROT	Sets protection for SSL
PWD	Prints the working directory
QUIT	Terminates the session
REIN	Re-initializes processing
RETR	Retrieves a file
RMD	Removes a directory
RNFR	Renames from
SITE	Specifies a site-specific command; see the section “SITE Commands” (page 2-112)
STAT	Prints status
STOR	Stores a file
STRU	Sets the structure of file to be used
SYST	Displays the type of operating system
TYPE	Specifies representation type: ASCII, EBCDIC, IMAGE
USER	Specifies a user login identifier

VSE Server Command Replies

The FTP daemon issues the messages described in this section to an FTP client. These messages are not displayed on the console. Messages are identified by a three-digit number and are issued only in response to a command sent to the daemon. Message levels have the following meanings:

- Messages at the 100 level indicate partial results and are followed by at least one additional message before another command is accepted.
- Messages at or above the 200 level indicate completion or failure of a command. No further messages are issued until the daemon receives a new command.

A message can be divided into multiple lines if it is large or if it consists of several parts. In this case, the first message line has a hyphen (-) immediately following the message number. Additional lines follow with no leading number. The last line of the message begins with the same message number that started the group, but a blank (instead of a hyphen) follows the number.

Note: The TCP/IP protocols assign specific meanings to the message numbers. The text, however, is usually arbitrary. The following list is accurate for messages issued from the TCP/IP FOR VSE daemon; messages in the list may not precisely reflect messages sent from implementations of other daemons.

The following is a listing of the VSE FTP server command replies.

VSE FTP Server Messages	
120	Service is being setup for new user.
150	File status okay; about to open data connection
150	About to open data connection
200	Command okay
200	Command okay. Operating in Unix simulation mode
202	Command or Parameter not implemented, superfluous
202	Command invalid, include a command to pass
202	Command invalid, a subdirectory structure is not defined
202	Command invalid, special SITE was restricted
202	Command invalid, you are not in the right directory
202	Command invalid, you are in an invalid subdirectory
202	Command invalid, no special SITE commands are available
202	Command invalid, special SITE command failed
211	Supported extensions AUTH TLS PBSZ PROT
212	Currently selected values:
214	----- Help System Display (@1.) -----

VSE FTP Server Messages

214 ----- Help System Display Complete -----
214 Help system unable to locate information requested
215 VSE system type
215 UNIX Simulation on VSE system
220 Ready for new user
221 FTP Daemon closing control connection
225 Data connection open; no transfer in progress
226 Bytes received: @1.
226 Bytes sent: @1.
226 Closing data connection
226 Data connection aborted
227 Entering Passive Mode (@1.)
230 User logged in, proceed
234 TLS/SSL Security negotiation needed
235 Local data connection enabled for SSL
250 Requested file action okay, completed
331 User name okay, need password
350 File exists, ready for destination name
421 Service not available, closing control connection
425 Cannot open data connection
426 Connection closed; transfer aborted
450 Requested file action not taken
451 Requested action aborted: local error in processing
500 Syntax error, command invalid
500 Syntax error, command unrecognized
500 Command rejected, user not logged on
500 Syntax error, command missing
500 Syntax error, command too big
501 Syntax error in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command not implemented for that parameter
504 SSL not initialized
505 Translate value cannot be overridden
505 Translate table not found; value unchanged

VSE FTP Server Messages

507 Initialization failed
521 Data connection cannot be opened with this PROT setting
530 Not logged in
531 User ID already received
545 Request failed reason=@1. RC=@2.
550 Requested action not taken
550 Access denied
550 Action not taken: Security Violation
550 Action not taken: Entry not in directory
550 Action not taken: Entry open failure
550 Action not taken: No directory within Entry
550 Action not taken: Entry not in sub-directory
550 Action not taken: No higher directory
550 Action not taken: Local definition failure
550 Action not taken: OpenDir attempt failed
550 Unable to locate requested directory or file
550 Action not taken: Name too long
551 File close failed
552 Retrieve PDF translation failed
552 Tranfile failed RETR aborted
552 Zero bytes retrieved with ZEROERR=YES
552 Zero bytes stored with ZEROERR=YES
552 UNIX failed
552 NLST failed
553 DBCS file transfer aborted
553 Approximate location: @1. @2. (hex)
554 Received command is too long
556 Save file failed because of wrong length record
557 Save file write failed
558 Data connection failed while receiving file
559 Event request failed

SITE Commands

Sometimes you must explicitly tell TCP/IP FOR VSE's FTP server how to transfer a certain file. You may need to declare a file's logical record length or provide a non-standard translate table. To provide for such needs, most TCP/IP FTP daemons accept SITE commands. These commands are generally unique to the implementation of TCP/IP. TCP/IP FOR VSE provides a comprehensive set of SITE commands that allow you to perform many functions.

Before you issue SITE commands, check with your system administrator. Some of these options can be set on the TCP/IP FOR VSE DEFINE FILE and DEFINE FTPD commands, thereby making it unnecessary to issue SITE commands.

Entering SITE and LSITE Commands

Each FTP client has its own mechanism for entering SITE commands. Command-line FTP clients are pretty straightforward. Graphical clients may require you to do a bit of digging to find out how to enter SITE commands. For example, if you are using WS_FTP, you right-click on the gray background to obtain a special menu. Remember that some clients do not support SITE commands. This is especially true of most Web browsers.

You can use the SITE commands from various FTP clients to communicate with the TCP/IP FOR VSE FTP daemon. If you want to provide those same instructions to TCP/IP FOR VSE while using the TCP/IP FOR VSE FTP client, you cannot use the SITE command because that command would then be processed by a foreign FTP daemon. Instead, you can use the LSITE command. *LSITE* stands for *Local Site*. To use the LSITE command to send instructions to TCP/IP FOR VSE, include it in the internal FTP batch client, the external FTP batch client, or the interactive (CICS) FTP client.

All SITE commands have similar syntax. For example, to assign an LRECL to a file, you might use the following command:

```
SITE LRECL 80
```

In each case, *SITE* or *LSITE* is the first word in the command.

Entering QUOTE and LQUOTE Commands

Some FTP clients do not support the SITE command, and others permit you to enter only a subset of possible commands. Surprisingly, the FTP client that comes with MS Windows, for example, does not contain SITE support. If you need to issue SITE commands from an FTP client that does not contain SITE support, you can substitute the QUOTE command. In our preceding example for logical record length, we showed the format of a SITE command. To substitute the QUOTE command, use the following format:

```
QUOTE SITE LRECL 80
```

The QUOTE command passes the command string (without the *QUOTE*) to the remote FTP daemon. To pass a string to the local FTP daemon, use the LQUOTE command. The client does not examine the command before passing it.

It is important to note that you must be careful when you use QUOTE and LQUOTE, and you must anticipate the effect of the commands you issue. Although you can use QUOTE and LQUOTE to issue all FTP commands, most commands that you issue must be understood and executed by both client and daemon. For example, if you issue a QUOTE EBCDIC command, it would put the daemon into EBCDIC mode but leave the client in ASCII mode. This would be disastrous.

The TCP/IP FOR VSE FTP client command set does include the LSITE command, so you do not need to issue an LQUOTE command while using the FTP client.

Categories of SITE Commands

As we discussed in the previous section, SITE commands allow you to perform a variety of functions. These functions can be divided into four categories:

- SITE commands that provide TCP/IP FOR VSE session control.
- SITE commands that perform system control functions on VSE, such as POWER commands.
- SITE commands that provide explicit instructions about a file, or about how a file is to be transferred or manipulated.
- SITE commands that provide an interface to a specific file type. When programmers develop TCP/IP FOR VSE FILEIO driver routines, they are free to define their own SITE commands. These SITE commands are valid only while the current directory for the FTP session is in that particular type of file. For information about SITE commands in this category, see the documentation for the appropriate product.

In this section, we describe the SITE commands in the first three categories.

Session Control SITE commands that provide TCP/IP FOR VSE session control are summarized in the following table:

SITE Command	Description
REXX	Invokes a REXX program. Your system administrator should provide you with the name of the REXX program and any applicable parameters. For more information about writing a REXX program that is invoked with the SITE REXX command, see the <i>TCP/IP FOR VSE Programmer's Guide</i> .
UNIX	Indicates whether the FTP session is in UNIX-compatibility mode directory structure or in VSE mode.

System Control SITE commands that provide VSE system control functions are as follows.

SITE Command	Description
<i>power command</i>	Any SITE command that is not recognized is passed to POWER (if the user is in the POWER directory) and is issued as an operator command. The response is returned to the FTP client. The system administrator may restrict POWER commands so that they cannot be issued by using a parameter on the DEFINE FILE command.
WTO	Issues a message to the VSE console.

SITE Command Summary

SITE commands for file manipulation and transfer control are summarized in the following table.

SITE Command	Description
ALLOWABORT	Permits an FTP client to abort a transfer in progress when a CTRL-C is issued from a PC DOS window (for example).
BLKSIZE BLOCK BLOCKSIZE	Enables you to supply BLKSIZE information as needed. Note: Other settings may override this command. See " Command Precedence " (page 2-118) for details.
CC	Enables you to control the handling of carriage control characters. Note: Other settings may override this command. See " Command Precedence " (page 2-118) for details.
CLOSE	Closes an open VSAMCAT file.

SITE Command	Description
CRLF	Enables you to restrict the addition of a carriage return/line feed to each downloaded record. Note: Other settings may override this command. See “Command Precedence” (page 2-118) for details.
DIAGNOSE	Enables diagnostics messages for debugging problems.
DUPEKEY	Tells VSE to allow duplicate key conditions to occur while writing to a KSDS file.
EJECT	Suppress or use the first form feed in a report.
EXTTYPES	Enables or disabled using the EXTTYPES overrides.
FCB	Specifies the name of the FCB to be used when CC=ON print-formatted data is transferred.
FCBPREFIX	If an FCB name is passed with a “\$\$\$\$” prefix, then replace it with the four-character string specified by this command.
FIX, FIXRECFM	Enables compatibility with VM’s TCP/IP. It is the same as issuing the following three commands: SITE RECFM FB, SITE LRECL, and SITE BLKSIZE.
FTPMODEL	When writing a new file to a VSAMCAT directory, FTP takes values from the named member as defaults to initially define the new cluster. See override parameters below.
FULLBUFFER	Specifies that a buffer should be filled before it is sent. Default is ON.
HIGHKEY	Forces a buffer to be filled before it is sent.
JSEP	For KSDS files, indicates the ending record when transferring files from VSE. ‘*’ is the last record.
LDEST	Enables you to set the ultimate destination of a POWER spool file when uploading to the POWER LST queue.
LEADZERO	Formats message numbers to send back to the client. Controls whether message numbers have leading zeroes.
LOWKEY	For KSDS files, indicates the starting record to transfer from VSE. ‘*’ is the beginning record.
LRECL	Enables you to supply LRECL information as needed. For files with a RECFM of F or FB, records are padded on the right as required. Note: Other settings may override this command. See “Command Precedence” (page 2-118) for details.

SITE Command	Description
CRLF	Enables you to restrict the addition of a carriage return/line feed to each downloaded record. Note: Other settings may override this command. See “Command Precedence” (page 2-118) for details.
DIAGNOSE	Enables diagnostics messages for debugging problems.
DUPEKEY	Tells VSE to allow duplicate key conditions to occur while writing to a KSDS file.
EJECT	Suppress or use the first form feed in a report.
EXTTYPES	Enables or disabled using the EXTTYPES overrides.
FCB	Specifies the name of the FCB to be used when CC=ON print-formatted data is transferred.
FCBPREFIX	If an FCB name is passed with a “\$\$\$\$” prefix, then replace it with the four-character string specified by this command.
FIX, FIXRECFM	Enables compatibility with VM’s TCP/IP. It is the same as issuing the following three commands: SITE RECFM FB, SITE LRECL, and SITE BLKSIZE.
FTPMODEL	When writing a new file to a VSAMCAT directory, FTP takes values from the named member as defaults to initially define the new cluster. See override parameters below.
FULLBUFFER	Specifies that a buffer should be filled before it is sent. Default is ON.
HIGHKEY	Forces a buffer to be filled before it is sent.
JSEP	For KSDS files, indicates the ending record when transferring files from VSE. ‘*’ is the last record.
LDEST	Enables you to set the ultimate destination of a POWER spool file when uploading to the POWER LST queue.
LEADZERO	Formats message numbers to send back to the client. Controls whether message numbers have leading zeroes.
LOWKEY	For KSDS files, indicates the starting record to transfer from VSE. ‘*’ is the beginning record.
LRECL	Enables you to supply LRECL information as needed. For files with a RECFM of F or FB, records are padded on the right as required. Note: Other settings may override this command. See “Command Precedence” (page 2-118) for details.

SITE Command	Description
MODELNAME	Sets the variable "&MODELNAME" that an FTPMODEL member can use. The specified member is used by DEFINE CLUSTER in VSAMCAT when generating a new VSAM entry.
NAT	Enables the NAT interface. <i>x</i> .
NOBLANK	Synonym for STRIP
NOEJECT	Suppresses the printing of a blank page at the beginning of a report.
NULLRECORD	Specifies how zero-length records should be handled.
PADDING	For fixed-length records, pad short records to fill up to their full length. The pad is a blank for text and a X'00' for binary.
PASSWD	Enables you to specify a VSAM password before you access password protected VSAM clusters.
PDEST	Enables you to set the ultimate destination of a POWER spool file when uploading to the POWER PUN queue.
PDF	When issuing a GET from VSE, it tells the FTP daemon to convert the text to PDF format before transferring it to your location. You can use the default PDF configuration file <i>PDFSETUP</i> by setting the value to "ON," or you can specify the name of a custom configuration file you have cataloged.
PROCDATA	Allows you to set the "DATA=" flag when cataloging .PROC members within the VSE library.
RECCR	Formats the record to conform to sites that require records to end with a carriage return and no line feed.
RECORD_CONTINUE	If the last byte of a fixed-format record (RECFM=F) is an ampersand (&), concatenate the next record.
RECFM	Enables you to supply RECFM information as needed. Note: Other settings may override this command. See " Command Precedence " (page 2-118) for details.
RECLF	Enables you to control the definition of a record delimiter while uploading files to VSE.
SOSI	Controls the mechanism by which TCP/IP uses double-byte character sets.
STRIP	Strips the right-most blanks from records being transferred from the VSE system.
TERSE	Reduces the number of messages output from an FTP transfer.

SITE Command	Description
TRANSLATE	Enables you to supply a custom translate table. Standard translate tables are provided for conversions between ASCII and EBCDIC. If these tables do not fit your needs, you can create and provide your own. For more information, see the <i>TCP/IP FOR VSE Installation Guide</i> . To specify your table by name, use the SITE command.
TRCC	Simulates the ANSI carriage control codes of +, 0, -, and 1. Note: Other settings may override this command. See “Command Precedence” (page 2-118) for details.
XDEST	Enables you to set the ultimate destination of a POWER spool file when uploading to the POWER RDR queue.
The following SITE commands are used for overriding the FTPMODEL values when defining a new cluster in the VSAMCAT file area	
CAFREE	Free space for the control area
CASIZE	Control area size
CIFREE	Free space for the control interval
CISIZE	Control interval size
KEYLEN	Length of the key (for KSDS only)
KEYLOC	Beginning offset of the key (for KSDS only)
PALLOC	Primary allocation value
SALLOC	Secondary allocation value
SHARE	Share options
VOLID	Default volume identifier

Command Precedence

As described earlier, definitions in the EXTYPES.L member normally control transfer settings. These settings include the BLKSIZE, LRECL and RECFM parameter values. It is important to understand how these values may be controlled when you want to transfer a file.

By default, if any part of the file name matches a file type string in EXTYPES.L, default values assigned for that file type (EXTYPES overrides) are used. These values override all other settings.

The system administrator manages the EXTYPES.L member. For more information on the file type definitions in EXTYPES.L, see the *TCP/IP FOR VSE Installation Guide*.

If EXTTYPES processing is disabled (you use SET EXTTYPES OFF, for example), or if the file type is not found in the EXTTYPES.L member, then settings are determined as follows:

- If SET SITELAST NO is specified (the default), then parameter settings on the file's DEFINE FILE statement are used and take precedence over SITE commands.

The system administrator should know which parameters have been set on the file to be transferred. SITE command settings are used otherwise.

- If SET SITELAST YES is specified, then the following SITE commands take precedence over parameter settings on the file's DEFINE FILE statement:

```
SITE BLKSIZE
SITE CC
SITE CRLF
SITE LRECL
SITE RECFM
SITE TRCC
```

SITE Command Details

This section provides more detailed information on most SITE commands, including syntax, variable definitions, and appropriate usage.

SITE ALLOWABORT

The syntax is

```
SITE ALLOWABORT [ON|OFF]
```

This command permits the client to abort a transfer in progress, such as by issuing a CTRL-C from a PC DOS window. The default is ON.

SITE BLKSIZE, SITE LRECL, SITE RECFM

The syntax is

```
SITE BLKSIZE blksize
LSITE BLKSIZE blksize
SITE LRECL lrecl
LSITE LRECL lrecl
SITE RECFM recfm
LSITE RECFM recfm
```

These commands enable you to supply BLKSIZE information as needed.

The variables have the following meanings:

- *blksize* is the block size used by the file
- *lrecl* is the logical record length of the records in the file
- *recfm* is the record format of the file.

Acceptable values for *blksize*, *lrecl*, and *recfm* depend on how you originally defined the file, the access method you use, and the mode of access (input or output). When the acceptable values are dependent on the mode of access, you need to specify which mode you are defining. If you specify one mode and then you need to access the file in the other mode, your administrator can issue a second DEFINE FILE command to give the file a second name with characteristics of the other mode. To supply values that are not accepted by the DEFINE FILE command, use SITE commands.

Note: The SET SITELAST YES command allows these SITE/LSITE commands to override parameter settings on the file's DEFINE FILE statement. See the section "[Command Precedence](#)" (page 2-118) for details.

The following tables show acceptable values for *blksize*, *lrecl*, and *recfm*. *Input* indicates that you are reading from disk, and *Output* indicates that you are writing to disk. These terms do not indicate whether you are using the FTP client or the FTP daemon.

For **sequential disk files and VSAM-managed SAM files**, note the following information:

- Fixed-length records are padded when necessary. When padding occurs, text files are padded with blanks and binary files are padded with zeros.
- To eliminate the need for SITE commands, your VSE administrator can define the same physical file with two different public names (for input and output) and assign different LRECL and BLKSIZE values to each.
- You can process VSAM-managed SAM files as VSAM ESDS files, and it is more efficient to do so because the SAM emulation routines call VSAM anyway.
- The following table shows appropriate values for sequential disk files and VSAM-managed SAM files:

<i>recfm</i>	Input		Output	
	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A	record size plus 8	N/A
FB	record size	record size times blocking factor	record size	(record size times blocking factor) plus 8
V	maximum record size	N/A	maximum record size plus 8	N/A
VB	maximum record size	maximum block size	maximum record size	maximum block size plus 8

	Input		Output	
<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
SU	maximum record size	N/A	maximum record size plus 8	N/A
SB	maximum record size	maximum block size	maximum record size	maximum block size plus 8

For **POWER files**, note the following information:

- Fixed-length records are padded if necessary when writing to the POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with zeros.
- The minimum *lrecl* for POWER RDR queue files is 80 and the maximum is 128.
- The minimum *lrecl* for POWER LST queue files is 1 and the maximum is 32766.
- The *lrecl* for POWER PUN queue files must be 80.
- The following table shows appropriate values:

	Input		Output	
<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A	N/A	N/A
V	maximum record size	N/A	N/A	N/A

For **VSAM files**, note the following information:

- Fixed-length records are padded if necessary when writing to the POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with zeros.
- The following table shows appropriate values for VSAM files:

	Input		Output	
<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A	N/A	N/A
V	max record size	N/A	N/A	N/A

For **ICCF files**, note the following information:

- The files are read only.
- The files always contain 80-byte records, regardless of specification.
- The following table shows appropriate values for ICCF files:

	Input		Output	
<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
F	N/A	N/A	N/A	N/A

For **Library files**, note the following information:

- Library members always contain fixed 80-byte records or a string file consisting of a single string of bytes.
- Library format *SV* is a special form of string file defined by CSI International. It is used to upload HTML members to VSE libraries.
- FTP of phases is not supported.
- The following table shows appropriate values:

	Input		Output	
<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>	<i>lrecl</i>	<i>blksize</i>
F	80	N/A	80	N/A
SV	N/A	N/A	N/A	N/A
S	N/A	N/A	N/A	N/A

SITE CC The syntax is

```
SITE CC [ON|OFF|RAW]
LSITE CC [ON|OFF|RAW]
```

This command tells TCP/IP FOR VSE how to handle carriage control (CC) characters. It is used primarily for POWER spool files, but it applies to all files. This command works in conjunction with SITE TRCC. The default is *ON*.

The SITE CC command is not necessary if you are using a file that is defined to use a specific value for CC. Your system administrator should know if this situation exists.

Note: The SET SITELAST YES command allows this SITE command to override the CC= parameter setting on the file's DEFINE FILE statement. See the section "[Command Precedence](#)" (page [2-118](#)) for details.

You can always determine the CC setting that was used for the transfer by examining the FTP output.

Use the following table to determine the effect of the SITE CC command:

Value	When sending data from VSE	When sending data to VSE
OFF	The first byte of each record is removed.	A blank character is prefixed to each record as it is written.
ON	The data is transmitted normally.	The data is transmitted normally.
RAW	The data is transmitted normally, except for POWER LST queue members where the first byte of the record is not translated from EBCDIC to ASCII. (The remainder is translated.) The first byte is assumed to be a machine carriage control character. Use <i>RAW</i> only if the member contains a machine carriage control (CC) character. <i>RAW</i> is used for advanced function printing (AFP) data streams.	The data is transmitted normally.

SITE CRLF The syntax is

```
SITE CRLF [ON|OFF]
LSITE CRLF [ON|OFF]
```

These commands are valid if you meet the following restrictions:

- You are in ASCII mode
- You are transmitting data from VSE.

The default is *ON*.

This command tells TCP/IP FOR VSE to add a carriage return/line feed combination to all records transmitted from VSE to an ASCII platform. This addition is required because ASCII platforms do not know what a record looks like, so record-oriented files are simulated by adding the CR/LF combination. Setting CRLF to *OFF* inhibits the addition of a CR/LF.

The SITE CRLF command is not necessary if you are using a file that is defined to use a specific value for CRLF. Your system administrator should know if this situation exists.

Note: The SET SITELAST YES command allows this SITE command to override the CRLF= parameter setting on the DEFINE FILE statement. See the section “[Command Precedence](#)” (page [2-118](#)) for details.

You can always find the CRLF setting that was used for the transfer by examining the FTP output.

SITE CLOSE The syntax is

```
SITE CLOSE file_id
```

This command tells TCP/IP FOR VSE to close the open VSAMCAT file specified by *file_id*.

The SITE CLOSE command can only be used for files that are defined using the DEFINE FILE statement with TYPE=VSAMCAT.

Before executing this command, the FTP client must be in the subdirectory where the file exists, as shown by the following generic command sequence:

1. CD / LCD VSAMCAT_ *public_name*
 2. (QUOTE) LSITE / SITE CLOSE *filename*.
-

SITE DIAGNOSE

The syntax is

```
SITE DIAGNOSE [ON|OFF]
```

This command enables extra diagnostic messages for debugging problems. Default is OFF.

SITE DUPKEY

The syntax is

```
SITE DUPKEY [ON|OFF]
```

This option tells VSE to allow duplicate key conditions to occur while writing to a KSDS file. Default is OFF.

SITE EJECT

The syntax is

```
SITE EJECT [ON|OFF]
```

By default, the first form feed in a report is used and prints a blank page (OFF). (This is the same as SITE NOEJECT ON.) When SITE EJECT is set to ON, it suppresses the first form feed and prevents printing a bank page.

SITE EXTYPES

The syntax is

```
SITE EXTYPES [ON|OFF]
```

This command enables (ON) or disables (OFF) using the EXTYPES.L table to control EBCDIC-to-ASCII translation during a file transfer. Default is ON. A security exit can disallow this command.

SITE FCB

The syntax is

```
SITE FCB name
```

This command specifies the name of the FCB to use when transferring CC=ON print-formatted data.

SITE FCBPREFIX

The syntax is

```
SITE FCBPREFIX string
```

This command specifies an FCB prefix string. If an FCB name is passed with a “\$\$\$\$” prefix, the prefix is replaced with this four-character string.

**SITE FIX,
SITE FIXRECFM**

The syntax is

```
SITE FIX nnn
SITE FIXRECFM nnn
```

This command enables compatibility with VM's TCP/IP. Using the SITE FIX command is the same as issuing the following three commands:

- SITE RECFM FB
- SITE LRECL *nnn*
- SITE BLKSIZE *nnn*

The variable *nnn* is the block size used by the file.

SITE FTPMODEL

The syntax is

```
SITE FTPMODEL [membername|FTPMODEL.L]
```

This command defines defaults used when writing new files to a VSAMCAT directory. FTP uses these values to initially define the new cluster.

The default for *membername* is "FTPMODEL."

The following SITE commands may be used to override the FTPMODEL values when defining a new cluster in the VSAMCAT file area.

SITE Command	Description
PALLOC	Primary allocation value
SALLOC	Secondary allocation value
VOLID	Default volume identifier
SHARE	Share options
CISIZE	Control interval size
CASIZE	Control area size
KEYLOC	Beginning offset of the key; for KSDS only
KEYLEN	Length of the key; for KSDS only
CAFREE	Free space for the control area
CIFREE	Free space for the control interval

SITE FULLBUFFER The syntax is

```
SITE FULLBUFFER [ON|OFF]
```

This command specifies that a buffer should be filled before it is sent. Default is ON.

SITE HIGHKEY See SITE LOWKEY

SITE JSEP The syntax is

```
SITE JSEP nnn
```

This command allows you to specify the JSEP (separator page count) value *nnn* when writing data to the POWER print queue.

The SEPARATOR_PAGES and SET PAGE_COUNT values are established as defaults for the POWER driven entry. The SEPARATOR_PAGES command either enables or disables separator pages for all FTP operations. These defaults can be overridden with the SITE JSEP command. If data being read from the POWER queue contains separator pages, based on how the data was created, and if JSEP is enabled, then those pages should be returned. If no JSEP pages were created for the entry, then none are returned.

SITE JSEP takes effect only when writing a report *into* the POWER print queue.

For more information about the SEPARATOR_PAGES and the SET PAGE_COUNT commands, see the *TCP/IP FOR VSE Command Reference*.

**SITE LDEST,
SITE PDEST,
SITE XDEST** The syntax is

```
SITE LDEST=[nodeid|(nodeid,userid)]
LSITE LDEST=[nodeid|(nodeid,userid)]
SITE PDEST=[nodeid|(nodeid,userid)]
LSITE PDEST=[nodeid|(nodeid,userid)]
SITE XDEST=[nodeid|(nodeid,userid)]
LSITE XDEST=[nodeid|(nodeid,userid)]
```

These commands are valid when you meet the following conditions:

- The current directory points to a file defined with *TYPE=POWER*.
- You are sending data to a POWER queue.

Note: The system administrator may have disabled the used of this facility by using the ALLOWSITE parameter on the DEFINE FILE statement when the POWER queue was defined.

POWER normally processes uploaded files immediately. For example, when you upload a file to the POWER RDR queue, it may begin execution; when you upload a file to the POWER LST queue, it may begin printing. These commands provide you with some flexibility when you upload jobs because they enable you to specify how POWER is to handle the jobs. You should use the commands for the following purposes:

- Use SITE LDEST to send files to the POWER LST queue.
- Use SITE PDEST to send files to the POWER PUN queue.
- Use SITE XDEST to send files to the POWER RDR queue.

The variables have the following meanings:

- When you use SITE LDEST, *nodeid* is the NJE node that is to receive the list output. If you do not specify *userid*, the central printer at the specified node processes the output. You can specify an * for LDEST to indicate that the output is to be processed at the node where you place the file.
- When you use SITE PDEST, *nodeid* is the NJE node that is to receive the punch output. If you do not specify *userid*, the specified node processes the output. You can specify an * for PDEST to indicate that the output is to be processed at the node where you place the file.
- When you use SITE XDEST, *nodeid* is the NJE node that is to run the job. If you specify *nodeid* and *userid*, *nodeid* is a VSE system running under control of the same VM system as the system you are uploading the file to.

If you do not specify a value for *nodeid* or *nodeid,userid*, the current values are returned.

The queue destinations have a common syntax. It is the same syntax that is used on the DEST= parameter in POWER JECL. For more information about the syntax of the LDEST, XDEST, and PDEST commands, see IBM manual *VSE/Enterprise Systems Architecture VSE Central Functions: VSE/POWER Administration and Operation* (SC33-6633). The Book Manager file is *D:\BOOKS\IESPAE01.BOO*.

SITE LEADZERO

The syntax is

```
SITE LEADZERO [ON|OFF]
```

This command formats the message numbers that are sent back to the client. If set to ON, leading zeroes are added to the message numbers. Default is OFF.

**SITE LOWKEY,
SITE HIGHKEY**

The syntax is

```
SITE LOWKEY lowkey
SITE HIGHKEY highkey
```

The LOWKEY and HIGHKEY commands allow you to obtain a specific key or range of keys. If LOWKEY is not set, the next command starts reading with the first record. If HIGHKEY is not set, the command reads to the last record. If the key is too long, it is truncated. If the key is too short, it is treated as a generic key. If the key ends with an asterisk, it also is treated as generic.

For example, if SITE HIGHKEY 55* is specified, the command stops reading the file when the first two bytes in the record key exceed 55.

SITE MODELNAME

The syntax is

```
SITE MODELNAME member_name
```

This command sets the variable “&MODELNAME” that an FTPMODEL member can use. The specified member defines characteristics that are used by DEFINE CLUSTER in VSAMCAT when generating a new VSAM entry.

For example, the entry “MODELNAME(&MODELNAME)” would be added to the FTPMODEL member as in the following example. If MODELNAME is not in an FTPMODEL member, then it is not used. See SITE FTPMODEL for related information.

```
DEFINE CLUSTER (NAME -
  (&NAME) -
  MODELNAME(&MODELNAME) -
  VOLUMES(&VOLID)) -
  DATA (NAME -
  (&NAME.D)) -
  CATALOG -
  (&CATALOG)
```

SITE NAT

The syntax is

```
SITE NAT [OFF|ON]
```

The SITE NAT command enables the NAT interface.x.

The default is OFF.

SITE NOBLANK Synonym for SITE STRIP

SITE NOEJECT The syntax is

```
SITE NOEJECT [ON|OFF]
```

When set to OFF, this command suppresses the printing of a blank page (skips the form feed) at the beginning of a report. The default ON allows a blank page to be printed if a form feed (FF) is present.

SITE NULLRECORD The syntax is

```
SITE NULLRECORD [NULL|BLANK|ZERO|NOTHING]
```

This command specifies how zero-length records should be handled.

If used, the parameter must be one of the following values:

Value	Meaning
NULL	Don't do anything special; leave the incoming "data" as is.
BLANK	For a text file, treat it as though it were a record containing only blanks. Use this setting for <i>text files only</i> .
ZERO	For a non-text file, treat it as though it were a record full of X'00'. Use this setting for <i>binary files only</i> .
NOTHING	Discard the zero-length "record."

SITE PADDING The syntax is

```
SITE PADDING [ON|OFF]
```

This command controls record padding for fixed-format records (RECFM=F). It pads short records to fill them to their full length. Blanks are used for text padding; X'00' is used for binary padding.

Default is OFF.

SITE PASSWD

The syntax is

```
SITE PASSWD password
LSITE PASSWD password
```

These commands are meaningful only when you are about to invoke an operation against a password-protected VSAM cluster.

The SITE PASSWD command allows you to specify the password for a VSAM CLUSTER before you access the cluster with a read or write command. For more information about using passwords for accessing VSAM catalogs and VSAM clusters, see the *VSE/VSAM Programmer's Reference* (IBM Manual: SC24-5145).

SITE PDF

The syntax is

```
SITE PDF [OFF|ON|member]
```

When you issue a GET from VSE, SITE PDF tells the FTP daemon to convert the text to PDF before transferring it to your location. You can use the default PDF configuration file *PDFSETUP.L* by setting this command to ON, or you can specify the name of a custom configuration member (omitting the “.L”) you have cataloged.

See [Chapter 8, “PDF Conversion Facility.”](#) for details on creating a custom PDF configuration member.

Default is OFF.

**SITE
power-command**

The syntax is

```
SITE power-command
LSITE power-command
```

These commands are valid when the current directory points to a public name that is defined with *TYPE=POWER* to represent the POWER directory structure.

Note: The system administrator may have disabled use of this facility by using the ALLOWSITE parameter on the DEFINE FILE statement during POWER queue definition.

There is no default.

This command issues the POWER command specified in variable *power-command*. The results are returned to the FTP client. You cannot issue the *PEND* command or the *D Q* command using this facility.

Important: If the current directory points to *POWER* and you issue a *SITE* command, that *SITE* command is passed to *POWER* for execution as a *POWER* command.

When you issue this command, you receive one of the following responses:

Response	Description
200 <i>command-response</i> 200 Command OK	The command is accepted and <i>POWER</i> has executed it. The <i>command-response</i> is returned to the FTP client. The Command OK message ends the response.
501 Syntax Error in Command or Arguments	The <i>SITE</i> command was passed to <i>POWER</i> , but <i>POWER</i> rejected it. Either the command syntax is invalid, or TCP/IP FOR VSE did not pass the command to <i>POWER</i> for another reason. The command could have been rejected because it would not have been executed.

SITE PROCDATA

The syntax is

```
SITE PROCDATA [YES|NO]
LSITE PROCDATA [YES|NO]
```

These commands are valid only when you are uploading to a library member. The library member must have an extension of *.PROC*.

The default is *NO*.

Setting *PROCDATA* to *YES* has the same effect as specifying *DATA=YES* on the VSE Librarian *CATALOG* command. It enables you to use in-line *SYSIPT* data as part of the member. For information about when you must specify *YES*, see IBM manual *VSE/Enterprise Systems Architecture: Guide to System Functions*. The information is also contained on the BookManager CD-ROM in *BOOKS\IESSFE01.BOO*.

SITE RECCR

The syntax is

```
SITE RECCR [ON|OFF]
```

This command formats the record to conform to sites on the receiving end that require a carriage return (CR) only (no line feed) as the record terminator. Default is *OFF*.

SITE RECLF

The syntax is

```
SITE RECLF [ON|OFF]
LSITE RECLF [ON|OFF]
```

These commands are valid if you meet the following restrictions:

- You are in ASCII mode
- You are transmitting data to VSE.

The default is *OFF*.

TCP/IP FOR VSE normally determines an end-of-record condition for an ASCII file by the existence of a carriage return/line feed combination. If RECLF is set to *ON*, TCP/IP FOR VSE determines an end-of-record condition in an ASCII file by the existence of a line feed.

You can determine the RECLF setting that was used for the transfer by examining the FTP output.

**SITE
RECORD_CONTINUE**

The syntax is

```
SITE RECORD_CONTINUE [ON|OFF]
```

This command controls concatenation for fixed-format records. If the last byte in the record being processed is an ampersand (&), the next record is concatenated with that record. Default is *OFF*.

SITE SOSI

The syntax is

```
SITE SOSI [CONVERT|KEEP|XLATE|BLANK|NONE]
LSITE SOSI [CONVERT|KEEP|XLATE|BLANK|NONE]
```

These commands apply when you are transmitting DBCS files.

The default is *CONVERT*.

This command supports installations that use Double Byte Character Set (DBCS) formats. For more information about using DBCS with TCP/IP FOR VSE, see the *TCP/IP FOR VSE Installation Guide*, chapter 11, “ASCII-to-EBCDIC Translation.”

EBCDIC data streams always use SO/SI characters to bracket DBCS character strings. There is no restriction on the value of the first byte of a DBCS character, except that it cannot be SI or SO, CR, LF, or NL. Other values may also be reserved, such as X'00' and X'FF'.

ASCII data streams never use SO/SI characters but rely instead on the value of the first byte of the DBCS pair. Therefore, there must be a place in the SBCS code page to accommodate each range of 256 DBCS characters.

During standard translation, SO/SI characters must be added or removed as appropriate. An optional protocol permits the SO/SI characters to be retained as placeholders in the ASCII data stream. When translated in this manner, the EBCDIC and ASCII data streams have the same length (except for the ASCII end-of-record delimiters). Note that these placeholders are not used when translating back to EBCDIC. Instead, we rely on the first-byte values. The appropriate placeholder must be present, however, or an invalid data stream condition exists and the transfer stops. SO/SI handling is controlled by the SITE command.

When transferring a DBCS data file from a PC to VSE, the destination file must be able to contain records larger than the specified LRECL for the file. This is because SO/SI characters are added for each record containing DBCS characters. When CRLF=OFF, the ASCII byte stream is split into records using the specified LRECL value. The data is then translated to EBCDIC using SITE SOSI CONVERT, which inserts 2 bytes for each DBCS string included in the record. Therefore, the destination file on VSE must be defined with a record length of LRECL + 2 or, if multiple (n) DBCS strings are included in SBCS, with a record length of LRECL + ($n \times 2$).

The following table shows the specific SITE commands and their meanings:

Command	Description
SITE SOSI CONVERT	This default indicates that the data stream is converted and that SO/SI characters are added or removed as appropriate.
SITE SOSI KEEP	SO/SI characters are retained as placeholders in the ASCII data stream.
SITE SOSI XLATE	SO/SI characters are retained as placeholders in the ASCII stream but are translated to their ASCII equivalents of X'1E' and X'1F'.
SITE SOSI BLANK	SO/SI characters are retained as placeholders in the ASCII stream but are translated to ASCII spaces of X'20'.
SITE SOSI NONE	SO/SI characters are not present in the EBCDIC data. Every pair of bytes is treated as DBCS. This mode is referred to as the graphic character set.

SITE STRIP

The syntax is

```
SITE STRIP [ON|OFF]
```

When enabled, the SITE STRIP command strips the right-most (trailing) blanks from records for non-binary transfers. This effectively allows the CR or LF to indicate the end of the record. This action can enhance performance by reducing the unnecessary transfer of blanks on the end of each record.

A FTP941 message displays the total number of bytes stripped off when using this site option. This only applies to data being sent from VSE to a foreign FTP server, and it is independent of the file I/O driver being used.

SITE TERSE

The syntax is

```
SITE TERSE [ON|OFF]
```

This command reduces the number of messages output from an FTP transfer. Default is OFF.

SITE TRANSLATE

The syntax is

```
SITE TRANSLATE translate-table  
LSITE TRANSLATE translate-table
```

These commands are effective for transfers with a type of *ASCII*.

The default for *translate-table* is the System Translation Table Default Name that is set with the TCP/IP FOR VSE DEFINE TRANSLATION command.

This command enables you to specify the name of the translate table that is to be used to perform ASCII-to-EBCDIC or EBCDIC-to-ASCII translation. Before you can specify a specific *translate-table*, you must verify that your system administrator has defined the table.

The SITE TRANSLATE command is not required in the following situations:

- You are using a file that is defined to use a specific translate table.
- You are using an FTP daemon that is defined to use a specific translate table.

Your system administrator should know if these situations exist.

For FTP file transmissions, the translate table that is used depends on the table's precedence, which is as follows. The table with the highest precedence is listed first.

1. The TRANSLATE= parameter setting on the DEFINE FILE command that was used to define the file being transferred.
2. The TRANSLATE= parameter setting on the DEFINE FTPD command that was used to start the current FTP daemon.
3. The translate table named in the SITE TRANSLATE command.
4. The translate table defined as the system default table. This is the table that is named in the DEFINE TRANSLATION command.

This means that if you specify the TRANSLATE= parameter on either the DEFINE FILE command or the DEFINE FTPD command, you cannot override it with the SITE TRANSLATE command. Also, using the SET SITELAST YES command does not change this precedence.

You can always determine which translate table was used in the transfer by examining the FTP output.

SITE TRCC

The syntax is

```
SITE TRCC [ON|OFF]
LSITE TRCC [ON|OFF]
```

These commands are valid at any time if you meet the following conditions:

- The CC command is set to *OFF*, indicating that the first byte of each record is to be removed.
- You are translating files from EBCDIC to ASCII.
- You are transferring files to another platform. SITE TRCC is ignored if you are transferring data to VSE.

The default is *OFF*.

This command tells TCP/IP FOR VSE if it should simulate the ANSI carriage control operation. This command works in conjunction with SITE CC. If TRCC is set to *ON*, TCP/IP FOR VSE substitutes the correct number of line feeds and top-of-form characters to simulate the effect of the CC character (which is why the CC character itself must be removed).

The SITE TRCC command is not necessary if you are using a file that is defined to use a specific value for TRCC. Your system administrator should know if this situation exists.

Note: The SET SITELAST YES command allows this SITE command to override the TRCC= parameter setting on the DEFINE FILE statement. See “[Command Precedence](#)” (page 2-118) for details.

You can always determine the TRCC setting that was used for the transfer by examining the FTP output. Specifically, when TRCC is set to *ON*, the following operations occur:

ANSI Carriage Control Code	Mechanism Used for Simulation
1 (form feed)	Substitute X'0C0D', which is the ASCII form feed/carriage return combination
0 (skip two spaces)	Substitute X'0D0A0D0A', which is two ASCII CR/LF combinations
+ (suppress line feed)	Substitute X'0D', which is the ASCII CR combination
(blank) (normal CR/LF)	Substitute X'0D0A', which is the ASCII CR/LF character

SITE UNIX

The syntax is

```
SITE UNIX [ON|OFF]
LSITE UNIX [ON|OFF]
```

These commands are valid at any time.

The default is *OFF*.

This command tells TCP/IP FOR VSE whether to place an FTP session in UNIX mode or take it out of UNIX mode. See the discussion of UNIX mode earlier in this chapter for more information about the effects of issuing this command.

You can always determine the UNIX setting that was used for the transfer by examining the FTP output.

SITE WTO The syntax is

```
SITE WTO operator-message  
LSITE WTO operator-message
```

These commands are valid when the current directory points to a public name that is defined with *TYPE=POWER* to represent the POWER directory structure.

There is no default.

This command sends an operator message to the VSE console. The text of the message is contained in variable *operator-message*. This capability is useful if you want to send a message to a VSE operator or to a VSE-based automation product before or after you transfer a file.

Telnet and TN3270

Overview

TCP/IP FOR VSE includes a telnet daemon that supports the TN3270 protocol. This daemon enables you to log on to VTAM applications directly from your workstation or from another client that is connected to TCP/IP. The mechanics of logging on to a VSE application are almost completely dependent on the nature of your TN3270 client. Fortunately, most TN3270 clients are similar. Two clients that work are:

- TN3270 Plus. This client is a shareware product and is available for download at www.sdisw.com.
- OWS3270. This client is also a shareware product and is available for download at www.jollygiant.com.

Before you use TN3270 to connect to VSE, you need the following information:

- The IP address of the VSE system you plan to connect to.

Simply put, someone has to tell you what this address is. When TCP/IP FOR VSE is initialized, this value is specified by the SET IPADDR initialization parameter. If you have access to the VSE console, you can use the QUERY SET command to display the assigned value.

- The port number you plan to connect to.

Usually you specify port number 23, which is the standard port for TN3270 connections. TCP/IP FOR VSE allows your VSE administrator to assign other ports for specific purposes. Your VSE administrator can tell you which port number to use.

- The best type of terminal for your TN3270 client to emulate.

Some TN3270 clients have more functionality than others. TCP/IP FOR VSE is designed to work with most TN3270 clients. It supports most TN3270 advanced functions, including 7 Color Support and Extended Data Streams. It supports 327x models 2, 3, 4, and 5.

Note About Color

On some TN3270 clients, the colors may be different from the colors on real 3270 terminals. Most TN3270 clients allow you to customize the colors. Look for the customization dialogue box, which should let you set the colors to match the ones on real 3270 screens.

Interacting with a TN3270 Client

TN3270 clients have many of the same problems as other 3270 emulators. The biggest hurdle to overcome is keyboard mapping. From the server perspective, we cannot do much to help you, except to point out that most TN3270 clients have keyboard mapping utilities. You should be aware that many TN3270 clients are capable of multiple sessions, as shown in the example below. Some advanced clients are able to record sessions and play them back. A few TN3270 clients allow you to automate interactions through a scripting language.

**Caution About
Closing a Telnet
Session**

If you use the CICS Interactive Interface, it is very important to log off CICS before you close your telnet session or reboot your PC. If you do not log off, your user ID may remain logged in to the Interactive Interface.

Using the CICS Telnet Client

The telnet client that is provided with TCP/IP FOR VSE operates in either line mode or full-screen mode under a CICS session. It is used to initiate a terminal session with another TCP/IP-connected host in 3270 mode or in line mode. In this context, the TCP/IP FOR VSE telnet client is the *client* and the other host is the *server*. The TCP/IP FOR VSE telnet client always attempts to negotiate a 3270 session with the foreign host, but drops back to TTY mode (line mode) if that is all that the foreign host can understand. You can override this, if necessary, when you invoke the client.

Starting the Telnet Client

To start a terminal session, you must enter the telnet CICS transaction. If you are using the VSE Interactive Interface (II), press PF9 to clear your screen. Then enter the TELNET command:

```
TELNET {ip-address|symbolic-name}[ ,port][ ,ID=nn][ ,LINEMODE ]
```

The parameters have the following meanings:

- *ip-address* is the IP address of the host. For example, to enter a telnet session with the host at IP address *100.90.90.90*, enter the following command:

```
TELNET 100.90.90.90
```

- *symbolic-name* is the symbolic name that is assigned to the host. To enter a telnet session with the host that your administrator has defined symbolically as *VSE21TST*, enter the following command:

```
TELNET VSE21TST
```

- *port* is the port number that the remote server is monitoring. The default is *23*. If the remote server is monitoring a different port number, you can provide that number after the address.
- *ID=nn* enables you to specify a TCP/IP FOR VSE partition that has an ID other than *00*. The number you specify for *nn* should match the value specified in the PARM field of the EXEC statement that started the TCP/IP FOR VSE partition.
- *LINEMODE* is an optional keyword that prevents 3270-mode negotiation. If you do not specify this keyword and the client attempts to connect in 3270 mode with a remote host that does not support it, the telnet session request fails.

After you enter the TELNET command, you can expect the following response:

```
TELNET 100.50.90.90
TEL200I Telnet CLIENT -- STARTUP --
TEL211I CONNECTING TO PORT: 000023 AT IP: 100.050.090.090 ID:00
TEL209I ATTEMPTING TO ESTABLISH CONNECTION
```

The next messages you see should be from the foreign host. The messages are in full-screen mode (a 3270 session) if the foreign host supports it. Otherwise, they are in line mode.

Using the Batch Telnet Client

As interactions between VSE and remote systems increase, it is sometimes necessary to automate your interactions with the remote systems. The batch telnet client enables you to initiate a telnet session with any line-mode telnet daemon and issue commands to it. The following job stream demonstrates the use of the client. In many instances, it can be used as a replacement for a remote shell client.

```

$$ JNM=RMTEXEC,CLASS=A,DISP=D
$$ LST CLASS=A,DISP=H
// JOB RMTEXEC
// LIBDEF *,SEARCH=(PRD2.TCPIP)
// EXEC TELNET,PARM='IP=ipaddr,PORT=port,ID=nn'
command 1
%WAIT 1
%SUPPRESS
command 2
%WAIT 1
%REVEAL
command 3
. . .
/*
/&
* $$ E0J

```

The parameters are keyword parameters, so you can specify them in any order. The parameters are described in the following table:

Parameter	Description
IP	Specifies the IP address of the foreign host (server) that you want to connect to. This is a required parameter.
PORT	Specifies the TCP port number on the foreign host (server). If you do not specify this parameter, <i>PORT=23</i> is the default.
ID	Specifies the system identifier of the TCP/IP FOR VSE partition that is to serve as the local host (client). Remember that you can have more than one copy of TCP/IP FOR VSE running at one time (such as production and test). If you do not specify this parameter, <i>ID=00</i> is the default.

Interactions

A telnet session consists of commands and responses. Responses received from the remote host are sent to SYSLST. Each record read from SYSIPT is sent to the remote telnet server as an input transaction.

In addition to input records, you can code the meta commands described in the following table:

Command	Description
%WAIT <i>nn</i>	Pauses the telnet client for <i>nn</i> seconds before proceeding. This is useful when, after sending an input line to the remote telnet server, you must wait for a response before sending additional command lines.
%SUPPRESS	Stops the client from sending data to SYSLST. It is appropriate to include %SUPPRESS before providing a password entry.
%REVEAL	Counteracts the effects of %SUPPRESS.

Return Codes

The following table describes the return codes issued from your telnet session.

Return Code	Description
0	Your telnet session completed normally. This does not mean that the commands sent to the foreign system were processed successfully. It simply indicates that the commands were actually sent to the foreign system. The only way that you can verify the results of your commands is to examine the output of the job.
8	A parameter error occurred. This error is accompanied by one or more SYSLST messages that can help you identify the exact cause of the problem. The problem involves either a syntax error in the parameters to the telnet program or a syntax error detected while processing a meta command.
16	TCP/IP FOR VSE is unable to establish a session with the foreign host or, if a session is already established, TCP/IP FOR VSE is unable to send data across the connection to the foreign host. You can also receive this return code if the TCP/IP FOR VSE partition is inactive at the time your batch job runs.
24	The message skeletons cannot be loaded. Check your LIBDEF statement to ensure that it specifies * and not <i>PHASE</i> .

Checking Command Execution

It is appropriate to use the batch telnet client when you want to issue one or more commands to the foreign system and you don't particularly care if the commands are processed successfully. You can use VSE conditional JCL to determine whether the commands are delivered to the foreign host, but the return code does not indicate whether the commands actually processed or that they did what you expected.

You can write a program to check the output of the commands you issue on the foreign host. To do this, use the REXX Sockets Interface. For more information, see the REXX Sockets documentation in the *TCP/IP FOR VSE Programmer's Guide*.

4

Printing Files with LPR

Overview

You can use the TCP/IP FOR VSE Line Printer Requester (LPR) client to print data on printers that are connected to remote TCP/IP hosts. In this chapter we discuss the following topics:

- Setting up the LPR client
 - Running the LPR client, including four different methods you can use to start it
 - Completing an LPR transaction
 - Using LPR client commands, including an LPR command summary and a detailed explanation of each command
 - Using the LPR client with an InfoPrint printer
 - Using a Forms Control Buffer (FCB) to control printing
 - Controlling printers with the INSERTS phase
-

Setting Up the LPR Client

Before you can use the LPR client, you must install a Line Printer Daemon (LPD) on the system that owns the printers you plan to use. To print mainframe data on a desktop printer, you must start an LPD on your desktop. To use network printers, you need to run TCP/IP and an LPD on the Network Operating System (NOS). Some printers can participate directly in your network with no attached host. In this case, the printer runs its own implementation of TCP/IP and, in some cases, an LPD. TCP/IP FOR VSE is compatible with these network-capable printers. For example, TCP/IP FOR VSE can directly access a Hewlett-Packard printer with a JetDirect card.

To begin, you need the following two pieces of information:

- The IP address of the system you plan to print on.
 - The name of an LPD queue. Unfortunately, there is no general method you can use to identify the queue name. You must find someone in your installation that knows the name, or you must issue a platform-specific command to obtain the name. For Windows-based platforms, the queue name is usually the name that Windows assigns to the printer. For HP JetDirect print servers, the queue name is usually *RAW* or *TEXT*. For more information about queue names, see the documentation for the foreign host or the printer.
-

Running the LPR Client

You can run the LPR client using any of the following four methods:

- Interactively using the LPR CICS transaction.
- Automatically using the AUTOLPR facility.
- In batch mode using the LPR batch client.
- In an application program using the REXX, Assembler, COBOL, or PL/1 Sockets interface.

We cover the first three methods in this chapter. For information about using the Sockets interface, see the *TCP/IP FOR VSE Programmer's Guide*.

It is important to remember that the LPR client is actually a program that runs in the TCP/IP stack. While a batch or CICS application may appear to execute the client, each application is actually calling a simple program that issues special socket calls to the stack. These socket calls tell the stack to perform the LPR client activities and return the results of each request. This is why, for example, if you cancel a batch LPR program while the stack is transmitting mainframe data, the data continues to be transmitted. The stack is controlling the transmission.

Using CICS LPR

To start the LPR client from your CICS terminal, log on to CICS and issue the following command:

```
LPR
```

There are no parameters. The LPR transaction will use the SYSID value established by the // OPTION SYSPARM statement in the CICS startup. If SYSID is not set there, a default of 00 is used. This value cannot be modified otherwise. LPR responds by displaying the following messages:

```
LPR
TCP200I Client -- Startup --
TCP207I Copyright © 1996-2006 Connectivity Systems Incorporated
TCP202I Attempting to Establish Connection
TCP204I Connection has been Established
Client manager connection Established.
LPR Ready:
```

The LPR client is now in command mode and can accept LPR client commands. The LPR client commands are explained later in this chapter. To terminate the LPR client from your CICS terminal, use the QUIT command.

Using AUTOLPR

You can use the automatic LPR client (AUTOLPR) to send the output from a job to a specific host and printer automatically. To determine how to handle your output, AUTOLPR processes information from the following sources:

- The DEFINE EVENT command, which your system administrator codes.
- The POWER JECL JOB, LST, and PUN parameters, which you code. Values on the LST and PUN statements in a POWER JECL override values on the * \$\$ JOB statement.
- LPR scripts, which are coded and stored in a .L book by your system administrator.
- LPR insert phases, which you code and assemble.

Each source is discussed in the remainder of this chapter.

If an LPR fails, AUTOLPR can retry the operation up to nine times. The RETRY option in DEFINE EVENT sets this value.

DEFINE EVENT

Before you can use AUTOLPR, your system administrator must use the DEFINE EVENT command to define one or more POWER LST or PUN queues for TCP/IP FOR VSE to monitor. To determine which POWER queues are monitored, you can issue the QUERY EVENTS command from a VSE operator console. When a given LST queue is monitored, TCP/IP FOR VSE uses information from the DEFINE EVENT command and the * \$\$ LST or * \$\$ PUN statement to determine its next course of action.

The following options may be used to define a queue for AUTOLPR. For more information on these options, see the *TCP/IP FOR VSE Command Reference*.

```
DEFINE EVENT, ACTION=LPR, CLASS=power_class, FCBPREFIX=value,
HOSTNAME=[USERINFO|ROOM|DEST|DEPT|BLDG],
ID=event_name, NULLFILE=[SKIP|PROCESS],
ORDER=[YES|NO], PASSWORD=value, POWERSYSID=id,
PRIORITY=[YES|NO], QUEUE=[LST|PUN], RETRY=count,
RETRY_TIME=interval, SINGLE=[YES|NO],
SCRIPTTYPE=[L|file_extension], USERID=id
```

**POWER JECL
Statements**

The format of the * \$\$ LST or * \$\$ PUN statement that you use depends primarily on two factors:

- Whether you plan to use an automatic LPR script
- The HOSTNAME that the system administrator specified on the DEFINE EVENT command for the LST or PUN class you are using.

To complete your * \$\$ LST or * \$\$ PUN statement, you also need to know the IP address and the printer name (queue name) of the printer you want to use.

The formats of the * \$\$ JOB and * \$\$ LST statements that you include in your POWER JECL are as follows:

```
* $$ JOB ROOM=name,DEPT=name,BLDG=name
* $$ LST CLASS=class,DISP=disp,USER='name',DEST=(*,prname),
* $$ LST UCS=inserts,FCB=fcb,BLDG=name,ROOM=name,
* $$ LST DEPT=name
```

The formats in these statements only show the relevant options.

The variables in the above statements are described in the following table:

Variable	Description
<i>class</i>	The POWER LST queue job class. TCP/IP FOR VSE must monitor this class as directed by the DEFINE EVENT command.
<i>disp</i>	The output disposition of the report. AUTOLPR only looks for and processes DISP=D or DISP=K entries. By default, DISP=D entries are deleted after being sent, while DISP=K entries are held. For more information, see “SET DISP” (page 4-41).
<i>fcb</i>	The name of a Forms Control Buffer (FCB). An FCB permits you to emulate more complex line spacing. See the section “Using an FCB to Control Printing” (page 4-69) for details.
<i>inserts</i>	The name of an INSERTS phase. This phase is an assembled module that contains binary strings the user has defined to be inserted automatically into different parts of a report. The INSERTS phase is discussed later in this chapter.

Variable	Description
<i>name</i>	<p>A string representing the address of the printer you want to print to. The variable <i>name</i> can be one of the following values:</p> <ul style="list-style-type: none"> • An IP address in dotted decimal notation, such as <i>192.168.0.7</i> • A script name • A symbolic IP address, which can be a name defined with the DEFINE NAME command or a name obtained with a domain name server <p>To determine which value <i>name</i> represents, TCP/IP FOR VSE checks whether it is one of the following:</p> <ul style="list-style-type: none"> • A valid IP address. To specify an IP address in dotted decimal notation, your system administrator can use the DEFINE EVENT command with the HOSTNAME=USERINFO parameter (for example). For <i>USERINFO</i>, the IP address is coded in the USER= field. • A script name that is defined by your system administrator using the DEFINE NAME command. A <i>script</i> is a set of LPR commands that is contained in a library. For more information about defining scripts, see the DEFINE NAME command in the <i>TCP/IP FOR VSE Command Reference</i>. • A symbolic name for an IP address or an LPD domain name. A symbolic name is defined with the DEFINE NAME command. If TCP/IP FOR VSE finds a symbolic name, it substitutes the associated IP address. If it does not find a symbolic name, it checks the Domain Name Server (if one is defined) to see if it can resolve the <i>name</i> value. If the value cannot be resolved, the automatic LPR fails. <p>For more information about specifying <i>name</i>, see the examples at the end of this chapter.</p>
<i>prname</i>	The printer name (or queue name) on the foreign system.

LPR Scripts

You can define a limited amount of information on JECL statements. To manipulate output using LPR client commands, you must create an LPR script. Here is the procedure:

1. Code a set of LPR commands. You must always supply a host name and a printer name. You can put this information in your script, or you can put it directly in the * \$\$ LST statement. Note that values in the POWER JECL statements override the values specified in the * \$\$ JOB statement. The values coded in LPR client commands, in turn, override any specifications on the POWER JECL statements.

2. Save the commands in a VSE library member that has a type that matches the SCRIPTTYPE option value on the DEFINE EVENT command. The default is '.L'. This member needs to be in a VSE library that is part of a SOURCE (or *) LIBDEF search chain within the TCP/IP FOR VSE partition.
3. Use the DEFINE NAME command to define the VSE library member as a script.
4. Identify the script by specifying its name as described in the *name* field in the above table.

Command Variables

The following command variables are available when creating AUTOLPR scripts. These variables are common to AUTOFTP and AUTOEMAIL. If user-defined POWER field names are included in the LST card (for example) that created them, then those variables will also be available. For details on defining user variables, see the IBM manual *VSE/POWER Administration and Operation*.

Variable	Description	Length
&CPUID	Numeric CPU identifier	6
&CPUSECS	Current time in 1/300 th -second units. This numeric value may be used as a random number for generating a unique file ID.	9
&CURDATE	Current date in the form <i>mmddycc</i> , where <i>mm</i> is the month, <i>dd</i> is the day, <i>yy</i> is the year, and <i>cc</i> is a century of 19 or 20. Note: Unlike the external FTP batch client, you do not need to issue the SETVAR command to access this variable.	8
&CURTIME	Current time in the form <i>hhmmss</i> , where <i>hh</i> is the hour (using a 24-hour clock), <i>mm</i> is the minute, and <i>ss</i> is the second. Note: Unlike the external FTP batch client, you do not need to issue the SETVAR command to access this variable.	6
&PWRBLDG	Building name assigned to the queue entry from the BLDG= parameter on the * \$\$ JOB statement	8
&PWRCDAT	Synonym for &PWRXDAT	
&PWRCLAS	LST or PUN class for the listing that is being transferred, from the * \$\$ LST or the * \$\$ PUN statement	1
&PWRCOPY	Number of requested copies as set by the COPY= parameter on the * \$\$ LST statement	2

Variable	Description	Length
&PWRCTIM	Synonym for &PWRXTIM	
&PWRDEPT	Department name assigned to the queue entry from the DEPT= parameter on the * \$\$ LST statement.	8
&PWRDEST	Name of the destination on the * \$\$ LST statement. This can also be the name of the '.L' book from which the automatic LPR client pulls LPR client commands.	8
&PWRDISP	Disposition for the LST or PUN queue entry from the DISP= parameter on the * \$\$ LST or * \$\$ PUN statement	1
&PWRDIST	Distribution code associated with the POWER queue entry from the DIST= parameter of the * \$\$ LST statement.	8
&PWRFLSH	FLASH field identifier. If not specified, it is blank. It is used for laser printers.	4
&PWRFLSN	Number of FLASH copies for laser printers. The default for this numeric field is 000.	3
&PWRFORM	Form identifier from the FNO= parameter on the * \$\$ LST statement.	4
&PWRFRMD	Special FORMDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRLCNT	Number of lines in the file	8
&PWRNAME	Name of the job that sent the output to the POWER LST or PUN class that TCP/IP FOR VSE is monitoring	8
&PWRNUMB	Number that POWER assigned to the job	5
&PWROFCB	Name of the Forms Control Buffer (FCB) specified on the FCB= parameter on the * \$\$ LST statement	8
&PWROUCB	Name of the character set specified on the UCS= parameter of the * \$\$ LST statement	8
&PWRPAGD	Special PAGEDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRPCNT	Number of pages in the file in numeric format. This is for LST data only.	8

Variable	Description	Length
&PWRPRGN	Name of the programmer from the PROGR= parameter on the * \$\$ JOB statement.	20
&PWRPRTY	Priority of the queue entry from the PRI= parameter on the * \$\$ JOB or * \$\$ LST statement	1
&PWRQDAT	Synonym for &PWRXDAT	
&PWRQTIM	Synonym for &PWRXTIM	
&PWRQUE	Either <i>PUN</i> or <i>LST</i> to specify the queue the listing came from	3
&PWRROOM	Room number assigned to the queue entry from the ROOM= parameter on the * \$\$ JOB statement	8
&PWRSID	System ID that is to process the queue entry, from the SYSID parameter on the * \$\$ LST statement	1
&PWRSUFF	Suffix that POWER assigned to the listing	3
&PWRTNOD	Target node name for the queue entry in a PNET environment, as specified on the LDEST parameter of the * \$\$ JOB statement for the LST queue and the PDEST parameter of the * \$\$ JOB statement for the PUN queue	8
&PWRUINF	User information field specified on the USER= parameter from the * \$\$ JOB statement or the * \$\$ LST statement. This field is arbitrary and may be changed with the PALTER command.	16
&PWRUSRI	VSE user ID associated with the job	8
&PWRXDAT	Date of the POWER queue entry in either <i>ddmmyyyy</i> or <i>mmddyyyy</i> format, depending on the IPL option of your VSE system. This value reflects when the entry was initially created or opened.	8
&PWRXTIM	Time of the POWER queue entry in <i>hhmmss</i> , where <i>hh</i> is the hour using a 24-hour clock. This value reflects when the entry was initially created or opened.	6
&QNUM	Unique POWER queue member identifier. (This is not the POWER job number.)	5

AUTOLPR Output

The following table shows how TCP/IP FOR VSE processes the queue entry after the LPR operation. The LPR client command SET DISP may change the actions in the table. For more information about disposition processing with AUTOLPR, see the command SET DISP ([page 4-41](#)).

* \$\$ LST Disposition	Successful AUTOLPR	Unsuccessful AUTOLPR
DISP=D	The job is deleted from the queue.	After the first unsuccessful attempt, the job is requeued. If all subsequent attempts fail, the report's POWER disposition is changed to "Y" and it remains in that state. If you use the PALTER command to change the disposition to K or D, AUTOLPR queues the request again.
DISP=K	The job is kept in the POWER LST or PUN queue with a disposition of L.	

AUTOLPR Example 1

In our first example, the system administrator uses the DEFINE EVENT command to monitor POWER LST queue output in class Z. The DEFINE EVENT command also specifies *HOSTNAME=USERINFO*. When the following job runs, it automatically uses LPR to print the output at IP address *192.168.0.7* on printer name *local*.

```
* $$ JOB JNM=LIBRLIST,CLASS=A,DISP=D
* $$ LST CLASS=Z,DEST=(*,LOCAL),USER='192.168.0.7'
// JOB LIBRLST
// EXEC LIBR
LISTDIR S=PRD2.TCPIP
/*
/&
* $$ EOJ
```

AUTOLPR Example 2

In this example, the system administrator uses the DEFINE EVENT command to monitor POWER LST queue output in class Z. The DEFINE EVENT command also specifies *HOSTNAME=DEPT*. When the following job runs, it automatically uses LPR to print the output at the IP address associated with symbolic name *VMSYS* on printer name (or queue name) *local*.

```
* $$ JOB JNM=LIBRLIST,CLASS=A,DISP=D,DEPT=VMSYS
* $$ LST CLASS=Z,DEST=(*,LOCAL)
// JOB LIBRLST
// EXEC LIBR
LISTDIR S=PRD2.TCPIP
/*
/&
* $$ EOJ
```

**AUTOLPR
Example 3**

In this example, the system administrator uses the DEFINE EVENT command to monitor the POWER LST queue output in class Z. The DEFINE EVENT command also specifies *HOSTNAME=ROOM*. The system administrator also defines a script named NTPORT to print to an LPD that is the final destination. (This LPD can reside on VSE or anywhere in the network that TCP/IP FOR VSE can route the data. For information on how to connect to an LPD, see the LPD vendor's documentation.)

After the following job runs and its output is in the POWER LST queue, TCP/IP FOR VSE takes control. It searches its local name table for *NTPORT*. It determines that *NTPORT* is not a symbolic representation of an IP address and then checks for a script with that name. When it finds the script, AUTOLPR reads and processes the commands in it.

```
* $$ JOB JNM=LIBRLIST,CLASS=A,DISP=D,ROOM=NTPORT
* $$ LST CLASS=Z,DEST=(*,LOCAL)
// JOB LIBRLST
// EXEC LIBR
LISTDIR S=PRD2.TCPIP
/*
/&
* $$ EOJ
```

Using Batch LPR

The TCP/IP FOR VSE LPR client has a batch mode that allows you to start LPR from a batch environment. The following example shows a batch job's format:

```
* $$ JOB JNM=LPRBATCH,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB LPRBATCH
// EXEC CLIENT,PARM='APPL=LPR,ID=nn,QUIET=YES'
SET HOST address
SET PRINTER printer
Lpr command1..
/*
/&
* $$ EOJ
```

Parameters The EXEC CLIENT parameters are keyword parameters, so you can specify them in any order. They are described in the following table:

Parameter	Description
APPL=LPR	This parameter is required to run LPR.
ID= <i>nn</i>	Specifies the system ID of the TCP/IP partition to which you want to connect. Remember that you can run multiple copies of TCP/IP at one time (for example, production and test). The default is 00.
QUIET=YES	Directs LPR to suppress informational messages from the VSE console. (YES is the only valid value.)

Commands You can issue any number of LPR commands using the batch LPR client. You must use the SET command to specify the host name and the printer name before you issue a PRINT command. (If you do not issue a PRINT command, the whole exercise is probably not worth the effort.) For a summary of LPR commands, see the section [“Using LPR Client Commands”](#) (page 4-17).

Return Codes The return codes for LPR client are described in the following table:

Return Code	Description
0	Your LPR commands completed successfully.
4	A syntax error occurred in one of the LPR commands in the job stream. To find the problem, study the output. Common problems include an inability to change to the specified directory, file not found, and printer not found.
8	An error occurred in one of the LPR commands in the job stream. The error could be a syntax error or it could be another type of error. To find the problem, study the output. Note that one or more LPR commands might have completed successfully, so you need to determine the current status before you rerun the job.
12	One of the parameters was incorrectly specified.
16	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that the ID= parameter value corresponds with the SYSID for an active TCP/IP FOR VSE partition.

Generating the Batch LPR Job

At times it might be useful to use a VSE/REXX program to generate the LPR batch job stream for you. This enables you to print multiple files in the same job. The following example shows how VSE/REXX is used and how it can issue VSE/POWER commands and retrieve the response. Based on the response to a VSE/POWER command, an LPR batch job stream is created and submitted. This sample is available from CSI International's website at www.csi-international.com/download.htm.

Note: While this is one method of generating a batch LPR job to execute on VSE, a more direct method is to create a REXX program that actually issues the SOCKET calls. For more information on REXX sockets programs, see the *TCP/IP FOR VSE Programmer's Guide*.

```
* $$ JOB JNM=MULTILPR,CLASS=4,DISP=D,ECHO=(ALL,VSEUSER)
* $$ LST CLASS=A,DISP=D,DEST=(,VSEUSER)
// JOB MULTILPR (VSEUSER)
// LIBDEF *,SEARCH=(PRD2.TCPIPCFG,PRD2.TCPIP)
// EXEC LIBR
ACC SUB=PRD2.MARC
CATALOG MULTILPR.PROC                REPLACE=YES
/* REXX */
/*****/
/* This REXX program issues a "D Q,LST,jobname"          */
/* command, traps the response, and then creates a      */
/* job that LPRs all entries in the list.                */
/* All of the entries have, by necessity, the           */
/* same characteristics. You can adjust these           */
/* characteristics by adjusting the constants in        */
/* the beginning of the program.                         */
/*                                                       */
/*                                                       */
/*****/
say 'Initializing Multiple LPR facility'
jobname = 'ALL'           /* Jobname to LPR goes here          */
printerip = '192.168.0.7' /* LPR address to LPR to goes here */
printername = 'LOCAL'    /* Printer name goes here           */
insertsname = 'NONE'     /* Inserts Phase goes here          */

address 'CONSOLE'
'ACTIVATE NAME MASTER PROFILE REXNORC'
if rc \= 0 then,
do
  say 'Problem with ACTIVATE - MULTILPR TERMINATING'
  exit
end
say 'Multilpr - ACTIVATE Successful'
'CONSWITCH MASTER'
if rc \= 0 then,
do
  say 'Problem with CONSWITCH- MULTILPR TERMINATING'
  exit
end
say 'Multilpr - CONSWITCH Successful'
address 'CONSOLE'
'CART RED'
```

```

'D LST,'jobname          /* Issue the D Q,LST command          */
rc = getmsg(msg.,'RESP','RED',,100)
say 'MultiLPR - return code from D LST command was' rc
say 'number of lines is' msg.0
nl = msg.0
nj = 0
do i = 3 to nl
  nj = nj + 1
  listjobname.nj = word(msg.i,4)          /* pick up job name    */
  listjobnum.nj = word(msg.i,5)          /* pick up job number */
  listjobqueue.nj = word(msg.i,8)       /* pick up job class  */
end
say 'MultiLPR -- following is a list of jobs that are printing'
do i = 1 to nj
  say listjobname.i listjobnum.i
end
'DEACTIVATE MASTER'
/*****
/* Now that we have our list of jobs to print, let's          */
/* build our batch LPR job.                                    */
/*                                                              */
/*****
jcl.1 = '* $$ JOB JNM=LPRBATCH,CLASS=4,DISP=D'
jcl.2 = '* $$ LST CLASS=A,DISP=D,DEST=(,VSEUSER)'
jcl.3 = '// JOB LPRBATCH (VSEUSER,ECHO=YES)'
jcl.4 = '// LIBDEF *,SEARCH=(PRD2.TCPIP)'
jcl.5 = "// EXEC CLIENT,PARM='APPL=LPR'"
jcl.6 = "set host="printerip
jcl.7 = "set printer="printername
jcl.8 = "cd power.lst.all"
njcl = 8
if insertsname \= 'NONE' then,
do
  njcl = njcl + 1
  jcl.njcl = 'SET INSERTS='insertsname
end
njcl = njcl + 1
jcl.njcl = 'Q OPT'          /* document the options in effect */
/*****
/* Generate the PRINT command for each file to be            */
/* printed.                                                  */
/*                                                              */
/*****
do i = 1 to nj
  njcl = njcl + 1
  jcl.njcl = 'PRINT' listjobname.i.'listjobnum.i
end

```

```

/*****
/* The JCL to print each file has been generated;          */
/* now we must generate end-of-job statements.            */
/*                                                         */
/*****
njcl = njcl + 1
jcl.njcl = '/'
njcl = njcl + 1
jcl.njcl = '/&'
njcl = njcl + 1
jcl.njcl = '* $$ E0J'
jcl.0 = njcl                                           /* capture number of lines */
address 'POWER'
'PUTQE RDR STEM jcl.'
exit
/+
/*
// EXEC REXX=MULTILPR
/&
* $$ E0J

```

Generating LPR Separator Pages

TCP/IP FOR VSE controls whether POWER-generated page separators are permitted for LPR operations. The system administrator can use the SEPARATOR_PAGES command to enable or disable page separators for printing. This setting applies to all LPR operations, meaning that separator pages are either enabled or disabled for all LPR operations. If they are enabled, LPR asks POWER to generate them. (Separator pages are valid for POWER output only.)

One separator page is requested by default. This value is set by the SET PAGE_COUNT command during the TCP/IP initialization process. See the *TCP/IP FOR VSE Command Reference* for more information on this command and the SEPARATOR_PAGES command. To override the default page count when you print, use SET JSEPCOUNT ([page 4-51](#)) in your JCL. This LPR client command allows you to request a different number of separator pages or no pages at all.

Completing an LPR Transaction

Use the following steps to complete an LPR session:

1. Start the LPR client.
 2. Issue LPR client commands to identify the foreign host and queue name.
If you are using the automatic LPR option, this series of commands can be issued automatically.
 3. Issue additional LPR client commands to specify additional parameters.
 4. Issue LPR client commands to print the file.
 5. Terminate the LPR client by issuing the QUIT command.
-

Using LPR Client Commands

LPR provides a set of commands that you can use to direct the print operation. In general, these commands are valid from all LPR applications, including CICS, AUTOLPR, and batch. The commands fall into three categories:

- Execution control statements. These statements are used in batch files and scripts and include *, /, , GOTO, and IF.
- SET commands. These commands specify values that must be set before you issue the PRINT command. You must use SET HOST and SET PRINTER. Depending on your system, you may need to use other SET commands as well.
- PRINT and other client commands.

All commands are summarized in the following table. The SET commands are listed last. A detailed explanation of each command follows this table.

Command	Description
* <i>comment</i>	Adds a comment line or disables a command.
/ <i>label</i>	Labels a line that is the target of a GOTO skip.
CD CDUP	Changes the current directory of your LPR client. CDUP backs up one directory level.
DIR LS	Lists the specified directory in the TCP/IP FOR VSE file system.
DUMPDATA	Produces a data dump for diagnosis when LPR executes.
EXEC	Processes a series of LPR commands in a script file.
GOTO	Allows program flow to jump to another line in a batch file or script.
IF	Controls the next command's execution based on conditional logic.
INITVAR	Defines a variable and assigns a value to it in the current LPR environment.
LONG SHORT	Displays a longer view of the printer queue or a short summary of the printer queue for a given IP address/printer combination.
NOP	Executes without consuming cycles; performs no operation.
PRINT	Prints a file on the specified remote host and printer combination.

Command	Description
PWD	Displays the current working directory.
QUERY OPTIONS	Lists option values currently in effect. Issuing a “Q OPTIONS,SYS” before a PRINT command is a good way to document the LPR parameters in SYSLST.
QUIT	Terminates an LPR session.
SAY	Outputs a string to the console.
SETVAR	Sets the value of an existing variable in the current LPR client environment.
START	Tests a remote LPR host by sending a command to activate a printer queue connected to that host.
SET ALIGN	For FCB-based printing, specifies the line at which printing should start at top of each page.
SET ASA	Individual print lines from a POWER file are transmitted to the LPD with the ASA carriage control codes intact.
SET BADPDF	Specifies whether to continue sending a PDF file if the PDF generator returns an error.
SET BANNER	Specifies that a banner page should be printed before the file is printed.
SET BLANK	Specifies a hex string to be sent for a blank line.
SET CC	Specifies whether the LPR client should print your file with standard carriage control.
SET CHANNEL x	For FCB-based printing, sets a hex string to position the print line to the x channel.
SET CHANNELONLY	For FCB-based printing, specifies that channel positioning should be performed only by skipping lines and not by using the hex strings specified by SET CHANNEL x .
SET CLASS	Requests that the LPD set the output class.
SET CONTROL	Prevents LPR from adding commands and settings to the control file except as provided by SET DYNCONTROL.
SET COPIES	Specifies how many copies of a particular file you want to print.
SET CRLF	Specifies whether to insert carriage return/line feed (CR/LF) combinations into the data stream.

Command	Description
SET DATAFILE	Specifies whether the data file is sent to an LPD before or after the control file or whether the control file is sent at all.
SET DEFAULT_INSERTS	Specifies a default phase name for SET INSERTS.
SET DELREQ	Specifies whether your file is to be deleted from the print queue on the foreign host.
SET DISP	Specifies the disposition of a POWER LST or PUN queue entry after the PRINT command executes.
SET DYNCONTROL	Adds a user-defined control string to the control file.
SET EOD	Specifies the hex string used to indicate end of data.
SET EOF	Specifies the hex string used to indicate end of file.
SET EOL	Specifies the hex string used to indicate each line end.
SET EXTRA	Specifies the hex string used to skip extra lines when positioning the line according to an FCB (skip to channel).
SET FCB	Specifies the name of the phase that is to be used as the Forms Control Buffer (FCB).
SET FCBPREFIX	Specifies the value of the “\$\$\$” prefix in an FCB name.
SET FILENAME	Assigns a new name to the data file being sent to an LPD.
SET FNO	Adds a forms number to the control file.
SET FORMDEF	Adds a FORMDEF value to the control file when required by the LPD.
SET HOST SET IP SET IPADDR	Specifies the IP address of the host on which the LPD is running.
SET INFOPRINT	Specifies that you want to send a job to an InfoPrint printer.
SET INSERTS	Identifies a phase whose contents are to be transmitted with the file.
SET JOBNAME	Specifies the job name that is used on separator pages created by the host you that you are printing on.

Command	Description
SET JSEPCOUNT	Overrides the separator page count requested for POWER-generated reports.
SET LASTCOLUMN	Specifies the last column to be read in a command.
SET LEAVEBLANK	Specifies whether to truncate trailing blanks.
SET LONGLINE	Specifies that lines more than 256 bytes long are sent without truncating the data. Note: This command is no longer needed but is accepted for backward compatibility.
SET NOEJECT	Suppresses the initial form feed character at the beginning of a listing.
SET OPTION	Specifies whether to include the generic option parameter in the control file. This option is needed to add certain values.
SET PAGEDEF	Adds a PAGEDEF value to the control file when required by the LPD.
SET PAUSE	Specifies a delay after printing each copy when multiple copies are requested.
SET PCL5	Suppresses all formatting and translation options. Used for PCL printing.
SET PDF	Specifies that a document is to be converted to a PDF file.
SET PRINTER	Specifies the printer that you want to use.
SET RECFM SET BLKSIZE SET LRECL	Specifies the characteristics of the file to be printed.
SET REPEAT	Specifies whether the LPR client (on VSE) or the line printer daemon (LPD) is responsible for generating multiple copies.
SET RPORT SET LPORT	Sets the remote TCP port that LPR connects to on the foreign host and sets the local TCP port that LPR uses on VSE.
SET SENDLENGTH	Specifies whether you want to send a file's size with the file.
SET SKIP0	Specifies a hex string to skip zero lines (an overstrike).
SET SKIP1	Specifies a hex string to skip one line.
SET SKIP2	Specifies a hex string to skip two lines.
SET SKIP3	Specifies a hex string to skip three lines.

Command	Description
SET SOD	Specifies a hex string to indicate start of data.
SET SOSI	Controls shift-in, shift-out (SOSI) processing of double-byte character sets (DBCS).
SET TITLE	Specifies a title that prints on the title page of the output produced by the LPR client.
SET TOF	Specifies the hex string to position printing to top of form.
SET TRANSLATE	Specifies the name of a translate table for EBCDIC-to-ASCII translation.
SET USER SET PASSWORD	Identifies the user to the TCP/IP FOR VSE security system.
SET VALIDATE	Specifies whether to perform an initial translation of unprintable characters to blanks.

* *comment*

The syntax is

```
* comment
```

An asterisk (*) at the beginning of a line disables command processing on that line in batch files and scripts. All text on the line is ignored. This control is used to add comments to the code. It can also be used to deactivate a command during testing.

When a * *comment* line is used in a script, the passed comment is echoed back to the requestor and is followed by a “Ready: ” user prompt.

Note: Use a ‘/’ at the beginning of a line to disable statements beginning with “* \$\$”.

/ . *label*

The syntax is

```
/ . label
```

This statement labels a line that is the target of a GOTO skip in a batch file or script. The *label* can be eight characters long and can contain letters and numerals. It must exactly match the GOTO command’s argument. If a label statement is repeated, the GOTO skips to the first matching statement.

See also **GOTO**.

CD, CDUP

The syntax is

```
CD directory-name  
CDUP
```

The CD command changes the current directory of your LPR client. This enables your LPR client to process commands using the appropriate directory within the TCP/IP FOR VSE file system. To display the LPR client's current working directory, use the Print Working Directory (PWD) command.

Variable *directory-name* is the name of the directory that you want to be the active directory. There is no default. It must be a valid directory that is contained in the TCP/IP FOR VSE hierarchical file system. You can issue the DIR command to display the directories and files in the current working directory. The command *CDUP* or "*CD ..*" moves the current working directory up one level in the hierarchy. After a CDUP is performed, a PWD command executes automatically to show the current directory pointer.

The anticipated response is shown in the following example:

```
cd prd1  
Change has completed.  
Ready:
```

DIR, LS

The syntax is

```
DIR directory-name  
LS directory-name
```

The DIR command lists the specified directory in the TCP/IP FOR VSE file system. The DIR and LS commands are identical. The variable *directory-name* is the name of the directory you want to list. The name you specify may contain regular characters and wild card characters. Each file in the current working directory is matched against the name you specify, and all names that match are displayed. There is no default. The following wild card characters are accepted:

- An asterisk (*), which matches any number of characters
- A question mark (?), which matches any one character

As an example, assume we have a directory with the following files:

```
f1.txt  
file2.bjb  
fn3.zip  
filenum4.wav  
x.lst
```

The following table shows the effect of using the DIR command with wild card characters:

Command	Files Contained in Output Listing
DIR f*	f1.txt, file2.bjb, f3.zip, filenum4.wav
DIR f*.Z*	fn3.zip
DIR F?.TXT	f1.txt
DIR F?*	f1.txt, file2.bjb, f3.zip, filenum4.wav
DIR *	All files
DIR ?	No files are returned because the ? matches only one character

**DUMPDATA,
DUMPFGB,
DUMPINSERTS,
DUMPVARS**

The syntax is

```
DUMPDATA
DUMPFGB
DUMPINSERTS
DUMPVARS
```

These commands enable various data dumps that can assist with troubleshooting LPR problems. You should run them when CSI Technical Support requests that you use them for problem diagnosis.

These commands produce the following results:

- DUMPDATA: a dump of the data being sent to the LPD.
- DUMPFGB: a dump of the FCB data for an FCB-based job.
- DUMPINSERTS: a dump of the INSERTS data when using AutoLPR.
- DUMPVARS: a dump showing the value of the variables in effect.

See also DIAGNOSE in the *TCP/IP FOR VSE Command Reference*. This command can be used with the LPR option to enable diagnostic messages for troubleshooting LPR-related processes.

EXEC

The syntax is

```
EXEC membername
E membername
```

The EXEC command enables you to process a series of LPR commands that are contained in a script file. When LPR finds an EXEC command, it stops processing the commands in the input stream. It immediately processes the commands listed in *membername*, and then resumes processing the suspended input stream. This means that the statements in *membername* are logically included in the input stream as opposed to being called as a subroutine. If *membername* contains a QUIT command (which should be avoided), then the session actually QUITs.

Variable *membername* is an L book that is contained in the search chain for the TCP/IP FOR VSE partition. LPR automatically appends the *.L* suffix, so you do not need to specify it. There is no default.

GOTO

The syntax is

```
GOTO Label
```

This command allows program flow to jump to another line in a batch file or script. It often is used with an IF statement. It causes all JCL statements to be skipped up to the line containing the matching */.label* statement. Command processing then continues with the statement immediately following the label statement. If a matching label is not found, then the GOTO skips to the end-of-job statement.

For example, the command GOTO EXIT causes the lines that follow to be ignored up to the */.EXIT* label statement. Processing resumes with the next statement in the job stream.

This command is useful for creating conditional jobs as well as branching over normally used commands during testing.

See also */.label*.

IF

The syntax is

```
IF variable relational-operator constant [THEN]
```

An IF statement checks a condition and executes the statement that follows if the condition is true. If the result of the IF test is false, the statement following the IF statement is bypassed. Typically, a variable is compared to a constant. The first value (*variable*) is always compared to the second value (*constant*). The word “THEN” is optional.

The relational operators are described in the following table:

Relational Operator	Condition Checked
EQ =	Equals
NE <>	Not equals
GT >	Greater than
LT <	Less than
GE >=	Greater than or equals (not less than)
LE <=	Less than or equals (not greater than)

In the following example, a GOTO command executes if the condition in the IF statement is true:

```
IF &CURTIME LT 120000 THEN
GOTO MORNING
<commands>
/. MORNING
```

INITVAR

The syntax is

```
INITVAR &variablename = value
```

The INITVAR command defines a variable and assigns a value to it in the current LPR environment only if the variable does not exist. If the variable name exists, INITVAR does nothing. This is different from SETVAR, which always assigns a value to a variable.

The variables in the above syntax statement have the following meanings:

- *variablename* is the name of a variable you want to create. Names can be up to 8 bytes long. The first character is an ampersand (&) and is followed by up to seven alphanumeric characters.
- *value* is the value you want to assign. It must be non-null.

In the following example, "NOPLACE" is assigned as the default for the variable &PWRDIST:

```
INITVAR &PWRDIST="NOPLACE"  
SETVAR &WORKVAR="-oroom=" + &PWRDIST  
SET DYNCONTROL=&WORKVAR
```

If a value for &PWRDIST is passed to the script, it will be retained and used. Otherwise, the variable &PWRDIST is created with the specified default.

See also **SETVAR**.

LONG, SHORT

The syntax is

```
L  
LONG  
SHORT  
SH
```

The LONG command displays a long view of the printer queue. The SHORT command displays a short summary of the printer queue for a given IP address/printer combination. Depending on the specific LPD, the commands may return identical output. Before you issue these commands, you must set both the host name and the printer name. There is no default.

The anticipated response depends on the LPD. The responses shown in the following two examples are from the VM LPD and the Windows LPD. The first example uses the SHORT command:

```

short
Establishing connection with LP Daemon
Remote port: 515 Local Port: 721
Requesting the queue information
Receiving LP daemon response
SYS1: spooling to VM
Rank  Owner      Job Files      Total Size
active TCPIP      32  WT0.A      236080 bytes
Closing the connection with LP Daemon
Connection complete - Already closed
Ready:

```

The second example uses the LONG command:

```

long
Establishing connection with LP Daemon
Remote port: 515 Local Port: 721
Requesting the queue information
Receiving LP daemon response
                                Windows xx LPD Server
                                Printer \\192.168.0.1\FILE
Owner      Status      Jobname  Job-Id  Size  Pages Priority
-----
TCPIP (192. Printing WT0.A          10   236080    0    1
Closing the connection with LP Daemon
Connection complete - Already closed
Ready:

```

NOP

The syntax is

```
NOP
```

A NOP is a no-operation command in batch files and scripts. It is similar to the '*' statement in that it performs no action. The client recognizes it and then immediately stops processing it. NOP can be used to test communication between the batch client and the stack, for example. It has no arguments.

PRINT

The syntax is

```
PRINT file-name  
P file-name
```

This command transfers the file to the remote site where it is printed on the specified printer. Before you issue this command, you must set both the host name and the printer name.

Variable *file-name* is a valid file in the TCP/IP FOR VSE hierarchical file system. The name of the current working directory always precedes *file-name*, thus forming a fully qualified file name. For example, if the current working directory is *PRDI.MACLIB* and you issue command *PRINT WTO.A*, LPR prints the file *PRDI.MACLIB.WTO.A*. If you issue a CD to change the directory, you would need to issue a *PRINT PRDI.MACLIB.WTO.A* command to print the same file.

For a POWER print queue entry, you can specify a file name using the following formats:

```
PRINT name  
PRINT name.num  
PRINT name.num.seg
```

PRINT matches as much data as you specify. The first file found that matches the pattern is printed. You can print only one file with any single invocation of the PRINT command, but you can include any number of PRINT commands in the job stream.

When you invoke LPR from the batch client (// EXEC CLIENT), PRINT can send multiple LST QUEUE entries with one statement (starting in column 1). To do this, you must CD to the POWER.LST.ALL directory and then use one of the following patterns. Only members that match the pattern *and* have a disposition of *D* (DISP=D) are printed.

```
CD POWER.LST.ALL  
PRINT job_name/  
PRINT job_name/CLASS=n/TUSER=xxxx/FORM=xxxx
```

For example, "PRINT LSVCAT8/CLASS=Z" returns all reports named LSVCAT8 (with different job numbers) in class Z with DISP=D. If you specify a job name, it must match, or you can add an asterisk (*) to a job name to create a pattern. You can also use the POWER keyword *ALL*, as follows:

```
PRINT ALL/CLASS=A/FORM=STD
```

In this example, the PRINT command selects all jobs from class A with FNO=STD.

If you include “DEBUG=ON” in your “PARM=” string, then no actual printing is performed. The generated PRINT statements appear in the listing as comments.

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Prints the requested POWER LST or PUN queue member. You can include additional PRINT commands in your script. AUTOLPR does not require a PRINT command; if you include one, two copies of the output are printed.
Batch	None

The anticipated response is shown in the following example:

```
PRINT PRD1.MACLIB.WTO.A
Opening the file
Establishing connection with LP Daemon
Remote port: 515 Local Port: 721
Request the queue for output
Checking LP daemon
Transferring the data file
Data file transferred.
Transferring the control file
Control file transferred.
Closing the connection with LP Daemon
Connection complete
Ready:
```

PWD

The syntax is

```
PWD
```

The PWD command displays the current working directory. There is no default.

The anticipated response is shown in the following example:

```
pwd
"PRD2 . JOHN"
Ready:
```

QUERY OPTIONS

The syntax is

```
QUERY OPTIONS
Q OPTIONS
```

This command lists the options currently in effect for the LPR client. The displayed options and their values may change from release to release. After installing a new release of TCP/IP FOR VSE, you may want to run a non-printing "Q OPTIONS" LPR job to see what commands may be new to your release.

QUIT

This command causes an LPR session to terminate. There are no arguments.

SAY

The syntax is

```
SAY string
```

This command outputs a string to the console. It functions the same as the REXX SAY command and resolves all variables in the string before outputting the string.

The following example contains a variable. The text is displayed on the console.

```
SAY The current time is &CURTIME -- Begin PRINT process
```

SETVAR

The syntax is

```
SETVAR &variablename = value
```

The SETVAR command sets the value of an existing variable in the current LPR client environment. Variable names can be up to 8 bytes long. The first character is an ampersand (&) and is followed by up to seven alphanumeric characters.

Note: To create a variable, use the INITVAR command.

The labels in the above syntax statement have the following meanings:

- *variablename* is the name of a variable.
- *value* is the value you want to assign.

The following rules apply:

- You can assign a literal value to a variable by enclosing the value in quotes. For example, to set variable *&V1* to *ABC*, you would use

```
SETVAR &V1="ABC"
```

- You can set a variable to any other variable. For example, to set variable *&V2* to the value of variable *&V1*, you would use the following statement. If the value of *&V1* is *ABC*, as shown in the previous example, then *&V2* is set to *ABC*.

```
SETVAR &V2=&V1
```

- You can set a variable to a substring of another variable, as shown in the following example. After the command executes, variable *&V3* is set to *BC*.

```
SETVAR &V3=SUBSTR(&V2,2,2)
```

- You can concatenate values in a variable by using the plus sign (+). For example, after the following command executes, variable *&V4* is set to *ABCDEF*.

```
SETVAR &V4=SUBSTR(&V1,1,1) + &V3 + "DEF"
```

- You can use variables anywhere within a script file or a batch job, and they are resolved at execution time.

If the command is successful, there is no response. If the command is invalid or refers to a variable that does not exist, the response is an error message.

START

The syntax is

```
START
```

This command tells the client to connect to the server and process all commands previously sent. It can be used in testing a remote LPR host. There are no parameters.

You must issue SET HOST and SET PRINTER before issuing this command. If you want to print a job rather than test the host, use the PRINT command.

SET ALIGN

The syntax is

```
SET ALIGN = line_num
```

This command is for FCB printing only. It allows you to position the starting print line at a location other than line number 1. The default is 1, which means that printing starts on the first line.

See the section “[Using an FCB to Control Printing](#)” (page 4-69) for more information on FCB-based printing.

The variable is described in the following table:

Variable	Description
<i>line_num</i>	The line number at which printing is to start. The largest value permitted is the largest line number defined in the FCB.

The anticipated response is shown in the following example:

```
SET ALIGN=2
SET has completed
Ready:
```

SET ASA

The syntax is

```
SET ASA = ON|OFF
```

When set to ON or YES, the file is printed in ASA (or Fortran) mode. The individual print lines from a POWER file are transmitted to the LPD with the ASA carriage control codes intact. (The codes are not interpreted and converted to CR/LF sequences.) The lines are terminated by the SKIP1 and EOL values. Unrelated formatting options are bypassed. For example, SET CC=ON automatically becomes SET CC=ASA because SET CC=ON is for non-ASA output. The default is OFF.

A special “ASA PRINT” command is included in the control file. This means that the file must be sent to an LPD that supports this mode, such as the LPD provided by TCP/IP FOR VSE.

The anticipated response is shown in the following example:

```
SET ASA=ON
SET has completed
Ready:
```

SET BADPDF

The syntax is

```
SET BADPDF = [ CONTINUE | FAIL ]
```

This command specifies whether to continue sending PDF data if the PDF generator returns a bad code when it checks for the file’s length in the first pass. The default, *FAIL*, is to terminate the job before sending the data. This setting does not apply to printer types that only do a single pass.

The anticipated response is shown in the following example:

```
SET BADPDF=CONTINUE
SET has completed
Ready:
```

SET BANNER

The syntax is

```
SET BANNER = [ ON | OFF ]
```

This command tells the LPD that you want a banner page printed before the first page of the file. A byte is set in the control file to indicate this setting.

The anticipated response is shown in the following example:

```
SET BANNER=ON
SET has completed
Ready:
```

SET BLANK

The syntax is

```
SET BLANK = string
```

This command specifies the string to be sent to insert a blank line. It is similar to SET SKIP0

The variable is described in the following table:

Variable	Description
<i>string</i>	A hexadecimal string up to 50 bytes long. Coding a value of <i>NULL</i> or '*' resets the value to a null string. The default is 20 (one ASCII blank).

The anticipated response is shown in the following example:

```
SET BLANK=NULL
SET has completed
Ready:
```

SET CC

The syntax is

```
SET CC = ON|RAW|MCC|OFF
```

This command tells the LPR client whether you want your file to print with standard carriage control.

The parameters have the following meanings:

- *ON* or *ASA* (the default) turns standard carriage control (CC) on. When CC is on, the LPD examines the first character of each output line. Depending on the first character, LPR may add data to the output stream. The following table shows the specific actions LPR may take.

Character	Effect
1	If an FCB is in effect, a skip to channel 1 is performed. Otherwise, the value specified for TOF is inserted.
0	If an FCB is in effect, two lines are skipped. Otherwise, the value specified for SKIP2 is inserted.
- (a hyphen)	If an FCB is in effect, three lines are skipped. Otherwise, the value specified for SKIP3 is inserted.
+	If an FCB is in effect, an overstrike occurs. Otherwise, the value specified for SKIP0 is inserted.

- *RAW* or *ASIS* returns the data unformatted as it exists in the POWER queue. No formatting occurs (FCB use is not allowed).
- *MCC* is similar to *RAW*, but formatting is permitted (FCB use is allowed).
- *OFF* or *NO* turns standard carriage control off.

The AUTOLPR defaults are as follows:

- *YES₁* if a LST queue event triggered AUTOLPR.
- *NO₂* if a PUN queue event triggered AUTOLPR.

The anticipated response is shown in the following example:

```
SET CC=OFF
SET has completed
Ready:
```

SET CHANNEL_x

The syntax is

```
SET CHANNELx = string
```

This command is for FCB-based printing only. It specifies a control string to position the print line to the designated channel (denoted by *x* in the command name). When a “skip to channel_x” carriage control is encountered, normal FCB processing moves the printer to the correct line by using the appropriate number of CR/LF pairs. Once the correct line is reached, any value specified by the appropriate “SET CHANNEL_x=” command is transmitted. Note: The CR/LFs can be suppressed by also specifying SET CHANNELONLY=ON.

If this command is omitted, then line skipping is performed by inserting the character string specified by SET EXTRA=.

The variables are described in the following table:

Variable	Description
<i>x</i>	The channel identifier: 1–9, A–C.
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a ‘*’ resets the value to the default (NULL).

The anticipated response is shown in the following example:

```
SET CHANNEL2=0C0D
SET has_completed
Ready:
```

**SET
CHANNELONLY**

The syntax is

```
SET CHANNELONLY = ON|OFF
```

This command is for FCB-based printing only. When processing a “skip to channel x ” carriage-control code, normal processing positions the form by transmitting a series of CR/LF characters, followed by the value specified by the “SET CHANNEL x =” command. If you specify CHANNELONLY=ON (or YES), then the CR/LF positioning characters are omitted and only the “CHANNEL x =” value is sent. The default is OFF.

For example, if line-count positioning does not cause channel 1 to move to the top of the next form, you may want to code the following commands:

```
SET CHANNELONLY=OFF
SET CHANNEL1=0C0D
```

The anticipated response is shown in the following example:

```
SET CHANNELONLY=YES
SET has completed
Ready:
```

SET CLASS

The syntax is

```
SET CLASS = classname
```

This command attempts to set the LPD output class. Each LPD responds to this request in its own way. Only some LPDs actually support the CLASS parameter. RFC 1179 states that the CLASS parameter is used for the banner page.

The variable *classname* is a 1- to 16-character arbitrary class for this output. The first 16 characters beginning with the first non-blank character after the equal sign are used for the class name.

The following table shows which modes and defaults apply to this command:

LPR Client Type	Command Status	Default
CICS	Accepted	None
AUTOLPR	Accepted	Taken from the CLASS parameter on the * \$\$ LST statement that initiated the AUTOLPR
Batch	Accepted	None

The anticipated response is shown in the following example:

```
SET CLASS=A
SET has completed
Ready:
```

SET CONTROL

The syntax is

```
SET CONTROL = MANUAL | AUTO
```

When set to MANUAL, this command prevents LPR from generating control file information except as provided through SET DYNCONTROL= commands. You then must use SET DYNCONTROL to specify the control file's commands and settings explicitly. (An empty control file is sent otherwise.) The default setting is AUTO, which allows LPR to insert control information automatically.

For testing, you can use the DIAGNOSE LPR command to create a dump of the control file when it is sent. See the *TCP/IP FOR VSE Command Reference* for information on the DIAGNOSE command.

The anticipated response is shown in the following example:

```
SET CONTROL=MANUAL
SET has completed
Ready:
```

SET COPIES

The syntax is

```
SET COPIES = number
```

This command enables you to specify how many copies of a particular file you want to print. Note that LPR specifications do not provide for multiple copies. TCP/IP FOR VSE prints the file the requested number of times by looping through the print routine. Therefore, it is possible that other output on the printer may be interspersed between the copies. If you need to print multiple copies, consider using an INSERTS phase such as the INSCOPY2 sample. This sample INSERTS phase uses PCL to request that the printer print multiple copies of the file without involving VSE.

See also **SET PAUSE** and **SET REPEAT**.

The variable *number* is the number of copies you want to print.

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Taken from the COPIES parameter on the * \$\$ LST statement that initiated AUTOLPR
Batch	None

The anticipated response is shown in the following example:

```
SET COPIES=2
SET has completed
Ready:
```

SET CRLF

The syntax is

```
SET CRLF = YES|NO|UNIX
```

This command provides a shorthand method for setting the values of formatting variables. Once specified, the individual settings may be overridden.

The parameters have the following meanings:

- *YES* or *ON* (the default) sets SKIP0, SKIP1, SKIP2, SKIP3, TOF, SOD, EOD, and EOL to their default values.
- *NO* or *OFF* sets SKIP0, SKIP1, SKIP2, SKIP3, TOF, SOD, EOD, and EOL to a null string.
- *UNIX* sets the following values:

```
SKIP0=0D
SKIP1=0A
SKIP2=0A0A
SKIP3=0A0A0A
TOF=0C
SOD=null
EOD=1A
EOL=null
```

If SET CRLF is not issued, all values retain their default settings unless they are otherwise modified.

The anticipated response is shown in the following example:

```
SET CRLF=NO
SKIP0 now set to NULL
SKIP1 now set to NULL
SKIP2 now set to NULL
SKIP3 now set to NULL
TOF now set to NULL
SOD now set to NULL
EOD now set to NULL
EOL now set to NULL
SET has completed
Ready:
```

SET DATAFILE

The syntax is

```
SET DATAFILE = FIRST | LAST | ONLY
```

This command enables you to specify the order in which the data file is sent to an LPD. While most LPDs want the data file to arrive before the control file, you can change the order, or eliminate sending the control file altogether.

The parameters have the following meanings:

- *FIRST* (the default): The data file is sent before the control file.
- *LAST*: The data file is sent after the control file.
- *ONLY*: The data file is sent without a control file.

See also **SET SENLENGTH**.

The anticipated response is shown in the following example:

```
SET DATAFILE=ONLY
SET has completed
Ready:
```

SET DEBUG

The syntax is

```
SET DEBUG = ON | OFF
```

This command causes the internal LPR client to output a large number of messages to the console to help with debugging a problem with the LPR job. The default is OFF. It generates the same debugging messages as DIAGNOSE LPR, but the messages are enabled only for this job.

The anticipated response is shown in the following example:

```
SET DEBUG=ON
SET has completed
Ready:
```

**SET
DEFAULT_INSERTS**

The syntax is

```
SET DEFAULT_INSERTS = phasename
```

This command performs the same function as SET INSERTS, but the phase you specify is used only when SET INSERTS is not issued.

The following table describes the variable:

Variable	Description
<i>phasename</i>	The name of a default phase that is built with the INSERTS macro. If the name you specify cannot be loaded in the TCP/IP FOR VSE partition, you receive an error message. If the name you specify can be loaded but is not an INSERTS phase, the results are unpredictable. There is no default.

The anticipated response is shown in the following example:

```
SET INSERTS=HP4LJ
SET has completed
Ready:
```

SET DELREQ

The syntax is

```
SET DELREQ = ON|OFF
```

This command enables you to specify whether your file is to be deleted from the print queue on the foreign host. Note that this command has no control over the VSE disposition of the file. The foreign line printer daemon must support the use of SET DELREQ.

The parameters have the following meanings:

- *ON* or *YES* (the default) requests that the file be deleted from the foreign system.
- *OFF* or *NO* requests that the file be retained on the foreign system.

This command is accepted by all LPR client types.

The anticipated response is shown in the following example:

```
SET DELREQ=OFF
SET has completed
Ready:
```

SET DISP

The syntax is

```
SET DISP = KEEP|RESPECT|DELETE|HOLD
```

This command enables you to specify the disposition of a POWER LST or PUN queue entry after the PRINT command executes. If you are printing anything other than POWER queue entries, the command is ignored.

The following tables (for batch LPR and AUTOLPR) show the resulting disposition for each command setting and original disposition.

Batch LPR Command Setting	Resulting Disposition			
	Orig = K	Orig = D	Orig = H	Orig = L
SET DISP=HOLD	L	H	H	L
SET DISP=KEEP	K	D	H	L
SET DISP=RESPECT	L	Deleted	H	L
SET DISP=DELETE	L	Deleted	H	L

AUTOLPR Command Setting	Resulting Disposition			
	Orig = K	Orig = D	Orig = H	Orig = L
SET DISP=HOLD	L	Deleted	N/A	N/A
SET DISP=KEEP	L	Deleted	N/A	N/A
SET DISP=RESPECT	L	Deleted	N/A	N/A
SET DISP=DELETE	L	Deleted	N/A	N/A

The defaults for this command are as follows:

LPR Client Type	Default
CICS	<i>KEEP</i>
AUTOLPR	<i>HOLD</i> for DISP = K, <i>DELETE</i> for DISP = D
Batch	<i>KEEP</i>

SET DYNCONTROL

The syntax is

```
SET DYNCONTROL = variable
```

This command allows you to add control strings to the control file that is passed to an LPD. This permits adding various options and settings that are not provided for in the LPR/LPD RFCs. An additional string is added each time this command is issued. You can specify up to 85 characters of control information per command use. The aggregate length limit is 1024 bytes. The case of the operands is maintained, and they are translated to ASCII automatically. Each inserted string is terminated by an X'0A' (LF) as required by the LPR protocol.

If you use SET CONTROL = MANUAL, you must use SET DYNCONTROL commands to specify the contents of the control file explicitly. Otherwise, an empty control file is sent.

For example, to pass CLASS information in a control file, you would code the following statements:

```
SETVAR WORKVAR="-opassthru=class="+&PWRCLAS
SET DYNCONTROL=&WORKVAR
```

Note: "&PWRCLAS" is set automatically only under AutoLPR.

To examine the control file's contents, you can use DIAGNOSE LPR to produce a dump of the control file when it is sent.

This command is accepted by all LPR client types.

The anticipated response is shown in the following example:

```
SET DYNCONTROL=&VAR1
SET has completed
Ready:
```

SET EOD

The syntax is

```
SET EOD = string
```

This command sets the end-of-data (EOD) indicator. This hexadecimal string is sent to the printer following the final character in the data file. If an INSERTS phase is being used, any “END OF REPORT” data is sent following the EOD value.

See also **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0D0A).

The anticipated response is shown in the following example:

```
SET EOD=NULL
SET has completed
Ready:
```

SET EOF

The syntax is

```
SET EOF = string
```

This command specifies the end-of-file (EOF) indicator.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default ('/+').

The anticipated response is shown in the following example:

```
SET EOF=NULL
SET has completed
Ready:
```

SET EOL

The syntax is

```
SET EOL = string
```

This command specifies the end-of-line (EOL) string to be transmitted following each print line.

See also **SET ASA** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (NULL).

The anticipated response is shown in the following example:

```
SET EOL=0A
SET has completed
Ready:
```

SET EXTRA

The syntax is

```
SET EXTRA = string
```

This command is for FCB-based printing only. It specifies the string used to skip extra lines when positioning the line according to an FCB. When a “skip to channelx” carriage control is encountered, normal FCB processing moves the printer to the correct line by using the appropriate number of CR/LF (X'0D0A') pairs. Except for the first carriage return (X'0D'), which positions the printer to the left margin, they are redundant; only the line feeds (X'0A') are required. “SET EXTRA=” sets the string to be sent for each required line skip following the first.

See also **SET CHANNELx**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0A).

The anticipated response is shown in the following example:

```
SET EXTRA=NULL
SET has completed
Ready:
```

SET FCB

The syntax is

```
SET FCB = phasename
```

This command specifies the name of a phase to be used as the Forms Control Buffer (FCB). The FCB replaces the paper tape used by early model line printers and contains information for the line number associated with each carriage control character. If this parameter is omitted, limited carriage control emulation is performed.

Note: Carriage control (SET CC) must be set to ON or MCC before you use this command. Otherwise, the phase you specify is not processed.

See also **SET FCBPREFIX**.

The values for the parameter are described in the following table:

Value	Description
<i>phasename</i>	This is the name of a valid FCB phase as defined to VSE. If the name you specify cannot be loaded in the TCP/IP FOR VSE partition, you receive an error message. If the name you specify can be loaded but is not an FCB, the results are unpredictable and probably incorrect. Specify *NULL to delete previously specified FCB values.

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Taken from the FCB parameter on the * \$\$ LST statement that initiated AUTOLPR
Batch	None

The anticipated response is shown in the following example:

```
SET FCB=PRT1FCB
SET has completed
Ready:
```

SET FCBPREFIX

The syntax is

```
SET FCBPREFIX = value
```

This command sets the value of the “\$\$\$\$” prefix in an FCB name. It generally is used only for event (AUTOLPR) processing. The *value* is a four-character string.

When an FCB is specified on a JECL statement, the actual printer to be used may not be known. To allow for this, the programmer can code the eight-character FCB name with the first four positions set to “\$\$\$\$.” When POWER later determines the printer name, it substitutes a printer-specific string for “\$\$\$\$.” In this way, the derived FCB name is one that is appropriate for both the report and the printer.

When such a report is processed as an LPR event, the generic FCB specification is not resolved by POWER. But if a printer-specific string has been supplied by the SET FCBPREFIX command, then LPR processing can resolve the FCB name. An FCB image suitable for LPR processing—one that matches the report’s requirements with the destination’s attributes—is then loaded.

SET FILENAME

The syntax is

```
SET FILENAME = name
```

This command allows you to assign a different name to the file being sent to the LPD. By default, the file's original name is used. You can use this command to control the name that the LPD uses.

The variable is described in the following table:

Variable	Description
<i>name</i>	A 1- to 44-byte text string.

The anticipated response is shown in the following example:

```
SET FILENAME=JULY_PROJECT_RESULTS
SET has completed
Ready:
```

SET FNO

The syntax is

```
SET FNO = number
```

This command specifies a forms number to be included in the control file as the "-ofoms *number*" control. By default, no value is sent.

See also **SET OPTION**.

The anticipated response is shown in the following example:

```
SET FNO=112
SET has completed
Ready:
```

SET FORMDEF

The syntax is

```
SET FORMDEF = name
```

This command allows you to include a FORMDEF value in the control file when this value is needed by the LPR server/LPD host. This command may be used to supply the appropriate FORMDEF information because TCP/IP FOR VSE does not retrieve it from VSE. You can use this command along with SET PAGEDEF and SET INFOPRINT.

See also **SET OPTION**.

The variable is described in the following table:

Variable	Description
<i>name</i>	A 1- to 8-byte string. There is no default.

The anticipated response is shown in the following example:

```
SET FORMDEF=FName1
SET has completed
Ready:
```

SET HOST, SET IP, SET IPADDR

The syntax is

```
SET HOST = ipaddress|ipname
```

SET HOST, SET IP, and SET IPADDR are synonyms. These commands specify the IP address (IP name) of the host on which the LPD is running. You must specify this value and a printer name to print a file.

The variables are described in the following table:

Variable	Description
<i>ipaddress</i>	The IP address of the LPD
<i>ipname</i>	The domain name that represents the IP address

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Taken from the * \$\$ LST or * \$\$ PUN statement as defined by the HOSTNAME parameter on the DEFINE EVENT command
Batch	None

The anticipated response is shown in the following example:

```
SET HOST=192.168.0.7
VM.PROD
SET has completed
Ready:
```

The second line of the response varies depending on whether you specify *ipaddress* or *ipname*. If you specify *ipaddress*, the associated IP name, if applicable, is returned. If you specify *ipname*, the associated IP address is returned.

SET INFOPRINT

The syntax is

```
SET INFOPRINT = ON|OFF
```

This command allows you to send an InfoPrint-formatted job to an InfoPrint printer. Specifying *ON* causes LPR to ignore all other formatting options. Each line in the file is read and transmitted without any checks or modifications. This command also issues SET EOD=NULL and SET CC=RAW.

The QUERY OPTIONS command may be used to see the actual options in effect.

The destination host must be an IBM InfoPrint Manager server. Using a different LPD that does not understand the InfoPrint format may cause the file to be modified in an unacceptable manner.

The SET FORMDEF and SET PAGEDEF commands may be used to supply the FORMDEF and the PAGEDEF values, respectively, when they are required by the InfoPrint Manager server.

This command is accepted by all LPR client types. The default is *OFF*.

The anticipated response is shown in the following example:

```
SET INFOPRINT=ON
EOD now set to NULL
CC now set to ASIS (RAW)
SET has completed
Ready:
```

SET INSERTS

The syntax is

```
SET INSERTS = phasename
```

This command identifies a phase whose contents are to be transmitted along with the file. This enables you to include printer control data before the file, after the file, and after each form feed.

See the section [“Controlling Printers with the INSERTS Phase”](#) (page 4-71) for more information about using the INSERTS facility.

The variable is described in the following table:

Variable	Description
<i>phasename</i>	The name of a phase that is built with the INSERTS macro. If the name you specify cannot be loaded in the TCP/IP FOR VSE partition, you receive an error message. If the name you specify can be loaded but is not an INSERTS phase, the results are unpredictable and probably incorrect.

The defaults for this command are as follows:

LPR Client Type	Default
CICS	The name specified by SET DEFAULT_INSERTS. Otherwise, none.
AUTOLPR	Taken from the UCS parameter on the * \$\$ LST statement that initiated the AUTOLPR.
Batch	The name specified by SET DEFAULT_INSERTS. Otherwise, none.

The anticipated response is shown in the following example:

```
SET INSERTS=HP4LJ
SET has completed
Ready:
```

SET JOBNAME

The syntax is

```
SET JOBNAME = jobname
```

This command enables you to specify a job name that is printed on separator pages created by the printing host. See the documentation for your LPD to determine whether it supports printing this name.

The variable *jobname* is a 1- to 16-character arbitrary name for this output. You can specify a value from 1 to 99. The first 16 characters beginning with the first non-blank character after the equal sign are used for *jobname*.

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Taken from the JOBNAME parameter on the * \$\$ JOB statement that initiated AUTOLPR
Batch	None

The anticipated response is shown in the following example:

```
SET JOBNAME=VSEUSER
SET has completed
Ready:
```

SET JSEPCOUNT

The syntax is

```
SET JSEPCOUNT = ON|OFF|count
SET JSEP = ON|OFF|count
```

This command specifies how many POWER-generated separator pages are to be requested for this job. This setting applies only to POWER reports.

Note: You must use the SEPARATOR_PAGES command to enable POWER-generated separator pages for LPR. Otherwise, SET JSEPCOUNT has no effect. See the *TCP/IP FOR VSE Command Reference* for more information on the SEPARATOR_PAGES command.

The parameters have the following meanings:

- *ON* or *YES* (the default) specifies a JSEP value of 1 (one separator page).
- *OFF* or *NO* suppresses all JSEP pages.

- *count* is the number of separator pages to be requested. Valid values range from 0 to 9.

The anticipated response is shown in the following example:

```
SET JSEP=2
SET has completed
Ready:
```

SET LASTCOLUMN

The syntax is

```
SET LASTCOLUMN = number
```

This command specifies the last column of command strings from the client that you want LPR to read and process. (This command does not affect data being sent to an LPD.) Any command data that may be in column numbers greater than the LASTCOLUMN value are ignored. Setting LASTCOLUMN to a number greater than the default allows sequence numbers or other data in the higher-numbered columns to be read.

The variable is described in the following table:

Variable	Description
<i>number</i>	The last data column to be read. The default is 72. The maximum is 80.

The anticipated response is shown in the following example:

```
SET LASTCOLUMN=80
SET has completed
Ready:
```

SET LEAVEBLANK

The syntax is

```
SET LEAVEBLANK = ON|OFF
```

This command specifies whether to truncate trailing blanks. A value of *OFF* disables truncation and accommodates printers and software that require blank retention. The default is *OFF*.

SET NOEJECT

The syntax is

```
SET NOEJECT = ON|OFF
```

This command enables you to suppress the initial form feed character at the beginning of a listing. Many print files begin with a page eject character. TCP/IP FOR VSE normally translates this character into a form feed. On some printers, however, this creates a blank page.

The parameters have the following meanings:

- *ON* or *YES* suppresses the initial form feed character.
- *OFF* or *NO* (the default) does not suppress the initial form feed character.

This command is accepted by all LPR client types. The anticipated response is shown in the following example:

```
SET NOEJECT=ON
SET has completed
Ready:
```

SET OPTION

The syntax is

```
SET OPTION = ON|OFF
```

This command controls whether a generic option (“-o *option*”) specified by the LPR/LPD protocol is added to the control file. This option is needed to include the values for SET FORMDEF, SET PAGEDEF, and SET FNU. Use SET OPTION = OFF if an LPD does not accept this option. The default is ON.

The anticipated response is shown in the following example:

```
SET OPTION=OFF
SET has completed
Ready:
```

SET PAGEDEF

The syntax is

```
SET PAGEDEF = name
```

This command allows you to include a PAGEDEF value in the control file when this value is needed by the LPR server/LPD host. This command can be used to supply the appropriate PAGEDEF information because TCP/IP FOR VSE does not retrieve it from VSE. You can use this command along with SET FORMDEF and SET INFOPRINT.

See also **SET OPTION**.

The variable is described in the following table:

Variable	Description
<i>name</i>	A one- to eight-byte string. There is no default.

The anticipated response is shown in the following example:

```
SET PAGEDEF=PName1
SET has completed
Ready:
```

SET PAUSE

The syntax is

```
SET PAUSE = delay
```

This command sets the delay between copies. When multiple copies of a document are requested, the LPR client causes printing to pause after each copy. This delay is needed by some printers. The delay you specify must be appropriate for the printers you use.

See also **SET COPIES** and **SET REPEAT**.

The variable is described in the following table:

Variable	Description
<i>delay</i>	The number of seconds from 0 to 9. The default is 2.

The anticipated response is shown in the following example:

```
SET PAUSE=3
SET has completed
Ready:
```

SET PCL5

The syntax is

```
SET PCL5 = ON|OFF
```

This command causes the LPR client to suppress all formatting and translation options when sending the file. Each line in the file is read and transmitted without any checks or changes. This command also issues SET EOD=NULL and SET CC=RAW. The default is OFF.

This command is required for PCL printing. See the section “[Controlling Printers with the INSERTS Phase](#)” (page 4-71) for more information on PCL printing.

The anticipated response is shown in the following example:

```
SET PCL5=ON
EOD now set to NULL
CC now set to ASIS (RAW)
SET has completed
Ready:
```

SET PDF

The syntax is

```
SET PDF = ON|OFF|member
```

This command specifies whether a document is to be converted to PDF. It also allows you to specify a custom configuration member to control the conversion. If the command is set to ON, the default member “PDFSETUP” is used. This member is in the VSE library. The default is OFF.

See [Chapter 8, “PDF Conversion Facility.”](#) for more information on converting files to PDF and creating a custom PDF configuration member.

The variable is described in the following table:

Variable	Description
<i>member</i>	An “.L” member in the VSE library. (Omit the “.L” from the member name.) This member contains customized conversion parameter settings.

SET PRINTER

The syntax is

```
SET PRINTER = printer-name
SET PRT = printer-name
```

This command enables you to specify the printer you want to use. To run LPR, you must specify both a host name and a printer name.

The variable is described in the following table:

Variable	Description
<i>printer-name</i>	<p>A printer name that is valid to the LPD host that you specified in the SET HOST command. It can be up to 16 characters long. This name is sometimes referred to as the queue name.</p> <p>In general, the queue name is obvious from the LPD software. For most Windows daemons, the queue name is the name that you assign to the associated printer. For other LPDs, these names are not so clear. In VM, for example, the queue names are assigned through the LPD CONFIG file on the A disk for the LPSERVE virtual machine. For HP JetDirect cards, you can choose the queue names <i>TEXT</i> or <i>RAW</i>. This excerpt from the HP JetDirect manual (5967-9801) highlights the differences:</p> <p>The line printer daemon on the HP JetDirect print server treats data in the text queue as unformatted text or ASCII, and adds a carriage return to each line before sending it to the printer. (Note that the actual observed behavior is that a PCL line termination command (value of 2) is issued at the beginning of the job. This line termination command can be reset or changed within the job.) The line printer daemon treats data in the raw queue as formatted files in PCL, PostScript, or HP-GL/2 and sends the data without change to the printer.</p>

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	Taken from the DEST= parameter on the * \$\$ LST or * \$\$ PUN statement
Batch	None

The anticipated response is shown in the following example:

```
SET PRINTER=PR1
SET has completed
Ready:
```

Note that any printer name is accepted. If the name you specify is invalid (not defined to the LPD), you are not informed until you attempt to query the print queues or print a document.

SET RECFM, SET BLKSIZE, SET LRECL

The syntax is

```
SET RECFM=recfm
SET BLKSIZE=blksize
SET LRECL=lrecl
```

These commands allow you to specify characteristics of the file that is to be printed.

The variables have the following meanings:

- *recfm* is the record format of the file.
- *blksize* is the block size used by the file.
- *lrecl* is the logical record length of the records in the file.

The values you choose for these variables depend on whether you are printing

- Sequential disk files and VSAM-managed SAM files
- POWER files
- VSAM or VSAMCAT files
- ICCF or BIM-EDIT files
- Library files.

For **sequential disk files and VSAM-managed SAM files**:

- Fixed-length records are padded when necessary. When padding occurs, text files are padded with blanks.
- You can process VSAM-managed SAM files as VSAM ESDS files, and it is more efficient to do so because the SAM emulation routines call VSAM anyway.

- The appropriate values are as follows:

recfm	lrecl	blksize
F	record size	N/A
FB	record size	record size times blocking factor
V	maximum record size	N/A
VB	maximum record size	maximum block size
SU	maximum record size	N/A
SB	maximum record size	maximum block size

For **POWER files**:

- Fixed-length records are padded if necessary when writing to the POWER spool. When padding occurs, text files are padded with blanks, and binary files are padded with zeros.
- The minimum *lrecl* for POWER RDR queue files is 80 and the maximum is 128.
- The minimum *lrecl* for POWER LST queue files is 1 and the maximum is 32766.
- The *lrecl* for POWER PUN queue files must be 80.
- The appropriate values are as follows:

recfm	lrecl	blksize
F	record size	N/A
V	maximum record size	N/A

For **VSAM/VSAMCAT files**, the appropriate values are as follows:

recfm	lrecl	blksize
F	record size	N/A
V	maximum record size	N/A

For **ICCF/BIM-EDIT files**:

- The files always contain 80-byte records, regardless of specification.
- The appropriate values are as follows:

recfm	lrecl	blksize
F	N/A	N/A

For **Library files**:

- Library members always contain fixed 80-byte records or a string file consisting of a single string of bytes.
- The appropriate values are as follows:

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	80	N/A
SV	N/A	N/A
S	N/A	N/A

The defaults for these commands are *RECFM=FB*, *BLKSIZE=133*, *LRECL=133*.

The anticipated responses are shown in the following example:

```
SET RECFM=FB
SET has completed
Ready:
SET LRECL=133
SET has completed
Ready:
SET BLKSIZE=133
SET has completed
Ready:
```

SET REPEAT

The syntax is

```
SET REPEAT = LPD|VSE
```

This command enables you to select whether the LPR client or the Line Printer Daemon (LPD) is responsible for processing multiple-copy requests.

If you specify *LPD*, TCP/IP FOR VSE requests multiple copies from the LPD. Some LPDs, however, do not honor this request. If you specify *VSE*, TCP/IP FOR VSE generates multiple copies of the data stream. Specifying *VSE* always works, but the disadvantage is that it increases VSE overhead because the data stream is generated and transmitted multiple times. The default is *LPD*.

See also **SET COPIES** and **SET PAUSE**.

The anticipated response is shown in the following example:

```
SET REPEAT=VSE
SET has completed
Ready:
```

**SET RPORT,
SET LPORT**

The syntax is

```
SET RPORT = remote-port  
SET LPORT = local-port
```

These commands are used to set the remote TCP port that LPR connects to on the foreign host (SET RPORT) and the local TCP port that LPR uses on VSE (SET LPORT). You should use these commands only when you are directed to do so by CSI Technical Support.

The variables have the following meanings:

- *remote-port* is the remote TCP port that LPR connects to on the foreign host. You must ensure that an LPD is listening on the port you specify. The default is *515*.
- *local-port* is the local TCP port that LPR uses on VSE. You must ensure that the port you select is acceptable to the foreign LPD. If it is not, the connection request is rejected. The default is *721*.

The anticipated response is shown in the following example:

```
SET RPORT=515  
SET has completed  
Ready:
```

SET SENDLENGTH

The syntax is

```
SET SENDLENGTH = ON|OFF
```

This command specifies whether you want to send the file's length before sending before the file itself. In the LPR process, files that are sent can be of any length, and there is no way to know when a file has been completely received. Therefore, by default, each file is preceded by a command that reports the file's total length. The LPD then receives the file for the specified byte count, responds with an acknowledgment byte, and continues with the next command. The biggest drawback to this process is that TCP/IP FOR VSE must read and format each file twice—the first time simply to determine its length.

If the LPD supports it, and if the data file is to be sent last, then SENDLENGTH can be set to OFF. In this mode, TCP/IP FOR VSE reports a file length of 0 before sending the file. In this mode, the LPD considers all data up to connection close to be part of the data file. This method of sending a file is more efficient, but a broken connection is considered the end-of-file.

See also **SET DATAFILE**.

The anticipated response is shown in the following example:

```
SET SENDLENGTH=OFF
SET has completed
Ready:
```

SET SKIP0

The syntax is

```
SET SKIP0 = string
```

This command sets the string to be transmitted before an overstrike line (carriage control: '+').

See also **SET CC** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0D).

SET SKIP1

The syntax is

```
SET SKIP1 = string
```

This command sets the string to be transmitted before a single-spaced line (carriage control: ' ').

See also **SET ASA** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0D0A).

SET SKIP2

The syntax is

```
SET SKIP2 = string
```

This command sets the string to be transmitted before a double-spaced line (carriage control: '0').

See also **SET CC** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0D0A0A).

SET SKIP3

The syntax is

```
SET SKIP3 = string
```

This command sets the string to be transmitted before a triple-spaced line (carriage control: '-').

See also **SET CC** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (0D0A0A0A).

SET SOD

The syntax is

```
SET SOD = string
```

This command sets the start-of-data (SOD) indicator. This hexadecimal string is sent to the printer in advance of the first data byte. If an INSERTS phase is being used, any "START OF REPORT" data is sent following the SOD value.

See also **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a '*' resets the value to the default (NULL).

The anticipated response is shown in the following example:

```
SET SOD=NULL
SET has completed
Ready:
```

SET SOSI

The syntax is

```
SET SOSI = CONVERT | KEEP | XLATE | BLANK | NONE
```

This command controls the shift-in, shift-out (SOSI) processing of double-byte character sets (DBCS).

The valid values are described in the following table:

Value	Meaning
CONVERT	(The default.) Indicates that the data stream is converted and SO/SI characters are added or removed as appropriate.
KEEP	SO/SI characters are retained as placeholders in the ASCII data stream.
XLATE	SO/SI characters are retained as placeholders in the ASCII stream, but they are translated to their ASCII equivalents of X'1E' and X'1F'.
BLANK	SO/SI characters are retained as placeholders in the ASCII stream, but they are translated to ASCII spaces of X'20'.
NONE	SO/SI characters are not present in the EBCDIC data. Every pair of bytes is treated as a DBCS. This mode is referred to as the graphic character set.

The anticipated response is shown in the following example:

```
SET SOSI=KEEP
SET has completed
Ready:
```

SET TITLE

The syntax is

```
SET TITLE = title
```

This command allows you to specify a string that prints on the title page of the output produced by the LPR client. See the documentation for your LPD to determine whether it supports this capability.

The variable *title* is the text that is to appear on the title page. The first 16 characters beginning with the first non-blank character after the equal sign are used for the title address. There is no default.

The anticipated response is shown in the following example:

```
SET TITLE=FROMVSEB
SET has completed
Ready:
```

SET TOF

The syntax is

```
SET TOF = string
```

This command specifies the string that is sent to position printing to top of form (TOF).

Note: FCB-based printing does not involve “top-of-form” processing. If an INSERTS phase is being used, any “PAGE” data is sent following the TOF value.

See also **SET CC** and **SET CRLF**.

The variable is described in the following table:

Variable	Description
<i>string</i>	1 to 50 hexadecimal pairs. Coding a <i>NULL</i> nullifies the string. Coding a ‘*’ resets the value to the default (0C0D).

The anticipated response is shown in the following example:

```
SET TOF=NULL
SET has completed
Ready:
```

SET TRANSLATE

The syntax is

```
SET TRANSLATE = table-name
```

This command specifies the name of a translate table to be used for EBCDIC-to-ASCII translation. Translation occurs when the VSE data to be printed is in EBCDIC and other hosts expect ASCII.

The variable is described in the following table:

Variable	Description
<i>table-name</i>	The name of a valid translate table. The specified table must be available to TCP/IP through the DEFINE TRANSLATIONS command as specified in the <i>TCP/IP FOR VSE Installation Guide</i> . You can use the QUERY TRANSLATIONS console command to obtain a list of valid translate tables. If you specify an invalid <i>table-name</i> , the EMAIL client prints a message and uses the default <i>table-name</i> . The EMAIL job does not terminate because of an invalid <i>table-name</i> .

The defaults for this command are as follows:

LPR Client Type	Default
CICS	None
AUTOLPR	None
Batch	Set by DEFINE TRANSLATION in the TCP/IP for VSE initialization deck. Otherwise, none.

The anticipated response is shown in the following example:

```
SET TRANSLATE=US_ENG_03
SET has completed
Ready:
```

**SET USER, SET
PASSWORD**

The syntax is

```
SET USER = tcpip-userid
SET PASSWORD = tcpip-password
```

These commands identify the user to the TCP/IP FOR VSE security system. If the system administrator has activated TCP/IP security, you must use a valid user ID and password to access the TCP/IP FOR VSE file system. This is useful for limiting a user's access to specific file locations, as defined by the security exit.

The variables have the following meanings:

- *tcpip-userid* is a valid user ID that is known to the TCP/IP FOR VSE security system. This string can be up to 31 characters long. If you specify an invalid user ID, you are not notified until you attempt to access the file system. A valid user ID is not required to use the SHORT or LONG commands to query printers on foreign hosts.
- *tcpip-password* is a valid password that is known to the TCP/IP FOR VSE security system. The password can be up to 16 characters long.

The defaults for *tcpip-userid* are as follows:

LPR Client Type	Default
CICS	\$LPR
AUTOLPR	<i>AUTOLPR</i>
Batch	\$LPR

The anticipated responses are shown in the following example:

```
SET USER=MARC
SET has completed
Ready:
SET PASSWORD=MARCz1
SET has completed
Ready:
```

SET VALIDATE

The syntax is

```
SET VALIDATE = ON|OFF
```

The SET VALIDATE command specifies whether to perform an initial translation of unprintable characters to blanks. All characters below X'40' and above X'F9' are converted to blanks. This action prevents unprintable characters from affecting an ASCII printer. The default is *OFF*.

The anticipated response is shown in the following example:

```
SET VALIDATE=ON  
SET has completed  
Ready:
```

Using LPR with InfoPrint Printers

InfoPrint-formatted jobs may be sent to an IBM InfoPrint printer. To do this, you must specify the following LPR SET commands. These commands either supply or disable certain formatting controls.

- SET INFOPRINT
- SET PAGEDEF (optional)
- SET FORMDEF (optional)

See the SET command descriptions earlier in this chapter for more information on these commands.

The following example shows how a batch LPR job may be used to send a file to an InfoPrint printer. You must substitute user-provided values for all settings except SET INFOPRINT=ON.

```
// EXEC CLIENT, PARM='APPL=LPR, ID=00'  
SET HOST=192.168.1.161  
SET PRINTER=INFOPRINT  
CD POWER.LST.ALL  
SET INFOPRINT=ON  
SET FORMDEF=FDEFSTD  
SET PAGEDEF=PDEFSTD  
PRINT INFOJOB1  
QUIT  
/*
```

Using an FCB to Control Printing

A Forms Control Buffer (FCB) controls the vertical movement of forms as they are printed. It controls channel skipping, line spacing, and printing at 6 and 8 lines per inch. With FCB formatting, each line's position on the page is represented by a line in the FCB. It is assumed that the printer is already positioned to top-of-form. As each text line is to be printed, its ASA code is interpreted and the FCB is examined to determine how many lines are to be skipped before printing starts.

In addition to using the standard '+', '-', '0', and ' ' (blank) carriage-control characters, you can specify channels 1 through 9 and A through C. Each line in the FCB may contain a flag for each channel. When a "skip to channel *x*" command (a hexadecimal string) is encountered, lines are skipped until the particular channel is located. By convention, channel 1 is the first printed line, channel 9 is middle of the form, and channel 12 is the last line of the form.

Normal FCB formatting adds only normal line-spacing controls to handle all skips. Specifically, a channel 1 skip is not a form feed; it is achieved by inserting single line spacing until both the form and the FCB wrap to the channel 1 position. This method of skipping lines must be used because there is no requirement that channel 1 be assigned as the top-of-form.

Problems arise when the physical printer is not configured to the correct number of lines per page. For example, if the FCB specifies 8 LPI, then a standard 8.5 × 11 form would contain exactly 88 lines. Many printers, however, can't print to the edge of the paper. This means that, with one-inch margins, each new page will start eight lines lower than the last. Conversely, if the printer uses a smaller font and prints more than 88 lines per page, FCB formatting can cause the printed text on pages to creep upward.

These problems are avoided when control strings are defined to move to specific channel positions. For example, issuing SET CHANNEL1=0C0D results in a skip to channel 1 and an actual form feed when the string 0C0D is encountered in a line.

The LRP client commands related to FCB formatting are described in the following table. See the section [“Using LPR Client Commands”](#) (page 4-17) for more information on these commands.

Command	Description
QUERY OPTIONS	Displays option values currently in effect.
SET ALIGN	Specifies the line at which printing should start at top of each page. This command allows you to set the starting alignment of your paper against the FCB. Normal processing assumes that when a report starts printing, the printer and FCB are both aligned to position 1. For example, if you have a report and an FCB that run on fanfold paper, operator alignment is to position the printer on the fold. The 8-LPI FCB specifies that channel 1 is on line 9. This causes the printer to space 8 lines from the fold, leaving a 1-inch margin at the top of each page. When converting this report to a network printer, the printer automatically starts printing 1 inch from the edge of the paper. By specifying SET ALIGN=9, the FCB is positioned to account for this predefined margin. This setting also has the effect of reducing the number of lines per page to allow for the same margin on each page.
SET CC	Specifies whether the file should be printed with standard carriage control turned on.
SET CHANNEL x	Specifies a hex string to position the print line to the x channel (1 through 9, A through C). The specified hexadecimal characters are added to the output stream when the corresponding channel skip is encountered. This string is emitted immediately before the printable text. One use of this command is to provide absolute positioning of text on a form.
SET CHANNELONLY	Specifies that channel positioning should be performed only by skipping lines and not by using the hex strings specified by SET CHANNEL x commands. For example, setting CHANNEL1=0C0D and CHANNELONLY=YES causes a channel 1 skip to omit the single-line spacing to the channel 1 position and generate a form feed instead.
SET EXTRA	Specifies the hex string used to skip extra lines when positioning the line according to an FCB (when a corresponding CHANNEL x string is not specified).
SET FCB	Specifies the name of a phase that is to be used as the FCB.
SET FCBPREFIX	Specifies the value of the “\$\$\$\$” prefix in an FCB name.

Controlling Printers with the INSERTS Phase

TCP/IP-attached printers are not equivalent to normal POWER printers. Most desktop printers do not handle EBCDIC-to-ASCII translation. In addition, they usually have capabilities that far exceed those of traditional mainframe units. This means that you often need to include initialization data along with the text to be printed.

In this section we discuss how to code and assemble an INSERTS macro. Using this macro, you can generate a phase containing PCL commands that control more aspects of your print operation than other LPR methods can control. After you code and assemble the macro and then generate a phase, you can use the SET INSERTS LPR command to load and execute this phase when LPR runs.

What is PCL?

You can access the advanced capabilities of PC printers through a command structure known as Printer Command Language (PCL). Different printers support different levels of PCL, but most support PCL5 or PCL6.

PCL was created by Hewlett-Packard, and you can find documentation for PCL commands on their website at www.hp.com. A typical printer command contains

- Escape Sequence
- Category of Command
- Value Field
- Termination Character

You can combine commands in PCL5 as long as they have the same command category.

PCL enables you to manipulate printers in many ways. For example, you can set

- Number of copies
 - Paper source and paper size
 - Page orientation, such as portrait or landscape
 - Font
-

Specifying PCL Commands

From VSE, you need a way to specify PCL commands to the printer, and you also need a way to transmit that data along with the print job. To do this, you use an INSERTS phase. You can associate an INSERTS phase with a print file either by using the SET INSERTS command or, for AUTOLPR, by coding it as the object of the UCS= parameter on the * \$\$ LST card.

When you code the macro that generates the INSERTS phase, you can specify that PCL commands be inserted

- Before the first byte of the print file
- Immediately after each form feed (this is not available if an FCB is specified)
- After the last byte of the print file

You must enter the ASCII equivalents of PCL commands in hexadecimal.

The syntax of the INSERTS macro is described in the next section.

Note: You must use the SET PCL5 LPR client command in your print job along with SET INSERTS when specifying an INSERTS phase containing PCL commands. The SET PCL5 command disables all LPR-generated formatting controls.

INSERTS Macro

The INSERTS macro is coded once to generate each INSERTS phase. The syntax is as follows:

<i>phase</i>	INSERTS DEFINE [,HEADER= <i>head</i>]	*
	[,TRAILER= <i>trail</i>]	*
	[,PAGE= <i>page</i>]	

The variables are as follows:

- *phase* is the name to be assigned to the INSERTS phase.
- *head* is a string of hexadecimal characters (0 through 9 and/or A through F) that is to be transmitted before the print file.
- *trail* is a string of hexadecimal characters (0 through 9 and/or A through F) that is to be transmitted after the print file.
- *page* is a string of hexadecimal characters (0 through 9 and/or A through F) that is to be transmitted after the form feed character at the beginning of each page. Note that the page data is suppressed for any file that has an associated FCB.

The maximum length of the HEADER, TRAILER, and PAGE strings is 256 bytes each. If you require more than 256 bytes, then you must manually define the INSERTS table that the macro normally generates. An example of this table is in the section “[INSERTS Example 2](#)” (page [4-74](#)).

INSERTS Example 1 The following INSERTS phase prints two copies of each output file.

```

* $$ JOB JNM=INSCOP2,CLASS=4,DISP=D
* $$ LST CLASS=A,DISP=D,DEST=(,VSEUSER)
// JOB INSCOP2
// LIBDEF *,SEARCH=(PRD2.TCPIP)
// LIBDEF PHASE,CATALOG=PRD2.TCIPCFG
// OPTION CATAL,LIST
// EXEC ASMA90
*****
*
* This inserts phase causes two copies of the output to be printed.*
*
* 1B45 = Printer Reset
* 1B266C3258 = Two Copies of Output (1B266C is the category ) *
* (32 is ASCII for 2 copies ) *
* (58 is the command code ) *
*
*
* A trailer of 1B45 resets the printer.
*
*****
INSCOPY2 INSERTS DEFINE,
                HEADER=1B451B266C3258,
                TRAILER=1B45
                END
/*
// EXEC LNKEDT
/&
* $$ EOJ

```

The sample *INSCOPY2.A* is in the TCP/IP FOR VSE installation library.

INSERTS Example 2 The following INSERTS phase puts the printer into landscape mode and prints the file in a semi-bold typeface.

```

* $$ JOB JNM=INSLAND,CLASS=4,DISP=D
* $$ LST CLASS=A,DISP=D,DEST=(,VSEUSER)
// JOB INSLAND
// LIBDEF *,SEARCH=(PRD2.TCPIP)
// LIBDEF PHASE,CATALOG=PRD2.TCPIPCFG
// OPTION CATAL,LIST
// EXEC ASMA90
*****
* This inserts phase causes the printer to shift into Landscape Mode*
* and to print the output in a semi-bold typeface. It is appropriate*
* for printing POWER output.
*
*
*
*
* 1B45 = Printer Reset
* 1B266C314F = Shift into Landscape (1B266C is the category ) *
* (31 is the parm ) *
* (4F is the command code) *
*
*
* 1B28733142 = Semi-Bold typeface (1B2873 is the category ) *
* (31 is the parm ) *
* (42 is the command code) *
*
*
* A trailer of 1B45 resets the printer.
*
*****
INSLAND INSERTS DEFINE, X
          HEADER=1B451B266C314F1B28733142, X
          TRAILER=1B45
          END
/*
// EXEC LNKEDT
/&
* $$ E0J

```

The sample *INSLAND.A* is in the TCP/IP FOR VSE installation library.

The generated table looks similar to the following example. This table can be manually defined to include large strings (>256 bytes).

INSLAND	CSECT		
INSHEADA	DC	A(HDR)	Address of Header
INSHEADL	DC	F'12'	Length of Header
INSTRLRA	DC	A(TRL)	Address of Trailer
INSTRRL	DC	F'2'	Length of Trailer
INSPAGEA	DC	A(PAG)	Address of Page
INSPAGEL	DC	F'0'	Length of Page
HDR	DS	0D	
	DC	X'1B451B266C314F1B28733142'	
TLR	DS	0D	
	DC	X'1B45'	
PAG	DS	0D	
	END		

Ping, Traceroute, DISCOVER Clients

Overview

This chapter describes diagnostic tools provided with TCP/IP FOR VSE:

- **Ping** is a feature of most TCP/IP implementations. It is a low-level mechanism that enables you to test your network connections and configuration. To ping a host, the client will send a short message to that domain, or IP address. The receiving host then returns a message response to the originator. Using this method, you can test network connections in a straightforward manner.
- **Traceroute** is another feature of most TCP/IP implementations that traces the path a packet must take to reach a destination. Ping and traceroute are similar in that both tell you if you have connectivity to the foreign system. Traceroute also tells you how the packet got there, showing you the IP address and the response time of each “hop.”
- The **DISCOVER** client can help you determine the maximum transmission unit (MTU) size between the VSE host and another host on your network. Like ping and traceroute, DISCOVER uses the ICMP protocol, but it does so with varying packet sizes to determine the best size for a specific destination.

The ping, traceroute, and DISCOVER utilities can be run programmatically, using a user-written SOCKET program such as a CICS transaction provided by CSI, or as a batch job using the CLIENT program.

Ping

Incoming Ping

TCP/IP FOR VSE includes the routines that enable you to reply to external ping requests. This function is self-contained within the stack and does not require any definition parameters to enable it. If your ping is not returned, then it is either because you either have a network connection problem, or because you have specified, within your security exit, to not respond to a specific IP address. (See the *TCP/IP FOR VSE Installation Guide*, chapter 9, “Security,” for more information.)

Ping reports the response time in milliseconds. By default, the following parameter is enabled, which will shows you every ping request coming to your stack:

```
SET PING_MESSAGE = ON
```

This command enables the TCP/IP FOR VSE console message that tells the operator that VSE has been pinged. It should be noted, however, that many network monitoring tools and other non-VSE clients may periodically issue a ping to collect information. The result can be a lot of message output to the console. It is recommended that you have set this value set to OFF if you see that you have a lot of ping traffic, and that these messages are normal for your environment.

Outgoing Ping

It often is useful to issue a ping from TCP/IP FOR VSE to a non-VSE host. This can help helping to determine the specific path that a datagram must travel through a TCP/IP network to reach its destination. We discuss these situations in the following sections.

DISCOVER

When you install TCP/IP FOR VSE, you must specify the MTU size for outgoing TCP connections. The easiest way to determine the MTU size is to refer to the chart in the *TCP/IP FOR VSE Installation Guide*, chapter 2, “Planning for Installation.” Unfortunately, IP implementations vary, so not everyone can use the value in the chart.

The DISCOVER client can help you determine the MTU size between the VSE host and another host on your network. To find the MTU size, the utility sends packets into the network starting with the smallest possible value (576) and incrementing by 16 with the *do not fragment* bit set ON in the packet header. The last successful transmission determines the MTU size for that connection.

When you configure TCP/IP FOR VSE, you can set the MTU size for each outgoing link. If you run the DISCOVER utility against multiple hosts that are all accessed with the same link, you may want to choose the smallest MTU size that is returned as the MTU size for the outgoing link for the best average performance across your network. Sometimes, however, this may not be the best choice if one destination gains at the cost of every other destination because the TCP/IP FOR VSE stack would now be sending smaller packets. So, if you do use the smallest returned size from DISCOVER, monitor the overall results for a period of time to determine whether this is the best choice for you.

Before you run DISCOVER, the MTU size should always be set at its **maximum** value as defined in the *TCP/IP FOR VSE Installation Guide*, chapter 2. The DISCOVER client never returns an MTU size that is greater than the currently set MTU size for the outgoing link.

The DISCOVER client is available interactively through CICS, through batch, and programmatically through the Sockets interface. For more information about the Sockets interface, see the *TCP/IP FOR VSE Programmer's Guide*.

Running CICS Clients

TCP/IP FOR VSE provides ping, traceroute, and DISCOVER clients that run as CICS transactions. Using these clients, you can direct ping, traceroute, and DISCOVER requests to any IP address on your network. For ping, if the TCP/IP stack at that address is functioning and supports ping, the stack echoes the request and you are notified.

Before you can use the CICS client, your CICS system programmer must install the client software on your CICS system. See the *TCP/IP FOR VSE Installation Guide* for instructions.

Note: By default, all TCP/IP FOR VSE clients use a stack identifier of "ID=00." When running batch utilities, you can modify this default by putting the matching stack value in the execution parameter. Because of how CICS operates, all CICS programs that run in the same partition communicate with the same stack. The stack identifier is set by including it in a TCP/IP FOR VSE OPTION SYSPARM statement in the CICS initialization JCL. Here is an example:

```
// OPTION SYSPARM=' 01'
```

Ping

The syntax of the ping command is

```
PING [nnn.nnn.nnn.nnn|symbolic-name]
```

There are no parameters. The variables have the following meanings:

- *nnn.nnn.nnn.nnn* is the IP address in dotted decimal notation.
- *symbolic-name* is the symbolic name of a TCP/IP host. The symbolic name is translated to an IP address by the local DNS client.

The following example shows two ping commands:

```
PING 100.50.50.90  
PING CSI-INTERNATIONAL.COM
```

The ping processor makes five attempts to reach the remote location. It lists the success or failure of each attempt and the time required for each response. This is also the format for issuing the PING command from the TCP/IP FOR VSE console interface.

Traceroute

The syntax of the traceroute command is

```
TRACERT [nnn.nnn.nnn.nnn | symbolic-name]
```

The variables have the following meanings:

- *nnn.nnn.nnn.nnn* is the IP address in dotted decimal notation.
- *symbolic-name* is the symbolic name of a TCP/IP host. The symbolic name is translated to an IP address by the local DNS client.

The following example shows two traceroute commands:

```
TRACERT 100.50.50.90  
TRACERT TCPIP4VSE.COM
```

The traceroute processor reports each hop on the way to the destination as well as the time it took to get to that hop. This is also the format for issuing the TRACERT command from the TCP/IP FOR VSE console interface.

The following example shows TRACERT output:

```
TRACERT YAHOO.COM  
TCP200I Client -- Startup --  
TCP207I Copyright (c) 1995-2006 Connectivity Systems Incorporated  
TCP202I Attempting to Establish Connection  
TCP204I Connection has been Established  
Client manager connection Established.  
TRACERT Service Client Selected.  
204.071.200.245  
Hop: Timeout has occurred.  
Hop: 192.168.000.001 at milliseconds: 00048.  
CSISERVER  
Hop: 204.210.234.001 at milliseconds: 00032.  
Hop: 024.095.082.006 at milliseconds: 00021.  
Hop: 024.095.081.065 at milliseconds: 00022.  
Hop: 024.095.081.143 at milliseconds: 00031.  
Hop: 024.095.080.186 at milliseconds: 00020.  
Hop: 024.095.080.181 at milliseconds: 00029.  
Hop: 012.125.143.061 at milliseconds: 00066.  
Hop: 012.123.005.074 at milliseconds: 00064.  
Hop: 012.122.001.153 at milliseconds: 00057.  
Hop: 012.122.002.150 at milliseconds: 00114.  
Hop: 012.122.001.162 at milliseconds: 00105.  
Hop: 012.123.013.057 at milliseconds: 00103.  
Hop: 012.127.201.018 at milliseconds: 00108.  
Hop: 206.132.150.237 at milliseconds: 00097.  
Hop: 206.132.254.041 at milliseconds: 00120.  
Hop: 208.178.103.062 at milliseconds: 00097.  
TRACERT was successful, milliseconds: 00102.  
TRACERT Service Client Completed.  
TCP201I Client -- Shutdown --  
TCP205I Connection Complete -- Already Closed
```

Note that some TCP/IP stacks do not permit traceroute requests to pass through them. VM is an example of one such stack, so if TCP/IP FOR VSE is running through TCP/IP for VM, the first hop always appears as a timeout.

DISCOVER

The syntax of the DISCOVER command is

```
DISCOVER {nnn.nnn.nnn.nnn | symbolic_name}
```

This is also the format for issuing the DISCOVER command from the TCP/IP FOR VSE console interface.

The variables are described in the following table:

Variable	Description
<i>nnn.nnn.nnn.nnn</i>	The IP address in dotted-decimal notation.
<i>symbolic_name</i>	The symbolic name of a TCP/IP host. The symbolic name is translated to an IP address by the local DNS client.

The following example shows two DISCOVER commands:

```
DISCOVER 100.50.50.90
DISCOVER TCPIP4VSE.COM
```

The DISCOVER processor makes five attempts to reach the remote location. It lists the success or failure of each attempt and the time required for each response.

CICS Command Mode

If you enter the DISCOVER, TRACERT, or PING commands interactively in CICS without any operands, you enter *command mode*. When you are in command mode, the command processor prompts you for subcommands. It does not matter which command you use to enter command mode. All of the subcommands are valid from all environments. These commands are summarized in the following table:

Subcommand	Description
HELP	Provides information about subcommands.
EXEC <i>name</i>	Reads the specified .L book member from the library and executes any subcommands that it finds in that member. You might use this facility to DISCOVER many different hosts with a single command.
DISCOVER	Executes the DISCOVER utility.

Subcommand	Description
PING	Executes the PING utility against the host defined in SET HOST.
TRACERT	Executes the traceroute utility against the host defined in SET HOST.
QUIT	Terminates DISCOVER mode.
SET HOST= <i>n.n.n.n</i> SET IP= <i>n.n.n.n</i> SET IPADDR= <i>n.n.n.n</i>	Specifies an IP address that is to be the target for future DISCOVERs.
QUERY OPTION	Displays the current address setting of the host.

The following example shows output from the CICS client:

```
DISCOVER TCPIP4VSE.COM
TCP200I Client -- Startup --
TCP207I Copyright (c) 1995-2006 Connectivity Systems Incorporated
TCP202I Attempting to Establish Connection
TCP204I Connection has been Established
Client manager connection Established.
Discover Service Client Selected.
206.152.227.171
The best MTU discovered: 01032.
Discover Service Client Completed.
TCP201I Client -- Shutdown --
TCP205I Connection Complete -- Already Closed
```

Running Batch Clients

TCP/IP FOR VSE also provides a batch client for ping, traceroute, and DISCOVER. Use the CLIENT command with the following syntax to run these utilities. The parameters are described below.

```
// EXEC CLIENT,PARM='APPL={PING|TRACERT|DISCOVER},ID=sysid,QUIET=YES'
```

Sample JCL

To issue a ping, traceroute, or DISCOVER command from batch, use the following job as an example:

```
* $$ JOB JNM={PING|DISCOVER},CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB PING
// EXEC CLIENT,PARM='APPL={PING|TRACERT|DISCOVER},ID=sysid,QUIET=YES'
command 1
command 2
. . .
command n
/*
/&
* $$ EOJ
```

You can run the batch ping or DISCOVER client once to issue multiple pings/DISCOVERs to multiple IP addresses. The ping, traceroute, and DISCOVER commands are described in the previous section.

The parameters are keyword parameters, so you can specify them in any order. They are described in the following table:

Parameter	Description
ID= <i>sysid</i>	Specifies the stack ID of the TCP/IP partition you want to connect to. Remember that you can run multiple copies of TCP/IP at one time (for example, production and test). The default is <i>00</i> .
APPL={PING TRACERT DISCOVER}	Specifies the client you want to run. This parameter is required.
QUIET=YES	Instructs the client to suppress informational messages from the VSE console. It is a keyword parameter (<i>YES</i> is the only valid value).

Return Codes

The return codes for each client type are described in the following table. To understand the return codes, you must note the difference between ping (or DISCOVER) attempts and ping (or DISCOVER) commands.

Return Code	PING	TRACERT	DISCOVER
0	The ping commands completed normally. Each PING command generates five ping attempts. If any of the five attempts succeeds, then the command is successful. If all PING commands in the input stream succeed, then the return code is zero.	The TRACERT command completed successfully.	The DISCOVER commands completed normally. The maximum size of a packet that can be generated without fragmentation has been provided.
4	At least one of the commands timed out during all five attempts. If you have multiple PING commands in the job stream, any one of the commands could cause the problem. Examine the job output to determine which specific command is failing.	The TRACERT command failed, meaning that TCP/IP FOR VSE could not locate the remote host.	At least one of the DISCOVER commands timed out. You probably do not have connectivity to the requested host.
8	One of the ping commands returned with an error. You receive this return code if you specify a symbolic name and TCP/IP FOR VSE is unable to locate the name by performing a DNS lookup or by searching the table created with the DEFINE NAME command.	One of the traceroute commands returned with an error. You receive this return code if you specify a symbolic name and TCP/IP FOR VSE is unable to locate the name by performing a DNS lookup or by searching the table created with the DEFINE NAME command.	One of the DISCOVER commands returned with an error. You receive this return code if you specify a symbolic name and TCP/IP FOR VSE is unable to locate the name by performing a DNS lookup or by searching the table created with the DEFINE NAME command.

Return Code	PING	TRACERT	DISCOVER
12	One of the parameters is incorrectly specified.	One of the parameters is incorrectly specified.	One of the parameters is incorrectly specified.
16	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that the specified ID= value corresponds to the stack ID for an active TCP/IP FOR VSE partition.	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that the specified ID= value corresponds to the stack ID for an active TCP/IP FOR VSE partition.	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that the specified ID= value corresponds to the stack ID for an active TCP/IP FOR VSE partition.

TCP/IP for VSE EMAIL

Overview

As with most computer applications, e-mail clients have their own terminology. To use e-mail, you need to understand the following common terms.

Term	Description
SMTP	Simple Mail Transport Protocol. SMTP is the communication protocol that sends e-mail from the client (in our case, VSE) to an SMTP server. The SMTP server is sometimes called a <i>mail gateway</i> . The SMTP client's responsibility is simply to send the mail to the SMTP server. After it is there, the SMTP server may store it for later delivery, forward it immediately to another SMTP server, or throw it away (for example, if it was spam). TCP/IP FOR VSE provides an SMTP client, but it does not provide an SMTP server.
POP, POP3	Post Office Protocol. POP provides a protocol that enables an end user to connect to a mail server and retrieve stored e-mails. With POP, PCs that receive e-mail can be turned off, disconnected, or otherwise inaccessible when the e-mail is sent.
MIME	Multipurpose Internet Mail Extensions. MIME extends SMTP capabilities to enable the transmission of binary objects such as pictures and programs. Standard SMTP protocol does not permit the transmission of binary objects.

You can use TCP/IP FOR VSE to send e-mail using the standard SMTP mechanism. Most e-mail systems can receive e-mails and attachments that use this protocol. In this chapter we discuss the following topics:

- Setting up the TCP/IP FOR VSE EMAIL client

- Running the EMAIL client, including the four methods you can use to start it
 - Sending e-mail
 - Using EMAIL client commands. This section includes a command summary list and a detailed explanation of each command.
-

Setting Up the EMAIL Client

Before you can use e-mail, you must install or have access to an SMTP server. Most organizations already have an e-mail mechanism in place. In this case, you can send mail from VSE as soon as you find out the IP address or the domain name address of your SMTP server. This address can be internal to your organization or it can be a public SMTP server whose address you obtain from your Internet Service Provider (ISP).

If you cannot access an SMTP server, there is an option with the client to use the SMTP server of the recipient instead of the SMTP server of the sender. It is recommended, however, that you use your own server and set the name as your default, thus eliminating the need for extra statements in your EMAIL client execution.

Running the EMAIL Client

You can run the EMAIL client using the following four methods:

- As a CICS transaction.
- Automatically, by using the AUTOEMAIL facility. AUTOEMAIL works like AUTOFTP and AUTOLPR in that e-mail is sent automatically when entries appear in a monitored POWER LST or PUN queue.
- In batch, by using the EMAIL batch client.
- In an application program, by using the REXX, Assembler, COBOL, or PL/1 Sockets interface.

We discuss the first three methods in this chapter. For information about using the Sockets interface, see the *TCP/IP FOR VSE Programmer's Guide*.

Using CICS EMAIL

To start the EMAIL client from your CICS terminal, log on to CICS and issue the following command:

```
EMAIL
```

There are no parameters. Note that EMAIL uses SYSID 00, and you cannot use a different SYSID. EMAIL responds by displaying the following messages on your screen:

```
EMAIL
TCP200I Client -- Startup --
TCP207I Copyright (c) 1995-2006 Connectivity Systems Incorporated
TCP202I Attempting to Establish Connection
TCP204I Connection has been Established
Client manager connection Established.
EMAIL Ready:
```

The EMAIL client is now in command mode and can accept EMAIL client commands. The commands are explained later in this chapter. To terminate the EMAIL client from your CICS terminal, use the QUIT command. You can also press the PF3 key to terminate the interactive program at any point where a command is ready to be accepted by the user. After you enter the QUIT command, the CICS transaction tells you to press the PF3 key to return to the IUI. As you can see, this is not the most convenient method of sending e-mail, which is why there are other forms of e-mail delivery.

Using AUTOEMAIL

You can use the automatic EMAIL client (AUTOEMAIL) to automatically send the output from a job to a specific e-mail destination. To determine how to handle your output, AUTOEMAIL processes information from the following sources:

- The DEFINE EVENT command, which your system administrator codes.
- The * \$\$ JOB, * \$\$ LST, and * \$\$ PUN statements, which you code. Note that values on the * \$\$ LST and * \$\$ PUN statements override values on the * \$\$ JOB statement.
- EMAIL scripts, which are coded and stored in an *L* book by your system administrator.

Each source is discussed in the remainder of this section.

If an e-mail fails, AUTOEMAIL can be set up to retry the operation as many as nine times. See DEFINE EVENT in the *TCP/IP FOR VSE Command Reference* for information on this setting.

DEFINE EVENT

Before you can use AUTOEMAIL, your system administrator must use the DEFINE EVENT command to define one or more POWER LST or PUN queues for TCP/IP FOR VSE to monitor. To determine which POWER queues are monitored, you can issue the QUERY EVENTS command from a VSE operator console. When a given LST queue is monitored, TCP/IP FOR VSE uses information from the DEFINE EVENT command statement and the * \$\$ LST or * \$\$ PUN statements to determine what actions to take. The options that may be specified with DEFINE EVENT are as follows:

```
DEFINE EVENT, ACTION=EMAIL, CLASS=power_class,
          FCBPREFIX=string,
          HOSTNAME=[USERINFO | ROOM | DEST | DEPT | BLDG],
          NULLFILE=[SKIP | PROCESS],
          ORDER=[YES | NO],
          PASSWORD=password,
          POWERSYSID=sysid,
          PRIORITY=[YES | NO],
          QUEUE=[LST | PUN],
          RETRY=count,
          RETRY_TIME=interval,
          SCRIPTTYPE=file_extension,
          SINGLE=[YES | NO],
          USERID=userid
```

JECL Statements

The format of the * \$\$ LST or * \$\$ PUN statement that you use depends primarily on the following two factors:

- Whether you plan to use an automatic EMAIL script.
- The HOSTNAME that the system administrator specifies on the DEFINE EVENT command for the LST or PUN class you are using.

The formats of the * \$\$ JOB and * \$\$ LST statements that you include in your POWER JECL are shown in the following box:

```
* $$ JOB ROOM=name,DEPT=name,BLDG=name
* $$ LST CLASS=class,DISP=disp,USER='name',DEST=(,name),
* $$ LST UCS=inserts,FCB=fcb,BLDG=name,ROOM=name,
* $$ LST DEPT=name
```

These formats show only the relevant options. The variables are described in the following table:

Variable	Description
<i>class</i>	The POWER LST queue job class. TCP/IP FOR VSE must monitor this class as directed by the DEFINE EVENT command.
<i>disp</i>	Disposition of the LST queue entry. TCP/IP FOR VSE honors the disposition after the queue entry is successfully processed depending on the value of the SET DISP EMAIL client command. For more information, see the SET DISP command description (page 6.35).
<i>fcb</i>	The name of a Forms Control Buffer (FCB). An FCB permits you to emulate more complex line spacing. See SET FCB (page 6-39) for more information.

Variable	Description
<i>inserts</i>	The name of an INSERTS phase. You can use an INSERTS phase to further customize the e-mail. The INSERTS phase is discussed later in this chapter; see SET INSERTS (page 6-43).
<i>name</i>	<p>A string representing the address of the SMTP server that is to process your e-mail request. This variable can be</p> <ul style="list-style-type: none"> • An IP address in dotted decimal notation, such as <i>192.168.0.7</i> • A script name • A symbolic IP address, which can be a name defined with the DEFINE NAME command or a name obtained with a domain name server. <p>To determine which value <i>name</i> represents, TCP/IP FOR VSE checks:</p> <ul style="list-style-type: none"> • Whether it is a valid IP address. To specify an IP address in dotted decimal notation, the system administrator can use the DEFINE EVENT command with the HOSTNAME=USERINFO parameter (for example). For <i>USERINFO</i>, the IP address is coded in the USER= field. • Whether it is a script name defined by your system administrator using the DEFINE NAME command. A <i>script</i> is a set of EMAIL commands and is contained in a library. For more information about defining scripts, see the DEFINE NAME command in the <i>TCP/IP FOR VSE Command Reference</i>. • Whether it is a symbolic name for an IP address or a mail server domain name. A symbolic name is defined with the DEFINE NAME command. If TCP/IP FOR VSE finds a symbolic name, it substitutes the associated IP address. If it does not find a symbolic name, it checks the Domain Name Server (if one is defined) to see if it can resolve the <i>name</i> value. If not, the automatic e-mail fails. <p>For more information about specifying <i>name</i>, see the examples at the end of this chapter in “E-mail Examples” (page 6-59).</p>

EMAIL Scripts

You can define a limited amount of information on JECL statements. In addition, you can use scripts to define commonly used e-mail parameters such as recipients, carbon copy recipients, and attachments. Use the following procedure to create an e-mail script:

1. Code a set of EMAIL commands. Note that you must always supply a host name and a recipient. You can put this information in your script, or you can put it directly on the * \$\$ LST statement. Values on the * \$\$ LST statement or the * \$\$ PUN statement override any values specified on the * \$\$ JOB statement. The values coded in EMAIL client commands, in turn, override any specifications on the * \$\$ JOB or * \$\$ LST statements.
2. Save the commands in an *L* book in a library that is accessible to the TCP/IP FOR VSE partition. If you want to use a book type other than “.L,” you need to indicate that in the DEFINE EVENT command using the SCRIPTTYPE= parameter.
3. Use the DEFINE NAME command to define the member as a script. Be sure to include the SCRIPT= parameter in the definition.
4. Identify the script by specifying its name as described in the *name* field in the above table.

Command Variables

The following command variables are available when creating AUTOEMAIL scripts. These variables are common to AUTOLPR and AUTOFTP. If user-defined POWER field names are included in the LST card (for example) that created them, then those variables will also be available. For details on defining user variables, see the IBM manual *VSE/POWER Administration and Operation*.

Variable	Description	Length
&CPUID	Numeric CPU identifier	6
&CPUSECS	Current time in 1/300 th -second units. This numeric value may be used as a random number for generating a unique file ID.	9
&CURDATE	Current date in the form <i>mmddycc</i> , where <i>mm</i> is the month, <i>dd</i> is the day, <i>yy</i> is the year, and <i>cc</i> is a century of 19 or 20. You do not need to issue the SETVAR command to access this variable.	8
&CURTIME	Current time in the form <i>hhmmss</i> , where <i>hh</i> is the hour (using a 24-hour clock), <i>mm</i> is the minute, and <i>ss</i> is the second. You do not need to issue the SETVAR command to access this variable.	6

Variable	Description	Length
&PWRBLDG	Building name assigned to the queue entry from the BLDG= parameter on the * \$\$ JOB statement	8
&PWRCDAT	Synonym for &PWRXDAT	
&PWRCLAS	LST or PUN class for the listing that is being transferred, from the * \$\$ LST or the * \$\$ PUN statement	1
&PWRCOPY	Number of requested copies as set by the COPY= parameter on the * \$\$ LST statement	2
&PWRCTIM	Synonym for &PWRXTIM	
&PWRDEPT	Department name assigned to the queue entry from the DEPT= parameter on the * \$\$ LST statement.	8
&PWRDEST	Name of the destination on the * \$\$ LST statement. This can also be the name of the “.L” book from which the AUTOEMAIL client pulls EMAIL client commands.	8
&PWRDISP	Disposition for the LST or PUN queue entry from the DISP= parameter on the * \$\$ LST or * \$\$ PUN statement	1
&PWRDIST	Distribution code associated with the POWER queue entry from the DIST= parameter of the * \$\$ LST statement.	8
&PWRFLSH	FLASH field identifier. If not specified, it is blank. It is used for laser printers.	4
&PWRFLSN	Number of FLASH copies for laser printers. The default for this numeric field is 000.	3
&PWRFORM	Form identifier from the FNO= parameter on the * \$\$ LST statement.	4
&PWRFRMD	Special FORMDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRLCNT	Number of lines in the file	8
&PWRNAME	Name of the job that sent the output to the POWER LST or PUN class that TCP/IP FOR VSE is monitoring	8
&PWRNUMB	Number that POWER assigned to the job	5
&PWROFCB	Name of the Forms Control Buffer (FCB) specified on the FCB= parameter on the * \$\$ LST statement	8

Variable	Description	Length
&PWROUCB	Name of the character set specified on the UCS= parameter of the * \$\$ LST statement	8
&PWRPAGD	Special PAGEDEF field used for CICS spooling of POWER data (set up during POWER initialization). Its length depends on the SPOOLOPEN request.	Variable
&PWRPCNT	Number of pages in the file in numeric format. This is for LST data only.	8
&PWRPRGN	Name of the programmer from the PROGR= parameter on the * \$\$ JOB statement.	20
&PWRPRTY	Priority of the queue entry from the PRI= parameter on the * \$\$ JOB or * \$\$ LST statement	1
&PWRQDAT	Synonym for &PWRXDAT	
&PWRQTIM	Synonym for &PWRXTIM	
&PWRQUE	Either <i>PUN</i> or <i>LST</i> to specify the queue the listing came from	3
&PWRROOM	Room number assigned to the queue entry from the ROOM= parameter on the * \$\$ JOB statement	8
&PWRSID	System ID that is to process the queue entry, from the SYSID parameter on the * \$\$ LST statement	1
&PWRSUFF	Suffix that POWER assigned to the listing	3
&PWRTNOD	Target node name for the queue entry in a PNET environment, as specified on the LDEST parameter of the * \$\$ JOB statement for the LST queue and the PDEST parameter of the * \$\$ JOB statement for the PUN queue	8
&PWRUINF	User information field specified on the USER= parameter from the * \$\$ JOB statement or the * \$\$ LST statement. This field is arbitrary and may be changed with the PALTER command.	16
&PWRUSRI	VSE user ID associated with the job	8
&PWRXDAT	Date of the POWER queue entry in either <i>ddmmyyyy</i> or <i>mmddyyyy</i> format, depending on the IPL option of your VSE system. This value reflects when the entry was initially created or opened.	8

Variable	Description	Length
&PWRXTIM	Time of the POWER queue entry in <i>hhmmss</i> , where <i>hh</i> is the hour using a 24-hour clock. This value reflects when the entry was initially created or opened.	6
&QNUM	Unique POWER queue member identifier. (This is not the POWER job number.)	5

AUTOEMAIL Output

The following table shows how TCP/IP FOR VSE processes the queue entry after the EMAIL operation. The EMAIL client command *SET DISP* may change the actions in the table. For more information about disposition processing with AUTOEMAIL, see the SET DISP command (page [6.35](#)).

\$\$ LST Disposition	Successful AUTOEMAIL	Unsuccessful AUTOEMAIL
DISP=D	The job is deleted from the queue.	After the first unsuccessful attempt, the job is requeued with disposition <i>Y</i> . If all subsequent attempts fail, the job remains in the LST or PUN queue with disposition <i>Y</i> . If you use the PALTER command to change the disposition to <i>K</i> or <i>D</i> , AUTOEMAIL tries again.
DISP=K	The job is kept in the POWER LST or PUN queue with disposition <i>L</i> .	

AUTOEMAIL Example

In this example, the system administrator uses the DEFINE EVENT command to monitor POWER LST queue output in class M. The DEFINE EVENT command also specifies the parameter *HOSTNAME=USERINFO*. When the sample job runs, it uses AUTOEMAIL to run the EMAIL script *mailjohn*, which was defined previously with the DEFINE NAME command. The sample job is as follows:

```
* $$ JOB JNM=LIBRLIST,CLASS=A,DISP=D
* $$ LST CLASS=M,DEST(*,LOCAL),USER='MAILHIM'
// JOB LIBRLST
// EXEC LIBR
LISTDIR S=PRD2.TCPIP
/*
/&
* $$ EOJ
```

The script *mailhim* might look like this:

```
SET HOST=smtp-server.ourcompany.com
SET FROM=vse@mymainframe.com
SET TO=person@someplace.com
SET SUBJECT=Automatic E-mail Listing
```

To use the automatic EMAIL client effectively, you may want to use an individual script for each recipient or group of recipients. The script *mailhim*, for example, can be used for any output that is sent to *person@someplace.com*.

Using Batch EMAIL

There may be times when you want to run e-mail from a batch environment. TCP/IP FOR VSE's EMAIL client has a batch mode that allows you to do this.

VSE Name Definition

Before you can use the batch EMAIL facility, the name of the local VSE system must be defined in the TCP/IP initialization deck. Use the DEFINE NAME command to add this definition, as follows:

```
DEFINE NAME,NAME=vse,IPADDR=ip.addr
```

The variables have the following meanings:

- *vse* is any name
- *ip.addr* is the IP address used in the SET IPADDR command.

See the *TCP/IP FOR VSE Command Reference* for more information about the DEFINE NAME command.

Batch Example

The JCL statements in the following box show how to use the batch EMAIL facility:

```
* $$ JOB JNM=EMAIL,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB EMAIL
// EXEC EMAIL,PARM='ID=nn, LASTCOL=80, QUIET=YES'
EMAIL command 1
EMAIL command 2
EMAIL command n
/*
/&
* $$ EOJ
```

Parameters The parameters for EXEC EMAIL are keyword parameters, so you can specify them in any order. They are described in the following table:

Parameter	Description
AUTOEXEC=OFF	Specifies that commands stored in EMAIL.AUTOEXEC are not executed automatically in each run. By default, the EMAIL client executes the commands in this file before processing the commands in your batch job. (Note: <i>OFF</i> is the only valid value.)
DEBUG=ON	Specifies that additional information is sent to SYSLST and SYSLOG for problem debugging. By default, debugging messages are disabled. (Note: <i>ON</i> is the only valid value.)
ECHO=[ON OFF]	Specifies that responses from the internal client are sent back to SYSLST. The default is <i>ON</i> .
ID= <i>nn</i>	Specifies the system ID of the TCP/IP partition you want to connect to. Remember that you can have more than one copy of TCP/IP running at once, for example, production and test. The default is <i>00</i> .
LASTCOL= <i>nn</i>	Specifies the last column of command and message text to be processed. The maximum is 80; the default is 72. Note: Setting the TEXT command's TRUNC parameter to <i>YES</i> limits message text lines to 72 columns. See the TEXT command (page 6.28) for more information.
QUIET=YES	Suppresses informational messages when the batch EMAIL client executes. By default, these messages are allowed. (Note: <i>YES</i> is the only valid value.)
UPPERCASE	Forces all messages to be in uppercase. If it is omitted, all messages are in mixed case. There are no values to set.
VAR=[ON OFF]	Specifies whether variable names contained in the commands are replaced with the values in the VSE JCL that enabled them. The default is <i>ON</i> .

Commands You can issue any number of EMAIL commands using the batch EMAIL client. Before using the SEND client command, however, you must use SET commands to specify the SMTP server name (SET HOST), the FROM value, and the TO value. These values can be set as defaults in the EMAIL command that is part of the TCP/IP initialization process. Note that if you do not issue a SEND command, nothing is actually sent. See “[Using EMAIL Client Commands](#)” (page [6-16](#)) for a summary of each command.

Return Codes The EMAIL client may return the following codes:

Return Code	Description
0	Your EMAIL commands completed successfully. Note that if you do not issue the SEND command at some point, you can receive a return code of zero even though no e-mail is sent.
4	A syntax error occurred in one of the EMAIL commands in the job stream. To find the problem, study the output. Common problems include an inability to change to the specified directory and file not found.
8	An error occurred in one of the EMAIL commands in the job stream. The error could be a syntax error or it could be another type of error. To find the problem, study the output. One or more EMAIL commands might have completed successfully, so you need to determine the current status before you rerun the job.
12	One of the parameters was incorrectly specified.
16	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that your ID= parameter value corresponds to the SYSID for an active TCP/IP FOR VSE partition.

Sending E-mail

To send e-mail from VSE, use these steps:

1. Start the EMAIL client using one of the methods described in the previous section.
 2. Issue EMAIL client commands to specify characteristics.
 3. Issue EMAIL client commands to specify the names of attachments (if any) you want to send.
 4. Issue the EMAIL SEND command to actually send the e-mail. (If you skip this step, nothing is sent.)
 5. The EMAIL client performs all of the desired actions and looks for more commands or a QUIT.
 6. The EMAIL client terminates when “QUIT” or some other terminator is received, such as PF3 from CICS, ‘/*’ from batch, or a SOCKET CLOSE from the API.
-

Using EMAIL Client Commands

The TCP/IP FOR VSE EMAIL facility provides a set of commands to direct EMAIL operation. In general, these commands are valid from all EMAIL applications, including CICS EMAIL transaction, AUTOEMAIL, and batch. The required commands are

- SET HOST, SET FROM, and SET TO. These commands must be in place before you issue the SEND command:
- SEND and QUIT. The SEND command sends the data; the QUIT command terminates the EMAIL client.

The EMAIL client commands are summarized in the following table. The SET commands are listed separately. A detailed explanation of all commands follows these tables.

Command	Description
ATTACH	Specifies a file to send as an attachment. This command can be repeated to send multiple files.
CD	Changes the current directory of your EMAIL client.
DIR LS	Lists the specified directory in the TCP/IP FOR VSE file system.
EXEC	Processes a series of EMAIL commands contained in a script file.
HELP	Displays the syntax of the requested EMAIL command.
PWD	Displays the current working directory.
QUERY OPTIONS	Displays the values specified with the SET command.
QUIT	Stops the EMAIL client.
RPORT	Specifies a port number to use other than the default. For example, Microsoft Exchange and others typically use port 25.
SEND	Sends an e-mail. You can attach a single file from the TCP/IP FOR VSE file system if you want to.
TEXT	Specifies arbitrary text to be placed in the body of the e-mail.

The SET commands are listed in the following table. Command synonyms are listed together.

Command	Description
SET ATTACHFAIL	Specifies whether to send an e-mail that was to include an attachment file if the attach process fails.
SET AUTH	Specifies whether the USERID/PASSWORD values are passed to the SMTP server prior to processing.
SET BIN SET BINARY	Manually specifies that the attachment file is to be processed as a binary file.
SET BCC SET BLIND SET BLINDcopy	Specifies an additional e-mail address that is invisible to all other recipients. See the detailed explanation for SET TO.
SET BCCLIST SET BLINDCOPYLIST SET BLINDLIST	Specifies a member that contains a list of “blind copy” addresses.
SET BLKSIZE	Specifies the block size of a file to be attached. See also SET LRECL and SET RECFM.
SET CC	Specifies that a file is to be e-mailed with standard carriage control.
SET CHECKNAME	Specifies that the EMAIL program does not validate the form of e-mail addresses used.
SET COPY	Specifies additional e-mail recipients. See the detailed explanation for SET TO.
SET COPYLIST	Specifies a member containing a list of addresses for the “CC” field.
SET CRLF	Specifies whether to insert carriage return/line feed (CR/LF) combinations into the attached file.
SET DEBUG	Enables extra debugging messages for troubleshooting a problem.
SET DISP	Specifies the disposition of a POWER LST or PUN queue entry after the SEND command executes.
SET DUPES	Permits the same attachment to be processed multiple times for the same session with the same name.
SET ECHO	Displays the responses from the server.
SET EOJEXEC SET EXECEJOJ	Specifies a command script to be executed after the last user command is processed and just before the EOJ.
SET EXTYPES	Specifies that the EXTYPES.L table is to be used to determine the MIME type for processing the attachment.

Command	Description
SET FCB	Specifies the name of a phase to be used as the Forms Control Buffer (FCB) for the attached file.
SET FCBPREFIX	If the POWER LST parameter has "FCB=\$\$\$\$.xxx," this parameter specifies the "\$\$\$\$" part of the name.
SET FILETYPE	Specifies the file type of a script containing e-mail commands.
SET FROM	Specifies the e-mail address of the sender. Some SMTP servers try to authenticate this name, and others allow you to use any name.
SET HOST SET IP SET IPADDR	Specifies the IP address of the host on which the SMTP server is running. You can specify this value in dotted decimal notation or as a domain name.
SET INSERTS	Specifies an inserts phase whose contents are to be transmitted with the attached file.
SET JOIN	Combines multiple text attachments into a single attachment.
SET JSEP SET JSEPCOUNT	Specifies the number of separator pages to request from POWER if the SET SEPARATOR command is ON. (The TCP/IP FOR VSE SEPARATOR_PAGES command sets the default count.)
SET LANGUAGE	Specifies a special language file for countries that use SBCS character sets that are not in the generally used alphabet.
SET LOCK	Prevents a POWER entry from being changed by another job while the current job is being sent.
SET LPASS SET LPASSWORD	Specifies a password. Used only by the security manager.
SET LRECL	Specifies the logical record length of a file to be attached. See the detailed explanation for SET BLKSIZE (page 6.31).
SET LUSER SET LUSERID	Used only by the security manager.
SET MAILSERVER	Indicates that the client is to find the IP address of the recipient's e-mail server and use that for sending output. The value used is the recipient's domain name.
SET NEWNAME	Assigns a name to the attachment file.
SET NOEJECT	Suppresses the initial form feed character, if present, at the beginning of an e-mail attachment.
SET NOTIFY	Indicates where to send a failure notice if e-mail delivery fails.

Command	Description
SET PASS SET PASSWORD	The password string used when SET AUTH is used for e-mail authentication.
SET PDF	Causes automatic translation of text to PDF for e-mail attachments.
SET PRIORITY	Specifies the urgency of the e-mail for the recipient.
SET RECFM	Specifies the record format of a file to be attached. See the detailed explanation for SET BLKSIZE (page 6.31).
SET REPLYTO	Upon a reply, specifies an e-mail address for the "TO:" field.
SET REPLYTOLIST	Upon a reply, specifies a member containing a list of e-mail addresses for the "TO:" field.
SET SEP SET SEPARATOR	When attaching from POWER, specifies that any JSEP pages that the POWER driver reads are to be returned rather than ignored. If there are no JSEP pages, then enabling this command has no effect.
SET SMTP	Synonym for SET MAILSERVER
SET SNAPDUMP	Specifies that a PDUMP of all datagram traffic for this session be placed in SYSLST in the TCP/IP partition.
SET SOSI	Sets the DBCS translation indicators.
SET SUBJECT	Specifies a subject line.
SET TLS	Specifies that the e-mail client should attempt to log in using an SSL/TLS handshake.
SET TO	Specifies the primary e-mail recipient.
SET TOLIST	Specifies a member that contains a list of addresses for the "TO" field.
SET TRANSLATE	Specifies a translate table for EBCDIC-to-ASCII translation of both the message and any attached file.
SET TRATT SET TRATTACHMENT	Specifies a translate table for EBCDIC-to-ASCII translation of the attached files only. It overrides the table name set by SET TRANSLATE.
SET TRBODY	Specifies a translate table for EBCDIC-to-ASCII translation of the message text only. It overrides the table name set by SET TRANSLATE.
SET TRUNC SET TRUNCATION	Prevents sending trailing blanks in text attachments.
SET USER SET USERID	Specifies a user ID when SET AUTH is used to enable e-mail authentication.

Command	Description
SET VALIDATE	Removes non-displayable characters from text attachments.
SET VCARD	Enables attaching a VCARD, a Windows® identification system, to the message.

ATTACH

The syntax is

```
ATTACH file-name
ATTACH file-name AS attachment-name
```

This command allows you to send multiple file attachments in an e-mail. You must issue the command for each file you want to send. The ATTACH command extends the functionality of SEND, which by itself allows only one file to be sent. You can issue ATTACH multiple times to attach more than one file, or even the same file under a different name. As with SEND, you can assign a new name to an attached file. There is no default.

Each ATTACH adds a file to the list of files to send, but the server is not contacted until a SEND is issued (with or without parameters). Therefore, SEND must follow any ATTACH commands you use.

The variables are described in the following table:

Variable	Description
<i>file-name</i>	This is either a valid file in the TCP/IP FOR VSE file system or an autonomous file (see text). For files defined to TCP/IP, the name of the current working directory always precedes <i>file-name</i> , thus forming a fully qualified file name. For example, if the current working directory is <i>PRD1.MACLIB</i> and you issue the command ATTACH WTO.A, the EMAIL client sends the file <i>PRD1.MACLIB.WTO.A</i> as an attachment. You can change directories (CD) prior to issuing an ATTACH, or you can specify the file's full path name.
<i>attachment-name</i>	The name the recipient sees for the attached file. Note: The file type (.TXT or .PDF, for example) is taken from <i>attachment-name</i> , if it is provided, and not from <i>file-name</i> .

Autonomous files are not defined to TCP/IP FOR VSE. To specify an autonomous file as an attachment in batch EMAIL, use the following syntax:

```
ATTACH %dlbl/tlbl-name,type[,recfm,lrecl[,blksize]] [AS attachment-name]
```

For example:

```
ATTACH %LSBESDS,ESDS,V,150 AS LSBTST.TXT
```

A percent sign (%) must precede the DLBL/TLBL name and indicates that this is an autonomous file. The RECFM, LRECL, and BLKSIZE values are optional, but the file type is required and must be one of the following: *SAM*, *ESDS*, *KSDS*, or *TAPE*. (Use *SAM* as the file type for BIM-EPIC-controlled files.) The valid values for RECFM are *F*, *FB*, *V*, *VB*, *SB*, and *SU*. See “[Autonomous Files](#)” (page 2-47) for more information on these formats. If SET LRECL (SET BLKSIZE) is not used, the default for LRECL (BLKSIZE) is 133. The value you set with the ATTACH command is the override. In general, LRECL and BKLSIZE values are ignored for VSAM and disk/tape manager files.

The DLBL/TLBL must be defined in either the batch EMAIL JCL or the system standard label area. For the ATTACH example above, the DLBL was defined in the JCL as follows:

```
// DLBL LSBESDS, 'LSB.ESDS.TEST66.FILE',,VSAM,CAT=VSESPUC
```

The following example shows how three attachments could be sent. The public files are defined to TCP/IP FOR VSE. The autonomous file is defined in the system standard labels.

```
ATTACH file1 /* TCP-IP public name
ATTACH %dlb1,ESDS,V,150 AS NEWNAME.TXT /* autonomous file
ATTACH file3 AS file3.pdf /* TCP-IP public name
SEND
```

See also “[Example 2](#)” (page 6-60).

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

CD

The syntax is

```
CD directory-name
CD ..
```

The CD command changes the current directory of your EMAIL client. This enables your EMAIL client to process commands using the appropriate directory within the TCP/IP FOR VSE file system. To display the EMAIL client’s current working directory, use the Print Working Directory (PWD) EMAIL client command.

The parameters are described in the following table:

Parameter	Description
<i>directory-name</i>	The name of the directory that you want to be the active directory. It must be a valid directory in the TCP/IP FOR VSE hierarchical file system. You can issue the DIR command to display the directories and files in the current working directory.
..	Moves the current working directory up one level in the hierarchy.

The following table shows which modes and defaults are valid for this command:

EMAIL Client	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	<i>POWER.lst/pun.cl</i> , where <i>lst/pun</i> is set based on whether the event initiating the automatic e-mail was a LST-queue or a PUNCH-queue event, and <i>cl</i> is the class specified on the DEFINE EVENT command.
Batch	Accepted	No default

DIR, LS

The syntax is

```
DIR directory-name
LS directory-name
```

This command lists the specified directory in the TCP/IP FOR VSE file system. The DIR and LS commands are equivalent.

The variable is described in the following table:

Variable	Description
<i>directory-name</i>	The name of the directory you want to list. The name you specify may contain regular characters and wild card characters. Each file in the current working directory is matched against the name you specify, and all names that match are displayed. The following wild-card characters are accepted: <ul style="list-style-type: none"> • An asterisk (*), which matches any number of characters. • A question mark (?), which matches any one character.

To show the effects of wild-card characters, assume that you have a directory with the following files:

```
f1.txt
file2.bjb
fn3.zip
filenum4.wav
x.lst
```

The following table shows the files listed by DIR when specific wild card patterns are used:

Command	Files Included in Output Listing
DIR f*	f1.txt, file2.bjb, f3.zip, filenum4.wav
DIR f*.Z*	fn3.zip
DIR F?.TXT	f1.txt
DIR F?*	f1.txt, file2.bjb, f3.zip, filenum4.wav
DIR *	All files
DIR ?	No files are returned because '?' matches only one character

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

E, EXEC

The syntax is

```
EXEC script_member
E script_member
```

This command executes a series of EMAIL commands that are contained in a script file. When EMAIL finds an EXEC command, it stops processing the commands in the input stream. It immediately processes the commands listed in *script_member*, and then it resumes processing the suspended stream. This means that the statements in *script_member* are logically included in the input stream as opposed to being a subroutine call. If *script_member* contains a QUIT command, the session actually terminates. There is no default for *script_member*.

The variable is described in the following table:

Variable	Description
<i>script_member</i>	This is a “.L” book that is contained in the search chain for the TCP/IP FOR VSE partition. The file type must be omitted from the command.

HELP

The syntax is

```
HELP [command-name]
```

This command displays the syntax of a specified EMAIL command or information on all commands if a command is not specified.

The variable is described in the following table:

Variable	Description
<i>command-name</i>	The name of the command for which you want information. By default, a listing of all commands is displayed.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	Displays a listing of all commands.
AUTOEMAIL	Not accepted	Not applicable
Batch	Not accepted	Not applicable

PWD

The syntax is

```
PWD
```

The PWD command displays the current working directory. There are no parameters.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

The anticipated response is shown in the following box:

```
pwd
"PRD2.JOHN"
EMAIL Ready:
```

QUERY OPTIONS

The syntax is

```
QUERY OPTIONS
```

This command generates a list of the values specified using the SET command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

QUIT

The syntax is

```
QUIT
```

This command stops the EMAIL client. It must appear after the SEND command.

See “[Example 1](#)” (page [6-59](#)) for an example of how to use QUIT in a script.

RPORT

The syntax is

```
RPORT port_num
```

This command specifies a port number to use other than the default. For example, Microsoft® Exchange and other applications typically use port 25. There is no default for *port_num*, which must be a valid port number.

SEND

The syntax is

```
SEND
SEND file-name
SEND file-name AS attachment-name
```

This command connects to the SMTP server and begins delivering any attachments as well as sending the e-mail (the SMTP headers and the in-body message). You can attach one file to the e-mail using the SEND command. (To send multiple attachment files, use one or more ATTACH commands.)

Before you issue the SEND command, you must issue the SET FROM, SET TO, and SET HOST commands, along with any ATTACH commands. The output of the SEND command is the entire dialogue between the EMAIL client and your SMTP server. It can be quite verbose.

The variables are described in the following table:

Variable	Description
<i>file-name</i>	This is either a valid file in the TCP/IP FOR VSE hierarchical file system or an autonomous file (see text). The name of the current working directory always precedes <i>file-name</i> , thus forming a fully qualified file name. For example, if the current working directory is <i>PRD1.MACLIB</i> and you issue the command SEND WTO.A, EMAIL sends the file <i>PRD1.MACLIB.WTO.A</i> as an attachment.
<i>attachment-name</i>	The name the recipient sees for the attached file.

Autonomous files are not defined to TCP/IP FOR VSE. To attach an autonomous file in batch EMAIL, use the following syntax:

```
SEND %dlbl/tlbl-name,type[,recfm,lrecl[,blksize]] [AS attachment-name]
```

For example:

```
SEND %LSBESDS,ESDS,V,150 AS LSBTST.TXT
```

A percent sign (%) must precede the DLBL/TLBL name and indicates that this is an autonomous file. The RECFM, LRECL, and BLKSIZE values are optional, but the file type is required and must be one of the following: *SAM*, *ESDS*, *KSDS*, or *TAPE*. (Use *SAM* for BIM-EPIC-controlled files.) The valid values for RECFM are *F*, *FB*, *V*, *VB*, *SB*, and *SU*. See “[Autonomous Files](#)” (page 2-47) for more information on these formats. If SET LRECL (SET BLKSIZE) is not used, the default for LRECL (BLKSIZE) is 133. The value you set with the SEND command is the override. In general, LRECL and BKLSIZE values are ignored for VSAM and disk/tape manager files.

The DLBL/TLBL must be defined in either the batch EMAIL JCL or the system standard label area. For the SEND example above, the DLBL was defined in the JCL as follows:

```
// DLBL LSBESDS, 'LSB.ESDS.TEST66.FILE' , ,VSAM,CAT=VSESPUC
```

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	Sends the requested POWER LST or PUN queue member. You can include additional SEND commands in your script. AUTOEMAIL does not require a SEND command. If you include one, you must precede it with its own list of recipients.
Batch	Accepted	No default

TEXT

The syntax is

```
TEXT EOD=[/+]eod TRUNC=[YES|NO]
e-mail_text
eod
```

This command specifies the body of the e-mail message. Entering the TEXT command places the EMAIL client in *text mode*. In text mode, you can enter as many lines of text as you choose. When you finish entering text, enter the appropriate two-byte terminator (defined by “EOD=eod”) at the beginning of the next line to terminate TEXT mode.

If you want to ignore columns 73–80, specify TRUNC=YES. The EOD and TRUNC parameters are separated with a space.

These parameters are described in the following table:

Parameter	Description
EOD	<p>The end-of-data characters. These two characters must begin the line that follows the text message. The default is ‘/+’.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. The EOD in the TEXT command and the EOD in the LIBR CATALOG command must be different if you are cataloging the input into a VSE library. Both commands use the same default (‘/+’), so the EOD in one of these commands must be set to a non-default value (for example, ‘/%’). 2. If you call the EMAIL client in a REXX program using the ADDRESS function and you use REXXIPT, do not use ‘/+’, ‘/*’, or ‘/&’ as the EOD in the TEXT command. Otherwise, REXXIPT will terminate SYSIPT processing prematurely.
TRUNC	<p>Truncates text lines to 72 characters. If set to YES, TRUNC overrides the LASTCOL parameter on EXEC EMAIL when LASTCOL is set to a value greater than 72. The default is NO.</p>

The variable is described in the following table:

Variable	Description
e-mail_text	<p>The text of the e-mail. The text length in 1.5F and prior versions is 100 bytes. A maximum of 80 bytes can be sent as part of SYSIPT-delivered data.</p>

If the first line of the text contains only the 6-byte string “<HTML>”, the text is sent with an HTML MIME type instead of a plain-text MIME type. This allows you to send formatted data. This also allows joining multiple lines of text that do not contain a “<P>” or a “
” HTML command.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	The <i>powerlistname</i> from the VSE listing
Batch	Not accepted	No default

SET ATTACHFAIL

The syntax is

```
SET ATTACHFAIL=[FLUSH|CONTINUE]
```

This command specifies whether to continue send the e-mail if the file-attachment process fails. The attachment process fails, for example, if a “File Not Found” error occurs. The default is FLUSH.

The parameters are described in the following table:

Parameter	Description
<u>FLUSH</u>	An attempt is made to tell the SMTP server to discard the e-mail, although in most test cases the SMTP server still tries to deliver the incomplete e-mail, usually with a null attachment (a zero-length file). The request generates a non-zero return code, and the rest of the job terminates prematurely.
CONTINUE	The e-mail is sent with a null attachment (a zero-length file).

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET AUTH

The syntax is

```
SET AUTH=[ON|OFF]
```

The USER ID and PASSWORD is sent to the SMTP server prior to processing the e-mail. The default is OFF.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET BINARY

The syntax is

```
SET BINARY=[ON|OFF]
```

The attachment file is to processed as a binary file. Remember that the preferred way of specifying how a file should be processed is to use a standard file type as defined in the EXTTYPES.L file. The default is OFF.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET BLINDLIST

The syntax is

```
SET BLINDLIST=membername
```

This command specifies a member that contains a list of “blind copy” e-mail addresses. The e-mail is copied invisibly to each address in the list. There is no default.

The variable is described in the following table:

Variable	Description
<i>membername</i>	The member name that contains a list of addresses. This is an “.L” type library member, but the file type must be omitted from the command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

**SET BLKSIZE,
SET LRECL,
SET RECFM**

The syntax is

```
SET BLKSIZE=blksize
SET LRECL=lrecl
SET RECFM=recfm
```

These commands specify characteristics of the attached file.

The variables are described in the following table:

Variable	Description
<i>blksize</i>	The block size used by the file. The default is 133.
<i>lrecl</i>	The logical record length of the records in the file. The default is 133.
<i>recfm</i>	The record format of the file. The default is FB.

The values you assign to these variables depend on the file type, as follows.

For **sequential disk and VSAM-managed SAM files**:

- Fixed-length records are padded when necessary. When padding occurs, text files are padded with blanks.
- You can process VSAM-managed SAM files as VSAM ESDS files, and it is more efficient to do so because the SAM emulation routines call VSAM anyway.
- The following table shows appropriate values:

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A
FB	record size	record size times blocking factor
V	maximum record size	N/A
VB	maximum record size	maximum block size
SU	maximum record size	N/A
SB	maximum record size	maximum block size

For **POWER files**:

- Fixed-length records are padded if necessary when writing to the POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with zeros.
- The minimum *lrecl* for POWER RDR queue files is 80 and the maximum is 128.

- The minimum *lrecl* for POWER LST queue files is 1 and the maximum is 32766.
- The *lrecl* for POWER PUN queue files must be 80.
- The following table shows appropriate values for POWER queues.

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A
V	maximum record size	N/A

For **VSAM files**, the following table shows appropriate values:

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	record size	N/A
V	maximum record size	N/A

For **ICCF files**, the files always contain 80-byte records, regardless of specification. The following table shows appropriate values:

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	N/A	N/A

For **Library files**, library members always contain fixed 80-byte records or a string file consisting of a single string of bytes. The following table shows appropriate values:

<i>recfm</i>	<i>lrecl</i>	<i>blksize</i>
F	80	N/A
SV	N/A	N/A
S	N/A	N/A

These SET commands are accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET CC

The syntax is

```
SET CC=[ON|YES|OFF|NO]
```

This command specifies whether you want the attached file to be sent with standard carriage control. The default depends on the client type (see below).

The parameters are described in the following table:

Parameter	Description										
ON YES	<p>Turns standard carriage control (CC) on. When CC is on, the EMAIL client examines the first character of each output line. Depending on the first character, the EMAIL client adds data to the output stream. The initial characters and their resulting actions are as follows:</p> <table border="1"> <thead> <tr> <th>Char</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Inserts a form feed, carriage return combination, unless the NOEJECT parameter is turned on or this is the first form feed in the data stream.</td> </tr> <tr> <td>(blank)</td> <td>Inserts a carriage return, line feed to skip one line.</td> </tr> <tr> <td>0</td> <td>Inserts a carriage return, line feed, carriage return, line feed combination into the data stream. This has the effect of skipping two lines.</td> </tr> <tr> <td>-</td> <td>Inserts a carriage return, line feed, carriage return, line feed, carriage return, line feed into the data stream. This has the effect of skipping three lines.</td> </tr> </tbody> </table>	Char	Action	1	Inserts a form feed, carriage return combination, unless the NOEJECT parameter is turned on or this is the first form feed in the data stream.	(blank)	Inserts a carriage return, line feed to skip one line.	0	Inserts a carriage return, line feed, carriage return, line feed combination into the data stream. This has the effect of skipping two lines.	-	Inserts a carriage return, line feed, carriage return, line feed, carriage return, line feed into the data stream. This has the effect of skipping three lines.
Char	Action										
1	Inserts a form feed, carriage return combination, unless the NOEJECT parameter is turned on or this is the first form feed in the data stream.										
(blank)	Inserts a carriage return, line feed to skip one line.										
0	Inserts a carriage return, line feed, carriage return, line feed combination into the data stream. This has the effect of skipping two lines.										
-	Inserts a carriage return, line feed, carriage return, line feed, carriage return, line feed into the data stream. This has the effect of skipping three lines.										
OFF NO	Turns standard CC off. POWER LST queue entries have ANSI carriage control in them, so if you turn CC off for POWER LST queue entries, your attachment contains carriage control characters in column 1.										

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	<i>NO</i>
AUTOEMAIL	Accepted	<i>YES</i> , if the event that triggered AUTOEMAIL was a LST queue event. <i>NO</i> , if the event that triggered AUTOEMAIL was a PUN queue event
Batch	Accepted	<i>NO</i>

SET CHECKNAME

The syntax is

```
SET CHECKNAME=[ON|OFF]
```

The EMAIL program validates the form of e-mail addresses used. The default is ON.

See also “[SET CRLF](#)” (page [6-35](#)).

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET COPYLIST

The syntax is

```
SET COPYLIST=membername
```

This command specifies a member that contains a list of “copy” e-mail addresses. The e-mail is copied to each address in the list. There is no default.

The variable is described in the following table:

Variable	Description
<i>membername</i>	Name of the member that contains a list of addresses. This is an “.L” type library member, but the file type must be omitted from the command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET CRLF

The syntax is

```
SET CRLF=[YES|NO|UNIX]
```

This command specifies whether to insert carriage return/line feed (CR/LF) combinations into the data stream for the attached file. VSE files typically are record oriented and thus do not contain CR/LF characters. ASCII systems are byte oriented and require CR/LF characters. This command is relevant only when the SET CC command specifies that carriage control is OFF. The default is YES.

The parameters are described in the following table:

Parameter	Description
<u>YES</u>	Inserts CR/LF characters into the data stream (the default).
NO	Does not insert CR/LF characters into the data stream.
UNIX	Inserts a line feed (LF) character into the data stream.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET DEBUG

The syntax is

```
SET DEBUG=[ON|OFF]
```

This command enables the output of additional system messages for debugging. The default is OFF.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET DISP

The syntax is

```
SET DISP=[KEEP|RESPECT|DELETE|HOLD]
```

This command specifies the disposition of a POWER LST or PUN queue entry after the SEND command executes. If you are sending anything other than POWER queue entries, the command is ignored. The default depends on the client type.

The following tables show the resulting disposition for each command setting and original disposition.

Batch EMAIL Command Setting	Resulting Disposition			
	Orig = K	Orig = D	Orig = H	Orig = L
SET DISP=HOLD	L	H	H	L
SET DISP=KEEP	K	D	H	L
SET DISP=RESPECT	Deleted	Deleted	Deleted	Deleted
SET DISP=DELETE	Deleted	Deleted	Deleted	Deleted

AUTOEMAIL Command Setting	Resulting Disposition			
	Orig = K	Orig = D	Orig = H	Orig = L
SET DISP=HOLD	L	Deleted	N/A	N/A
SET DISP=KEEP	L	Deleted	N/A	N/A
SET DISP=RESPECT	L	Deleted	N/A	N/A
SET DISP=DELETE	L	Deleted	N/A	N/A

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	<i>KEEP</i>
AUTOEMAIL	Accepted	For DISP = K: <i>HOLD</i> For DISP = D: <i>DELETE</i>
Batch	Accepted	<i>KEEP</i>

SET DUPES

The syntax is

```
SET DUPES=[ON|OFF]
```

When set to ON, this command permits the same attachment to be processed multiple times for the same session with the same name. The default is OFF.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET ECHO

The syntax is

```
SET ECHO=[ON|OFF]
```

This command enables display of all acknowledgement information sent from the SMTP server. The parameters are described in the following table:

Parameter	Description
ON	Enables display of all responses sent back to the client.
OFF	Only pertinent information is returned (the default).

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET EOJEXEC

The syntax is

```
SET EOJEXEC=membername
```

This command specifies the name of a script file that contains one or more commands. When the EMAIL program encounters a QUIT command, which would terminate the EMAIL client, the client invokes the script indicated in EOJEXEC prior to processing the QUIT, provided that the client terminates with a zero return code. The script file cannot contain the QUIT and EXEC commands.

In the following example, SET EOJEXEC is used in an AUTOEMAIL script. EOJEXEC is set to the script file *PRT.L*. Commands in the script send the e-mail's attachment, *REPORT1*, to a printer that is started in CLASS=T. The AUTOEMAIL script contains the following lines:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,DISP=D,LDEST=(*,RSB)
* $$ LST CLASS=A,DISP=D,DEST=(*,RSB)
// JOB LIBRCAT
// EXEC LIBR
ACC SUB=BIMLIB.CSICNFG
CATALOG RSBMAIL3.L REPLACE=YES EOD=/+
SET HOST=MAILSERVER.E-VSE.COM
(...other EMAIL commands)
SET EOJEXEC=PRT
CD POWER.LST.A
ATTACH REPORT1 AS DATA.TXT
QUIT
/+
/*
/&
* $$ EOJ
```

The script file *PRT.L* executes just prior to the QUIT and contains the following commands:

```
PALTER LST,REPORT1,DISP=D,CLASS=T
SAY REPORT1 has been redirected to the printer...
```

The variable is described in the following table:

Parameter	Description
<i>membername</i>	A script file in the search chain. This is an “L” type member, but the file type must be omitted from the command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET EXTTYPES

The syntax is

```
SET EXTTYPES=[ON|OFF]
```

This command specifies that the EMAIL client use the values in the EXTTYPES.L file instead of the SET BINARY setting to control which MIME type is sent to the SMTP server for the attached file. The default is ON.

A MIME type is a string that is sent to the SMTP server so that the recipient of the file (an e-mail client) can determine the best program to use for automatically opening the attachment. The MIME type also indicates whether the attachment data is plain text or if it was UUENCODED (a base64 translation of binary text into a longer string of ASCII characters) and needs to be UUDECODED back into a usable binary file.

The parameters are described in the following table:

Parameter	Description
ON	The EXTTYPES.L file is used to set the MIME type.
OFF	The SET BINARY setting is used to set the MIME type.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET FCB

The syntax is

```
SET FCB= [phasename | *NULL]
```

This command specifies the name of a phase that is to be used as the Forms Control Buffer (FCB). The FCB replaces the paper tape used by early model line printers and contains information for the line number associated with each carriage control character. If you omit this parameter, limited carriage control emulation is performed. The default values are explained below.

The variable is described in the following table:

Value	Description
<i>phasename</i>	This variable is the name of a valid FCB phase as defined to VSE. If the name you specify cannot be loaded in the TCP/IP FOR VSE partition, you receive an error message. If the name you specify can be loaded but is not an FCB, the results are unpredictable and are not likely to be the ones you want. You must set the carriage control option to <i>ON</i> (SET CC = ON) for the FCB parameter to be processed.
*NULL	Specify *NULL to eliminate any previously specified FCB value.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	The FCB parameter specified on the * \$\$ LST statement that initiated AUTOEMAIL
Batch	Accepted	No default

SET FCBPREFIX

The syntax is

```
SET FCBPREFIX= {string|*NULL}
```

This command specifies a different prefix for the POWER LST parameter name. If the name has the form “FCB=\$\$\$\$xxx”, the “\$\$\$” is replaced by *string*.

For example, if SET FCBPREFIX=PRT1 is used along with SET FCB=\$\$\$\$1234, then SET FCB=PRT11234 is generated by the EMAIL client. This command is used primarily in an environment where the FCB name is obtained from the POWER LST statement by AUTOEMAIL.

The variables are described in the following table:

Variable	Description
<i>string</i>	String to replace the “\$\$\$” prefix.
*NULL	Specify *NULL to remove the “\$\$\$” prefix.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	*NULL
AUTOEMAIL	Accepted	*NULL
Batch	Accepted	*NULL

SET FILETYPE

The syntax is

```
SET FILETYPE= {file_type|_}
```

This command specifies the file type of a script containing EMAIL commands. It allows the batch overriding of scripts. This command is useful at sites where a single library is used for multiple environments (for example, production and test) and a different file type is specified for each environment.

The variable is described in the following table:

Variable	Description
<i>file_type</i>	The defined file type of an EMAIL script file. The default is 'L'. For example, if PROG.TTT is a file containing EMAIL commands, FILETYPE could be set to TTT. In that case, if EXEC PROG is called, PROG.TTT is the member read from the VSE library.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	L
AUTOEMAIL	Accepted	L
Batch	Accepted	L

SET FROM

The syntax is

```
SET FROM=from-address
```

This command specifies the "FROM" field for your e-mail. This control is required. Note that if you specify an invalid address, you won't know it until you attempt to send the message.

The variable is described in the following table:

Variable	Description
<i>from-address</i>	The address to insert in the FROM field of your e-mail. The syntax depends on your SMTP server. Some servers accept almost anything. Others insist on a valid e-mail address, and still others try to authenticate the e-mail to eliminate forgeries. TCP/IP FOR VSE simply passes the value that you specify to the SMTP server.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Required	No default
AUTOEMAIL	Required	“VSE@ <i>default_domain</i> .COM,” where <i>default_domain</i> is set in the initialization deck.
Batch	Required	No default

SET HOST, SET IPADDR, SET IP

The syntax is

```
SET HOST={ipaddress | ipname}
SET IP={ipaddress | ipname}
SET IPADDR={ipaddress | ipname}
```

This command specifies the IP address of the host on which the SMTP daemon is running. To run EMAIL, you must specify the address of the SMTP daemon.

The variables are described in the following table:

Variable	Description
<i>ipaddress</i>	The IP address of the SMTP daemon
<i>ipname</i>	The domain name that represents the IP address

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Required	No default
AUTOEMAIL	Accepted	Taken from the * \$\$ LST or * \$\$ PUN statement as defined by the HOSTNAME parameter on the DEFINE EVENT command
Batch	Required	No default

The anticipated response is as follows:

```
SET HOST=sntp-server.columbus.rr.com
204.210.252.023
EMAIL Ready:
```

The second line of the response varies depending on whether you specify *ipaddress* or *ipname*. If you specify *ipaddress*, the associated IP name, if applicable, is returned. If you specify *ipname*, the associated IP address is returned.

SET INSERTS

The syntax is

```
SET INSERTS= phasename
```

This command identifies a phase whose contents are to be transmitted along with the attached file. This enables you to include printer control data or other modifications to the attached file before the file, after the file, and after each form feed. See [“Controlling Printers with the INSERTS Phase”](#) in chapter 4, “Printing Files with LPR,” for more information on how to use the INSERTS facility.

Although you probably do not want to insert PCL5 codes into your e-mail, you might want to use an INSERTS phase to insert page delimiters or other e-mail identifiers.

Variable	Description
<i>phasename</i>	The name of a phase that is built with the INSERTS macro. If the name you specify cannot be loaded in the TCP/IP FOR VSE partition, you receive an error message. If the name you specify can be loaded but is not an INSERTS phase, the results are unpredictable and are unlikely to be the ones you want.

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	Taken from the UCS parameter on the * \$\$ LST statement that initiated the AUTOEMAIL
Batch	Accepted	No default

SET JOIN

The syntax is

```
SET JOIN=[ON|OFF]
```

This command specifies that if there are multiple text attachments, the attachments are combined into a single attachment. The default is OFF.

The parameters are described in the following table:

Parameter	Description
ON	Multiple text attachments are combined.
OFF	Multiple text attachments are not combined.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET LANGUAGE

The syntax is

```
SET LANGUAGE=membername
```

This command instructs the EMAIL client to use a special language file for countries that use a single-byte character set (SBCS) that is not part of the generally used alphabet. The language file maps English alphabetic characters in a text message to hex values corresponding to characters of an alternate language such as Arabic or Hebrew. This command affects only the body text of an email message; attachments are ignored. There is no default for *membername*.

The first line of the language file contains an ISO character-set information string and a label. The label denotes the version, which is specific to an operating system. The second line contains an escape character (grave accent mark) for enabling and disabling language conversion. Both English and the alternate language may be included in a message by using the escape character to control conversion.

For example, to construct a dual-language message in English and Hebrew, LANGUAGE may be set to a file containing the Hebrew character set (Windows version). Part of this file appears as follows:

```
CHARSET=ISO-8859-8          STANDARD VISUAL-HEBREW CHARACTER SET
ESCAPE=`
&=E0      Aleph
a=E1      Beit
b=E2      Gimel
c=E3      Dalet
d=E4      Heh
e=E5      Vav
f=E6      Zayin
g=E7
...
(other characters)
...
t=F4      Peh
u=F5      Tzaddi Sofit
v=F6      Tzaddi
w=F7      Qoph
x=F8      Resh
y=F9      Shin
z=FA      Tav
```

The body of the message is written in both English and Hebrew by using the escape character (`) and characters representing the Hebrew alphabet. Such a message appears in the following box:

```
SET LANGUAGE=HEBREW
TEXT EOD=/+
<HTML>
<B>`ylem lj niyx&l - &lide bxeqno `</B>(Hello from Israel - Leo Langevin)
<p> P.S., I sent the message to both of your e-mail accounts in case your
company blocks HTML arriving from the outside.
/+
```

The command variable is described in the following table:

Variable	Description
<i>membername</i>	Language descriptor file. This is an “.L” type library member, but the file type must be omitted from the command. One file source is at www.columbia.edu/kernit/csettables.html .

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET LOCK

The syntax is

```
SET LOCK=[ON|OFF]
```

This command locks a POWER entry so that it cannot be accessed by another job during sending. The default is OFF.

SET LPASS

The syntax is

```
SET LPASS=password
```

This command specifies the password for the local security exit where you need a remote password as well. There is no default. For example, if you wanted to access the remote server and store the contents returned on your local VSE library, you would not only need a “SET PASS” for the remote system, but you would also need a “SET LPASS” if your security exit were to allow or prevent I/O on your local disk.

The variable is described in the following table:

Variables	Description
<i>password</i>	The local password

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET LUSER

The syntax is

```
SET LUSER=id
```

This command specifies the user identifier for the local security exit where you need a remote user ID as well. For example, if you wanted to access the remote server and store the contents returned on your local VSE library, you would not only need a SET USER for the remote system but also a SET LUSER if your security exit were to allow or prevent I/O on your local disk. There is no default for *id*.

The variable is described in the following table:

Variables	Description
<i>id</i>	The local user identifier

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET MAILSERVER

The syntax is

```
SET MAILSERVER= [ON|OFF]
```

This command directs the client to find the IP address of the recipient's e-mail server and use that for sending output. To do this, it performs a query on the DNS server, obtains the Mail Exchange record, and returns the IP address from that record. The IP address is then used for the "TO" address.

This function is useful when you don't have an SMTPD (Postfix SMTP server) of your own, or if the recipient rejects your e-mail address as spam because your VSE system is not recognized as a valid sending location. Some sites, such as AOL, may still consider your attempt to deliver e-mail as spam.

The parameters are described in the following table:

Parameter	Description
ON	The recipient's IP address is looked up and used.
OFF (default)	The recipient's IP address is not used.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET NEWNAME

The syntax is

```
SET NEWNAME=name
```

This command assigns a name to the attachment. There is no default for *name*.

The variables are described in the following table:

Variables	Description
<i>name</i>	The new attachment name

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET NOEJECT

The syntax is

```
SET NOEJECT= [ON|YES|OFF|NO]
```

This command suppresses the initial form feed character on an attached listing. Many print files begin with a page eject character. TCP/IP FOR VSE normally translates this character into a form feed, but on some printers this creates a blank page. There is no default.

The parameters are described in the following table:

Parameter	Description
ON YES	Suppresses the initial form feed character
OFF NO	Does not suppress the initial form feed character

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET NOTIFY

The syntax is

```
SET NOTIFY= e-mail_address
```

This command specifies where to send a failure notice if e-mail delivery fails. There is no default for *e-mail_address*.

The variables are described in the following table:

Variables	Description
<i>e-mail_address</i>	The destination address

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET PASS

The syntax is

```
SET PASS=password
```

This command specifies the remote password of the server to which you are connecting. When the server is an SMTP server, this command is forwarded when SET AUTH is used for e-mail authentication. There is no default for *password*.

The variable is described in the following table:

Variables	Description
<i>password</i>	The authentication string.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET PDF

The syntax is

```
SET PDF=[ON|OFF|name]
```

This command specifies that the text of the e-mail attachment(s) should be converted to PDF. The default is OFF. See [Chapter 8, “PDF Conversion Facility.”](#) for more information on converting files to PDF.

The parameters are described in the following table:

Parameters	Description
ON	The attachment text is converted using the default PDF configuration file <i>PDFSETUP.L</i> .
OFF	The attachment text is not converted.
<i>name</i>	The attachment is converted using the configuration file specified by <i>name</i> . This is an “.L” type library member, but the file type must be omitted from the command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET PRIORITY

The syntax is

```
SET PRIORITY=[HIGH|MEDIUM|LOW]
```

This command sets the e-mail’s urgency level for the recipient. There is no default.

The parameters are described in the following table:

Parameter	Description
HIGH MEDIUM LOW	The urgency level

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET REPLYTO

The syntax is

```
SET REPLYTO=e-mail_address
```

This command specifies an e-mail address for the “TO” field when the recipient replies to the e-mail. There is no default for *e-mail_address*.

The variable is described in the following table:

Variables	Description
<i>e-mail_address</i>	An alternative e-mail address

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET REPLYTOLIST

The syntax is

```
SET REPLYTOLIST=membername
```

This command specifies a list of e-mail addresses for the “TO” field when the recipient replies to the e-mail. There is no default for *membername*.

The variable is described in the following table:

Variables	Description
<i>membername</i>	The member that contains a list of addresses. This is an “.L” type library member, but the file type must be omitted from the command.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET SEPARATOR

The syntax is

```
SET SEPARATOR=[ON|OFF]
```

When attaching from POWER, ON indicates that any JSEP pages that the POWER driver reads are to be returned rather than ignored. A JSEP=*n* parameter must be added to the POWER LST card to specify the number of pages (*n*). If there are no JSEP pages, then this command has no effect. If the value (*n*) is not set, then the default value is based on the SEPARATOR_PAGES command setting. The default is OFF.

The parameters are described in the following table:

Parameters	Description
ON	Separator pages are inserted.
OFF	Separator pages are not inserted.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL.

SET SMTP

Synonym for SET MAILSERVER

SET SNAPDUMP

The syntax is

```
SET SNAPDUMP=[ON|OFF]
```

This command enables a PDUMP of all datagram traffic for this session. The dump data is placed in TCP/IP FOR VSE's SYSLST partition. Default is OFF.

The parameters are described in the following table:

Parameters	Description
ON	The PDUMP is enabled.
OFF	The PDUMP is disabled.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL.

SET SOSI

The syntax is

```
SET SOSI=indicator
```

This command sets the shift-in/shift-out (SOSI) translation indicators. There is no default. The SOSI indicators are hex values used in sending a double-byte character set (DBCS). A DBCS is required for a language that has more than 256 letters, such as Japanese. A shift-out hex value indicates when a string begins, and a shift-in value indicates when a string ends.

The variable is described in the following table:

Variable	Description
<i>indicator</i>	A hex string that controls DBCS translation

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET SUBJECT

The syntax is

```
SET SUBJECT=e-mail-subject
```

This command specifies the subject field for your e-mail. The TCP/IP FOR VSE EMAIL client does not require a subject, but some SMTP daemons might.

The variable is described in the following table:

Variable	Description
<i>e-mail-subject</i>	A text string up to 64 characters long

The following table shows which modes and defaults are valid for this command:

EMAIL Client Type	Command Status	Default
CICS	Accepted	No default
AUTOEMAIL	Accepted	The POWER listing name from the VSE listing
Batch	Accepted	No default

SET TLS

The syntax is

```
SET TLS=[ON|OFF]
```

This command tells the EMAIL client to check whether the server supports TLS. If it does, and if SSL/TLS has been enabled on your VSE, then EMAIL attempts to log in using an SSL/TLS handshake. If the server does not send a “TLS” response to an EHLO announcement by the client, this command is ignored. The default is OFF.

See the *TCP/IP FOR VSE Optional Features Guide* for more information on SSL/TLS.

**SET TO,
SET BLIND,
SET COPY**

The syntax is

```
SET TO=recipient-address
SET BLINDcopy=recipient-address
SET COPY=recipient-address
```

These commands specify the recipients of your e-mail. You must specify each recipient in a separate command and on a separate line, although some SMTP daemons accept multiple recipients on the same line if they are separated by semicolons. The commands are cumulative. This means that if you specify multiple SET TO, SET BLIND, and SET COPY commands, you send the e-mail to multiple recipients. The list of recipients is cleared when you issue the SEND command, so if you send multiple e-mails in the same job, you must reissue these commands. You can use an EXECUTE command to simplify this process. You can specify any number of recipients.

The SET TO command specifies the primary recipient. The SET BLINDcopy command specifies carbon copy recipients whose names should not appear on the TO or CC list. SET COPY specifies carbon copy recipients whose names appear on the CC LIST. There is no default for *recipient-address*.

The variable is described in the following table:

Variable	Description
<i>recipient-address</i>	An address to which you want to send the e-mail. Each address can be up to 64 characters long.

SET TO is required by CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients. SET BLIND and SET COPY are accepted by these clients.

SET TOLIST

The syntax is

```
SET TOLIST=membername
```

This command specifies a member containing a list of primary e-mail addresses. It replaces using the SET TO command multiple times to send to additional addresses. There is no default for *membername*.

The variable is described in the following table:

Variables	Description
<i>membername</i>	The member that contains the list of addresses. This is an “.L” type library member, but the file type must be omitted.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET TRANSLATE

The syntax is

```
SET TRANSLATE=table-name
```

This command specifies a translation table to be used for EBCDIC-to-ASCII translation on both the body of a message and any attached file. Translation occurs when VSE EBCDIC data is being sent because the recipient of the data requires that it be in ASCII.

Remember that there are two other translation commands (SET TRATT and SET TRBODY). If this is the only command used, then the other commands default to this value. That is, body-text and attachment translation both use this setting.

The variable is described in the following table:

Variable	Description
<i>table-name</i>	<p>The name of a valid translate table. The table must be available to TCP/IP through the DEFINE TRANSLATIONS command, as explained in the <i>TCP/IP FOR VSE Installation Guide</i>. You can use the QUERY TRANSLATIONS console command to view a list of valid translate tables.</p> <p>If you specify an invalid <i>table-name</i>, the EMAIL client prints a message and uses the default table-name. The e-mail job does not terminate because of an invalid value for <i>table-name</i>.</p>

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET TRATT

The syntax is

```
SET TRATT=table-name
```

This command specifies a translate table for EBCDIC-to-ASCII translation on attached files. This definition overrides the table name set by the SET TRANSLATE command only for attachments, and only if the attachments are text files.

If a NULL translation table is specified, then the attachment is not translated. This control can be useful when the desired effect cannot be achieved through the SET BINARY command or by modifying the definitions in the EXTYPES.L table.

The variable is described in the following table:

Variable	Description
<i>table-name</i>	<p>The name of a valid translate table. The table must be available to TCP/IP FOR VSE through the DEFINE TRANSLATIONS command as specified in the <i>TCP/IP FOR VSE Installation Guide</i>. You can use the QUERY TRANSLATIONS console command to obtain a list of valid translate tables. If you specify an invalid <i>table-name</i>, the EMAIL client prints a message and uses the default table name. The e-mail job does not terminate if an invalid table name is specified.</p> <p>To prevent translation of attachments, use “NULL” as the table name.</p>

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET TRBODY

The syntax is

```
SET TRBODY=table-name
```

This command specifies the name of a translate table to be used for EBCDIC-to-ASCII translation on the message body only. This definition overrides the table name set by the SET TRANSLATE command for the body text. Translation to ASCII occurs unless a NULL table is specified.

The variable is described in the following table:

Variable	Description
<i>table-name</i>	<p>The name of a valid translate table. The table must be available to TCP/IP FOR VSE through the DEFINE TRANSLATIONS command, as specified in the <i>TCP/IP FOR VSE Installation Guide</i>. You can use the QUERY TRANSLATIONS console command to obtain a list of valid translate tables. If you specify an invalid <i>table-name</i>, the EMAIL client prints a message and uses the default table name. The e-mail job does not terminate if an invalid table name is specified.</p> <p>To prevent translation of attachments, use “NULL” as the table name.</p>

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET TRUNC

The syntax is

```
SET TRUNC=[ON|OFF]
```

This command prevents sending trailing blanks in text attachments. The default is OFF.

The parameters are described in the following table:

Parameters	Description
ON	Trailing blanks are truncated.
OFF	Trailing blanks are not truncated.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET USER

The syntax is

```
SET USER=name
```

This command specifies the remote user identifier for the server you are connecting to. When the server is an SMTP server, this command is forwarded when SET AUTH is used for e-mail authentication. There is no default for *name*.

The variable is described in the following table:

Variable	Description
<i>name</i>	The remote user name

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

**SET USERID,
SET PASSWORD**

The syntax is

```
SET USERID=tcPIP-userid
SET PASSWORD=tcPIP-password
```

These commands identify the user to the TCP/IP FOR VSE security system. If the system administrator has activated TCP/IP FOR VSE security, you must use a valid user ID and password to access the TCP/IP FOR VSE file system. You need to access the file system if you are sending an attachment. There are no defaults.

The variables are described in the following table:

Variable	Description
<i>tcpip-userid</i>	A valid user ID that is known to the TCP/IP FOR VSE security system. The user ID can be up to 31 characters long. If you specify an invalid user ID, you are not notified until you attempt to access the file system.
<i>tcpip-password</i>	A valid password that is known to the TCP/IP FOR VSE security system. The password can be up to 16 characters long.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET VALIDATE

The syntax is

```
SET VALIDATE=[ON|OFF]
```

This command strips out any non-displayable characters from text attachments. The default is OFF.

The parameters are described in the following table:

Parameters	Description
ON	Non-displayable characters are deleted.
OFF	Non-displayable characters are allowed.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

SET VCARD

The syntax is

```
SET VCARD=card_name
```

This command enables the automatic attaching of a VCARD (VCF file), which is a Windows® identification system. A VCF file contains information related to the sender, and the recipient can simply click on it to add the information to their address book. The EMAIL client tries to attach the VCF file to the end after all other attachments are complete. There is no default for *card_name*.

The variables are described in the following table:

Variable	Description
<i>card_name</i>	A one- to eight-character name of a VCF binary file in a VSE library that is part of the TCP/IP LIBDEF search chain.

This command is accepted by the CICS EMAIL transaction, AUTOEMAIL, and batch EMAIL clients.

E-mail Examples

The examples in this section are shown as batch jobs, but you can use the commands in any of the supported environments.

Example 1

This job contains an e-mail that is sent from an accounting department.

```
* $$ JOB JNM=EMAIL,CLASS=Y,DISP=D
* $$ LST CLASS=C,DEST=(*,LEO)
// JOB EMAIL
// LIBDEF *,SEARCH=PRD2.TCPIP
// EXEC CLIENT,SIZE=CLIENT,PARM='APPL=EMAIL,ID=00'
SET HOST=SMTP-SERVER.COLUMBUS.RR.COM
SET FROM=Accounting Department
SET TO=Acct@tcpip4vse.com
SET Subject=Expense Report
TEXT
I noticed that you have a $50 expense here for your trip to
Universal Studios under the category "Customer Visit." Is this
company business? Attached is the denial.
/+
CD PRD2.NOTES
SEND DENIAL.TXT
QUIT
/*
/&
* $$ EOJ
```

Example 2

In this example, Eric runs accounting jobs every night, and the jobs create three output files. Eric wants to use e-mail to send two output files to a manager and all three output files to himself.

```
* $$ JOB JNM=EMAIL,CLASS=Y,DISP=D
* $$ LST CLASS=C,DEST=(*,LEO)
// JOB EMAIL
// LIBDEF *,SEARCH=PRD2.TCPIP
// EXEC CLIENT,SIZE=CLIENT,PARM='APPL=EMAIL,ID=00'
SET HOST=SMTP-SERVER.COLUMBUS.RR.COM
SET FROM=DACCT@mycompany.com
SET TO=manager@company.com
SET Subject=Your Daily Accounting Report.
TEXT
This is the daily accounting report. If you have questions, please
call me at 614 555-1000 x212
/+
CD POWER.LST.A
ATTACH REPORT1
ATTACH REPORT2
SEND
SET TO=eric@company.com
ATTACH REPORT1
ATTACH REPORT2
ATTACH REPORT3
SEND
QUIT
/*
/&
* $$ EOJ
```

TCP/IP FOR VSE REXEC Client

Overview

TCP/IP FOR VSE includes a *Remote Execution Protocol* (REXEC) client. REXEC enables you to issue a command to a foreign TCP/IP system that is running a REXEC daemon. Before using REXEC, you should verify that the target platform is running the REXEC daemon. Then decide whether REXEC or telnet would best satisfy your requirement. The following chart contrasts these two facilities:

REXEC	Telnet
REXEC is oriented toward single commands and is best if you want to issue only one command.	Telnet is oriented toward multiple commands. It is the best option if you want your session to mimic a user in front of a terminal that is connected to the foreign system.
The login protocol is managed for you. REXEC requires a user ID and password on the foreign system, but the mechanics of logging on to that system are masked from you.	You manage the login protocol. Because telnet is basically a session at the foreign host, you need to issue the proper commands to log on, supply the password, issue commands, wait for responses, and so on.
Many systems do not have a REXEC daemon, so it could be hard to find one.	Most systems support some form of telnet, so it is easy to find one.

You can use TCP/IP FOR VSE's REXEC client to issue commands on foreign hosts. In this chapter we discuss the following topics:

- Setting up the REXEC client.
- Running the REXEC client. There are three ways to start this client.
- Sending REXEC commands.
- Using REXEC client commands. This chapter includes a summary of the REXEC commands and a detailed explanation of each command

Setting Up the REXEC Client

Before you can use REXEC, you must make sure that the foreign system is running a REXEC daemon and that you are authorized to use it.

Running the REXEC Client

You can run the REXEC client using any of the following three methods:

- As a CICS transaction
- In batch, using the REXEC batch client.
- In an application program, by using the REXX, Assembler, COBOL, or PL/1 Sockets interface.

We describe the first two methods in this chapter. For information about using the Sockets interface, see the *TCP/IP FOR VSE Programmer's Guide*.

Using CICS REXEC

To start the REXEC client from your CICS terminal, log on to CICS and issue the following command:

```
REXEC
```

There are no parameters. REXEC uses SYSID 00, and you cannot use a different SYSID.

REXEC responds by displaying the following messages on your screen:

```
REXEC
TCP200I Client -- Startup --
TCP207I Copyright (c) 1995-2006 Connectivity Systems Incorporated
TCP202I Attempting to Establish Connection
TCP204I Connection has been Established
Client manager connection Established.
REXEC Ready:
```

The REXEC client is now in command mode and can accept REXEC client commands. The commands are explained later in this chapter. To terminate the REXEC client from your CICS terminal, use the QUIT command.

Using Batch REXEC

There may be times when you want to run REXEC from a batch environment. The REXEC client has a batch mode that allows you to do this. The JECL statements in the following box show how to use the batch facility:

```
* $$ JOB JNM=REXEC,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB REXEC
// EXEC CLIENT,PARM='APPL=REXEC,ID=nn,QUIET=YES'
REXEC command 1
REXEC command 2
REXEC command n
/*
/&
* $$ EOJ
```

Parameters

The parameters are keyword parameters, so you can specify them in any order. They are described in the following table:

Parameter	Description
APPL=REXEC	This parameter is required to run REXEC.
ID= <i>nn</i>	Specifies the system ID of the TCP/IP partition that you want to connect to. Remember that you can have more than one copy of TCP/IP FOR VSE running at one time (for example, production and test). The default is <i>00</i> .
QUIET=YES	Suppresses informational messages during execution of the batch REXEC client. (Note: <i>YES</i> is the only valid value.)

Commands

You can issue any number of REXEC commands using the batch REXEC client. You must use SET commands to specify the user ID, password, and host before you issue a COMMAND *command*, which starts a command on the remote client. (If you do not issue a COMMAND command, the whole exercise is probably not worth the effort.) For a summary of the REXEC commands, including the COMMAND command, see the section “[Using REXEC Client Commands](#)” (page [7-7](#)).

Return Codes The following table shows the return codes that apply to these commands:

Return Code	Description
0	Your REXEC commands completed successfully. Note: If you do not issue the COMMAND command at some point, you may receive a return code of 0 even though no REXEC command is sent.
4	A syntax error occurred in one of the REXEC commands in the job stream. To find the problem, study the output. A common problem is the inability of the foreign host to process a command.
8	An error occurred in one of the REXEC commands in the job stream. The error could be a syntax error or it could be another type of error. To find the problem, study the output. One or more REXEC commands might have completed successfully, so you need to determine the current status before you rerun the job.
12	One of the parameters was incorrectly specified.
16	The batch partition is unable to establish a session with TCP/IP FOR VSE. Verify that your ID= parameter value corresponds to the SYSID for an active TCP/IP FOR VSE partition.

Sending REXEC Commands

To send REXEC commands from VSE, follow these steps:

1. Start the REXEC client using one of the methods described in the previous section.
 2. Issue REXEC client commands to identify the host, user ID, password, and COMMAND commands.
 3. Issue additional REXEC client commands to specify additional parameters.
 4. Stop the REXEC client.
-

Using REXEC Client Commands

The TCP/IP FOR VSE REXEC facility provides a set of commands that you can use to direct the REXEC operation. In general, these commands are valid from all REXEC applications, including CICS and batch. The commands fall into the following three categories:

- **SET commands.** These commands set options that need to be in place before you issue the **COMMAND** command. You must use **SET** commands to specify the REXEC daemon's IP address, a user ID, and a password before you issue the **COMMAND** command.
- **The COMMAND command.** This command starts REXEC on the remote client.
- **The WAIT command.** This command pauses a specified number of seconds after the preceding command before processing the following command.

All commands are summarized in the following table. The **SET** commands are listed first. A detailed explanation of most commands follows this table.

Command	Description
SET BADMATCH	Specifies a string that should not be returned by the server. If the string is not found, then RC=0; otherwise, RC=4.
SET DEBUG	Turns extended messages ON or OFF.
SET GOODMATCH	Specifies a string that should be returned by the server. If the string is found, then RC=0; otherwise, RC=4.
SET HOST SET IP SET IPADDR	Specifies the IP address of the host on which the REXEC daemon is running. You can specify this value in dotted decimal notation or as a domain name.
SET TIMEOUT	Specifies the connection timeout period.
SET TRANSLATE	Specifies the name of a translate table to be used when communicating with the foreign REXEC daemon.
SET USERID SET PASSWORD	The user ID and password on the foreign host. Most REXEC daemons require authorization before you can use them.
COMMAND	Specifies the command to be issued to the foreign host. You can specify the COMMAND command as many times as you want in a single job. See also the WAIT command.
EXEC	Reads and executes client commands contained in a VSE library member.

Command	Description
QUERY	Lists the values specified with the SET command.
QUIT	Stops the REXEC client.
SETVAR	Sets variables for batch processing.
WAIT	Causes execution to pause for the specified amount of time.

SET HOST, SET IPADDR, SET IP

The syntax is

```
SET HOST={ipaddress|ipname}
SET IP={ipaddress|ipname}
SET IPADDR={ipaddress|ipname}
```

This command specifies the IP address of the host on which the REXEC daemon is running. To use REXEC, you must specify the address of the REXEC daemon.

The variables are described in the following table:

Variable	Description
<i>ipaddress</i>	The IP address of the REXEC daemon.
<i>ipname</i>	The domain name that represents the IP address.

The following table shows which modes and defaults are valid:

REXEC Client Type	Command Status	Default
CICS	Required	None
Batch	Required	None

The anticipated response is as follows:

```
SET HOST=VM
192.168.000.007
REXEC Ready:
```

The second line of the response varies depending on whether you specify *ipaddress* or *ipname*. If you specify *ipaddress*, the associated IP name, if applicable, is returned. If you specify *ipname*, the associated IP address is returned.

SET TIMEOUT

The syntax is

```
SET TIMEOUT = [54000|value]
```

This command allows you to specify the connection timeout period. The default period is 3 minutes. This default normally provides enough time for a command to execute on the other side and return data. If a command you send spawns a long process, however, more time may be needed.

The *value* you specify must be 300 times the number of seconds you want to maintain the active connection. For example, the default value is 54000 (3 minutes), which is 3 minutes × 60 seconds × 300.

The variable is described in the following table:

Variable	Description
<i>value</i>	The number of 1/300 th -second intervals indicating the maximum duration of the client-server connection.

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Accepted	54000
Batch	Accepted	54000

The anticipated response is as follows:

```
SET TIMEOUT=180000
REXEC Ready:
```

SET TRANSLATE

The syntax is

```
SET TRANSLATE=table-name
```

This command specifies the name of a translate table to be used for EBCDIC-to-ASCII translation on commands to be executed and on the responses coming back from the foreign system.

The variable is described in the following table:

Variable	Description
<i>table-name</i>	<p>The name of a valid translate table. The specified table must be available to TCP/IP FOR VSE through the DEFINE TRANSLATIONS command, as specified in the <i>TCP/IP FOR VSE Installation Guide</i>. You can use the QUERY TRANSLATIONS console command to obtain a list of valid translate tables.</p> <p>If you specify an invalid <i>table-name</i>, the REXEC client prints a message and uses the default <i>table-name</i>. The REXEC job does not terminate because of an invalid <i>table-name</i>.</p>

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Accepted	None
Batch	Accepted	None

The anticipated response is as follows:

```
SET TRANSLATE=US_ENG_03
REXEC Ready:
```

**SET USERID,
SET PASSWORD**

The syntax is

```
SET USER = foreign-userid
SET PASSWORD = foreign-password
```

These commands identify the user to the *foreign system*. Most REXEC daemons require some form of authentication, although some accept anonymous commands.

The variables are described in the following table:

Variable	Description
<i>foreign-userid</i>	A valid user ID on the foreign system. The user ID can contain up to 32 characters.
<i>foreign-password</i>	A valid password on the foreign system. The password can contain up to 16 characters.

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Accepted	None
Batch	Accepted	None

The anticipated responses are as follows:

```
SET USERID=JOHN
REXEC Ready:
SET PASSWORD=Xetuyohj
REXEC Ready:
```

COMMAND

The syntax is

```
COMMAND remote-command
```

This command initiates the REXEC operation. Before you issue the COMMAND command, you must issue the SET HOST, SET USERID, and SET PASSWORD commands. The response is returned to SYSLST, the CICS terminal, or the program if REXEC was invoked using the programming interface.

The variable is described in the following table:

Variable	Description
<i>remote-command</i>	The command to be sent to the remote system.

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Required	None
Batch	Required	None

The anticipated response depends on the REXEC daemon. The following sample response is from the REXEC daemon provided with VM/ESA:

```
command q cmslevel
Establishing connection with REXEC Daemon
Connection Established, Logging in.
CMS Level 15, Service Level 903
Closing the connection
REXEC Ready:
command q cplevel
Establishing connection with REXEC Daemon
Connection Established, Logging in.
VM/ESA Version 2 Release 4.0, service level 9903
Generated at 08/21/09 08:46:44 EDT
IPL at 07/06/09 09:23:45 EDT
Closing the connection
REXEC Ready:
command id
Establishing connection with REXEC Daemon
Connection Established, Logging in.
HPR      AT SYS1      VIA RSCS      09/05/12 21:30:23 EDT   WEDNESDAY
Closing the connection
REXEC Ready:
```

EXEC

The syntax is

EXEC <i>membername</i> E <i>membername</i>

This command enables you to process a series of REXEC commands that are contained in a script file. When REXEC finds an EXECUTE command, it stops processing the commands in the input stream. It immediately processes the commands listed in *membername* and then resumes processing the suspended input stream. This means that the statements in *membername* are logically included in the input stream as opposed to being a subroutine call. If *membername* contains a QUIT command, the session actually terminates.

The variable is described in the following table:

Variable	Description
<i>membername</i>	An “L” book that is contained in the search chain for the TCP/IP FOR VSE partition. REXEC automatically appends the “.L” suffix, so you do not need to specify it.

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Accepted	None
Batch	Not accepted	None

HELP

The syntax is

```
HELP [command-name]
```

This command displays the syntax of the specified REXEC command.

The variable is described in the following table:

Variable	Description
<i>command-name</i>	The name of the command you want to display. This value is optional. If you don't specify a name, a brief description of each command is displayed.

The following table shows which modes and defaults are valid for this command:

REXEC Client Type	Command Status	Default
CICS	Accepted	Displays a brief description of all commands
Batch	Not accepted	None

WAIT

The syntax is

```
WAIT interval[S]
```

This command pauses processing after the preceding command. This pause gives the server time to log the previous server request off before allowing the command following the WAIT to execute.

The variable is described in the following table:

Variable	Description
<i>interval</i>	A number indicating the length of time to pause execution. If the "S" suffix is used, the value is read as seconds. If the "S" suffix is not used, the value is the number of 300 th -second units. For example, WAIT 1500 (5 seconds) is equivalent to WAIT 5S.

REXEC Examples

This section contains two examples of REXEC jobs. The examples are batch jobs, but you can use the commands in any of the supported environments.

Example 1

Leo submitted a job based on VSE conditional JCL. Mysteriously, Leo prefers the CMS environment to ICCF, and so the job resides on VM. Leo uses the REXEC client to log on to VM and submit the job as shown in this example:

```
* $$ JOB JNM=REXEC,CLASS=Y,DISP=D
* $$ LST CLASS=C,DEST=(*,LEO)
// JOB REXEC
// LIBDEF *,SEARCH=PRD2.TCPIP
// EXEC CLIENT,SIZE=CLIENT,PARM='APPL=REXEC,ID=00'
SET HOST=VM
SET USER=LEO
SET PASSWORD=YEARIGHT
COMMAND SUBMIT MYJCL JOB
QUIT
/*
/&
* $$ EOJ
```

Example 2

Eric must watch the Windows® environment. He has an object REXX program that runs on a Windows operating system. The program combines the event logs and sends them to VSE. The name of the REXX program is *EVENTVSE*. He needs to run the program on three Windows systems. He can use the REXEC client to run the REXX program on all three machines based on a VSE schedule. The sample job is as follows:

```
* $$ JOB JNM=REXEC,CLASS=Y,DISP=D
* $$ LST CLASS=C,DEST=(*,ERIC)
// JOB REXEC
// LIBDEF *,SEARCH=PRD2.TCPIP
// EXEC CLIENT,SIZE=CLIENT,PARM='APPL=REXEC,ID=00'
SET HOST=NT1
SET USER=ERIC
SET PASSWORD=ERICNT1
COMMAND REXX EVENTVSE
SET HOST=NT2
SET USER=ERIC
SET PASSWORD=ERICNT2
COMMAND REXX EVENTVSE
SET HOST=NT3
SET USER=ERIC
SET PASSWORD=ERICNT3
COMMAND REXX EVENTVSE
QUIT
/*
/&
* $$ EOJ
```


PDF Conversion Facility

Overview

TCP/IP FOR VSE includes a *PDF Conversion Facility*. This facility allows VSE to produce nice-looking forms automatically.

The following methods invoke the PDF Conversion Facility:

- EMAIL—the SMTP client
- FTP—the File Transfer Protocol client or server
- LPR—the Line Printer Requester client
- HTTPD—the Web server

Because the PDF Conversion Facility is a stand-alone module that these other components can call, all data conversion from text to PDF format is handled externally, using 31-bit GETVIS whenever possible. The exceptions to this are old file I/O methods (such as LIBR) that are not 31-bit compatible, or file I/O methods (such as POWER) that degrade performance, where I/O throughput is faster in 24-bit storage.

So why would you want to convert text data to PDF?

PDF allows the data to be viewed in its intended format. This means that a fixed-length font is seen as fixed length, form-feeds are seen as form-feeds, and so forth. You can customize the look using a configuration command, as explained later.

Using the PDF Conversion Facility is intuitive. As more features are added to this product, no changes are needed by the user, with the exception of reassembling the PDFSETUP modules to take advantage of these new features.

Because the PDF Conversion Facility is a product of CSI International, it was inserted cleanly with the existing components, and it does not require special configuration. In fact, it works directly without needing any changes at all. Of course, if you want, you can easily modify the page size or change other default parameters.

The PDF Conversion Facility is fully reentrant and runs in 31-bit storage. It can communicate with every file I/O system that TCP/IP FOR VSE supports, including user-written file I/O drivers.

Keep in mind that if you do not need to convert a document to PDF, passing the data as plain text is better because there is far less data to transmit.

Enabling Conversion

The way you enable PDF conversion for a file depends on the component you are using—FTP, EMAIL, LPR, or the HTTP daemon (HTTPD).

FTP, EMAIL, LPR

The commands to enable PDF conversion in FTP, EMAIL, and LPR are as follows:

Client	Command
FTP	[QUOTE] SITE PDF {ON <i>pdf_configuration_member</i> }
EMAIL, LPR	SET PDF={ON <i>pdf_configuration_member</i> }

When either command is set to ON, the PDF Conversion Facility uses the default PDF configuration member, *PDFSETUP.L*, to control conversion formatting.

When either command is set to *pdf_configuration_member*, the conversion facility uses the parameter settings you specify in this custom configuration member. See “[Creating the Configuration File](#)” (page 8-7) for details on specifying parameters in a custom member.

Note: When specifying a configuration member, do not include the “.L” suffix in the file name. This means that you would specify *MYCONFIG*, for example, rather than *MYCONFIG.L*.

With FTP, the FTP server on the TCP/IP FOR VSE host converts the data as it is being transmitted to the remote user. (The remote host in this case is usually a PC equipped with software that enables viewing the PDF file.) You can even run the FTP client on VSE to talk to the FTP server on the same machine, copying the data from one location (such as the POWER print queue) to another (such as a VSE library) so that it can be accessed from the Web.

With EMAIL, the client converts the data from text to PDF. It then UUENCODes the resulting file using the BASE64 algorithm because, for all practical purposes, the PDF file is a binary file. Because BASE64 is used, the resulting file is 33 percent larger than a plain-text file. This is an EMAIL protocol restriction and does not apply to FTP. Therefore, the EMAIL PDF attachment is always larger and takes more time to send than if the data were sent using FTP. Because of the additional transfer size and time, you must determine the best way to transmit a given file.

Carriage Control Formatting

To maintain the form-feed structure of POWER files in the PDF file, you can set the client's CC option to ON. This tells the PDF Conversion Facility to use the carriage control (CC) characters in the first position of each record:

CC Character	Effect
1	Generates a form feed
(blank)	Skips one line
0	Skips two lines
- (hyphen)	Skips three lines
+	Skips zero lines (retains vertical positioning while resetting horizontal positioning)

You can also set the CC value in a custom PDF configuration member. The value you set in this member takes precedence over the CC value in the calling client program (FTP, LPR, or EMAIL). See "[CC={ON/OFF}](#)" (page [8-8](#)) for more information.

HTTP Daemon

With HTTPD, there are two special field names that allow you to transfer the data. They are as follows.

```
CSI_PDF={ON|OFF|configuration_file_name}
CSI_CC={ON|OFF}
```

You can pass the PDF and CC settings to HTTPD by including the CSI_PDF= and/or CSI_CC= fields either within the webpage or in the URL address bar in your Web browser. (These fields are recognized only by TCP/IP FOR VSE's HTTP daemon.)

For example, to transform a report in the POWER queue so that you can view it as a PDF file in a Web browser, you could enter the following address in the browser's address bar:

```
HTTP://url/POWER/LST/X/REPORT.01234?CSI_PDF=ON
```

In this example,

- the *url* is the host's Web address (such as *www.domain-name.com*) or IP address. In this case, the value would be the IP address of the TCP/IP FOR VSE stack on VSE.
- *REPORT.01234* is the report name in the POWER.LST.X queue.
- the question mark ("?") is a reserved character that separates the path from the first option (CSI_PDF=ON). You use an ampersand ("&") to separate additional options.

You also can put a hyperlink with this same syntax in your webpage. Or you can put a hyperlink in your webpage and set the field names as follows:

```
<HTML>
<TITLE>Test</TITLE>
<BODY>
<INPUT TYPE="hidden" NAME="CSI_PDF" value="ON">
<INPUT TYPE="hidden" NAME="CSI_CC" value="OFF">
Click <A HREF="myplace.com/power/lst/x/REPORT">HERE</a> to review the data
</BODY>
</HTML>
```

In this manner, you can easily transmit PDF reports back to the user without the user even knowing that they were not PDF reports in the first place. This is also true with CGI scripts; you can write a program in any CGI-supported language and send PDF data back to the user.

The following example shows a REXX CGI script that displays the contents of the print queue.

Note: This example was designed with a specific version of the VSE/POWER queue response. As you use later versions of VSE, the format will probably change, and you will need to adjust the logic of the program accordingly.

This example allows you to click on either the TEXT or PDF hyperlink in the resulting webpage. You then would view the POWER queue entry in either PDF or TEXT format, depending on what you selected.

```
CATALOG VSEPWR.PROC REP=Y EOD=/+
/* GET THE PASSED PARAMETERS */
USERID=ARG(1)
PASSWORD=ARG(2)
DATA=ARG(3)
INLEN=LENGTH(DATA)
/* RETURN THE WEB PAGE HEADINGS */
X=HTML(' <HTML><HEAD><TITLE>' )
X=HTML(' VSE POWER LST QUEUE DISPLAY' )
X=HTML(' </TITLE></HEAD>' )
X=HTML(' <BODY TEXT="#993300" BGCOLOR="#66FF99">' )
X=HTML(' <CENTER><H2><B><I><FONT COLOR="#000000">' )
X=HTML(' VSE POWER LST QUEUE DISPLAY' )
X=HTML(' </FONT></I></B><H2><P><HR>' )
X=HTML(' <FONT COLOR="#000066">' )
X=HTML(' <TABLE BORDER=1>' )
X=HTML(' <TBODY><INPUT TYPE="HIDDEN" NAME="CSI_CC" VALUE="ON">' )
X=HTML(' <TR><TD>View</TD><TD>View</TD><TD><B>Job Name</B></TD><TD>' )
X=HTML(' <B>Job No.</TD><TD><B>Pri</B></TD><TD><B>Disp</TD><TD>' )
X=HTML(' <B>Class</B></TD><TD><B>Pages</TD><TD><B>Date</TD></TR>' )
(continued)
```

```

/* INSERT THE LST QUEUE */
CALL OUTTRAP OUT.
ADDRESS POWER
'SETUID REXXPWR1'
'PDISPLAY LST,FULL=YES'
IF OUT.0>1 THEN DO
  DO I=2 TO OUT.0 BY 2
  'trace I'
  x=HTML('<TR><TD>')
  J=I+1
  PARSE VAR OUT.I MSG JOB JOBNUM PRI DISP CLASS PAGES IGNORE
  PARSE VAR OUT.J DATE1 IGNORE
  PDATE=SUBSTR(DATE1,3)
  LINKTXT='POWER/LST/'||CLASS||'/'||JOB||'/'||JOBNUM
  LINKPDF='POWER/LST/'||CLASS||'/'||JOB||'/'||JOBNUM||'?CSI_PDF=ON'
  part1a = '<A HREF="'||LINKTXT||'">Text</a></td><TD>'
  part1b = '<A HREF="'||LINKPDF||'">PDF</a></td>'
  part2='<TD>'||JOB||'</TD><TD>'||JOBNUM||'</TD><TD>'||PRI||'</TD><TD>'
  part3=DISP||'</td><td>'||CLASS||'</TD><TD>'||PAGES||'</TD>'
  part4='<TD>'||PDATE||'</TD></TR>'
  X=HTML(part1a)
  X=HTML(part1b)
  X=HTML(part2)
  X=HTML(part3)
  X=HTML(part4)
END
/* INSERT THE WEB FOOTER */
X=HTML('</TABLE></BODY></HTML>')
END
EXIT
/+

```

Finally, because the PDF Conversion Facility communicates with TCP/IP FOR VSE's file I/O system, you can generate a PDF from any type of file. This includes files from POWER, LIBR, and VSAM.

Creating the Configuration File

By default, the PDF Conversion Facility converts the file and sends it in landscape orientation using a 12-point Courier font. It also forces a page break after 45 lines or whenever it encounters a form feed. To change one more formatting parameters, you must set values in a configuration file.

The default file *PDFSETUP.L* was cataloged in your TCP/IP FOR VSE library during product installation. To use a PDF configuration file other than the default "PDFSETUP," you need to create one. The file must be located in a VSE sublibrary that is part of the TCP/IP FOR VSE LIBDEF search chain.

When the conversion begins, the PDF Conversion Facility checks whether the default configuration member exists. If the default member exists, the facility uses it only if no PDF configuration name was specified. For example, issuing "SET PDF=ON" in EMAIL invokes the default member, while issuing "SET PDF=LAND811C" invokes *LAND811C.L*, assuming this member was cataloged. If *PDFSETUP.L* is missing, the standard default values are used.

Note: Always create and use a custom member rather than edit *PDFSETUP.L*. Make only the changes that are needed to achieve the results you want.

The following JCL shows how to catalog a custom configuration member. The parameters included in this member are set to their default values.

```
* $$ JOB JNM=CATAL,DISP=D,CLASS=0
* $$ LST DISP=D,CLASS=A
// JOB CATAL
// EXEC LIBR
A S=PRD2.TCPIPCFG
CATALOG JOESETUP.L REP=Y
*           Joe's PDF Configuration File
* -----
FontName=Courier
FontSize=12
FontType=Normal
PaperName=Legal,Landscape
TopBorder=1
LeftBorder=1
BottomBorder=.5
MaxLinesPerPage=Auto
Rotate=0
* PaperSize Width=14 Height=8.5
* Author="John Doe"
* Subject="Power Report"
* Keywords="POWER REPORT VSEPROD PAYROLL"
VerticalSpace=12
* Matrix=(1 0 0 1 50 752)
/+
/*
/&
* $$ E0J
```

The following table describes the parameters you can set in a custom configuration member:

Parameter	Description
*	Any line that begins with an asterisk is considered a comment and is ignored. Blank lines are also ignored.
BOTTOMBORDER	<p>The size (in inches) of the bottom border only when MAXLINESPERPAGE=AUTO. The PDF Conversion Facility uses this size to calculate how many lines fit on a designated page. The default border size is 0.5".</p> <p>You can use decimal values (such as .5, 1.5, 1.55) with up to two digits to the right of the decimal point. For example, 1.66 is a valid value, but 1.651 is not.</p>
CC={ON OFF}	<p>Whether to use carriage-control (CC) data in column 1 of the report to maintain form-feed control in the generated PDF. ON uses the CC characters and removes them from the generated PDF file.</p> <p>OFF ignores the CC characters and leaves them in the PDF file.</p> <p>Note 1: This parameter's value takes precedence over the CC value in the calling client.</p> <p>Note 2: Do not use "CC=ON" if you use "STRIPCC=YES" in the PDF configuration file.</p> <p>See also "STRIPCC=[ON OFF]" (page 8-11).</p>
FONTNAME	<p>The font name to use. The only valid names are</p> <ul style="list-style-type: none"> • COURIER (default; recommended for reports) • TIMES-ROMAN • HELVETICA <p>No other text fonts will work.</p>
FONTSIZE	<p>The font size in points: a value from 10 to 72. The default is 12.</p>
FONTTYPE	<p>The font's style to use throughout the report. The valid values are</p> <ul style="list-style-type: none"> • NORMAL (the default) • BOLD • ITALIC • BOLD-ITALIC or ITALIC-BOLD

Parameter	Description
LEFTBORDER	<p>The size (in inches) of the border on the left side of the page. The default is 1.</p> <p>You can use decimal values (such as .5, 1.5, 1.55) with up to two digits to the right of the decimal point. For example, 1.66 is a valid value, but 1.651 is not.</p> <p>There is no RIGHTBORDER value because PDF does not support it. If the line is too long, it is truncated.</p>
MATRIX=(n n n n n n)	<p>If you are using the PAPERSIZE parameter and you need to manually adjust the type area matrix, you can do that here. This creates the PDF “Tm” setting as documented in the Adobe® PDF Reference manual. Typically, the first four values are “1 0 0 1.” The fifth value reflects the text area adjustment from the left-most border, in points. The sixth value indicates the total vertical text area in points. If the sixth value is too small, the text will be too near the bottom of the page. The larger the number, the higher the text position.</p> <p>Do not set this parameter unless you understand how PDF generation works. If invalid values are specified, the Adobe Reader software will not open the file.</p>
MAXLINESPERPAGE	<p>The maximum number of lines on a page. If you are not controlling the form feed, this parameter is especially helpful when you want the PDF Conversion Facility to automatically skip to the next page. The facility can determine the best number of lines per page if you set this parameter to AUTO instead of to a numeric value. The default is 45.</p> <p>When MAXLINESPERPAGE=AUTO, the PDF Conversion Facility converts the page height to points (72 points = 1 inch), deducts the top and bottom border values in points, divides the font size (plus 25 percent for leading—the spacing between lines), and rounds down the result. The result is the number of lines per page. The bottom border may not be the exact measurement specified because the conversion facility may need to eliminate a line to avoid an overflow. But the bottom border is close to the specified value.</p>

Parameter	Description																				
NOEJECT=[ON OFF]	<p>Whether to eject a page before printing the report. ON skips the initial blank page. OFF, the default, allows the blank page to be printed. (The page is printed only if STRIPCC is also OFF.)</p> <p>Note: This parameter's value takes precedence over the NOEJECT value in the calling client. Also, for HTTP, the value is always ON.</p>																				
PAPERNAME	<p>The form type. You can use one of the following standard names:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Size (inches)</th> </tr> </thead> <tbody> <tr> <td>A3</td> <td>11.69 × 16.53</td> </tr> <tr> <td>A4</td> <td>8.26 × 11.69</td> </tr> <tr> <td>A5</td> <td>5.85 × 8.26</td> </tr> <tr> <td>EXECUTIVE</td> <td>7.5 × 10</td> </tr> <tr> <td>LEDGER</td> <td>17 × 11</td> </tr> <tr> <td>LEGAL</td> <td>8.5 × 14</td> </tr> <tr> <td>LETTER</td> <td>8.5 × 11</td> </tr> <tr> <td>STATEMENT</td> <td>5.5 × 8.5</td> </tr> <tr> <td>TABLOID</td> <td>11 × 17</td> </tr> </tbody> </table> <p>If you want the paper to be in landscape mode, append “,LANDSCAPE” (note the comma) to the PAPERNAME value as in the JCL above.</p> <p>The default is “LEGAL,LANDSCAPE”.</p> <p>To specify a non-standard paper size, use the PAPERSIZE parameter.</p> <p>Note: If you specify a PAPERSIZE, do not specify a PAPERNAME.</p>	Name	Size (inches)	A3	11.69 × 16.53	A4	8.26 × 11.69	A5	5.85 × 8.26	EXECUTIVE	7.5 × 10	LEDGER	17 × 11	LEGAL	8.5 × 14	LETTER	8.5 × 11	STATEMENT	5.5 × 8.5	TABLOID	11 × 17
Name	Size (inches)																				
A3	11.69 × 16.53																				
A4	8.26 × 11.69																				
A5	5.85 × 8.26																				
EXECUTIVE	7.5 × 10																				
LEDGER	17 × 11																				
LEGAL	8.5 × 14																				
LETTER	8.5 × 11																				
STATEMENT	5.5 × 8.5																				
TABLOID	11 × 17																				
PAPERSIZE	<p>The page size to use if none of the standard paper-size designations will work. For example, to define an 8.5" × 11.75" piece of paper, specify the following:</p> <p>PAPERSIZE WIDTH=8.5 HEIGHT=11.75</p> <p>For landscape orientation, specify the larger value as the WIDTH. The default is 14" × 8.5".</p> <p>Note: If you specify a PAPERNAME, do not specify a PAPERSIZE.</p> <p>If you set PAPERSIZE and BOTTOMBORDER, do not set MAXLINESPERPAGE to a number.</p>																				

Parameter	Description										
ROTATEVALUE= [angle 0]	<p>The text rotation angle, which must be one of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Effect</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No rotation (the default)</td> </tr> <tr> <td>90</td> <td>90-degree clockwise rotation—the top of the page is to the right</td> </tr> <tr> <td>180</td> <td>180-degree clockwise rotation—the page is upside-down</td> </tr> <tr> <td>270</td> <td>270-degree clockwise rotation—the top of the page is to the left</td> </tr> </tbody> </table> <p>These are the only values you can specify.</p>	Value	Effect	0	No rotation (the default)	90	90-degree clockwise rotation—the top of the page is to the right	180	180-degree clockwise rotation—the page is upside-down	270	270-degree clockwise rotation—the top of the page is to the left
Value	Effect										
0	No rotation (the default)										
90	90-degree clockwise rotation—the top of the page is to the right										
180	180-degree clockwise rotation—the page is upside-down										
270	270-degree clockwise rotation—the top of the page is to the left										
STRIPCC=[ON OFF]	<p>Whether to remove CC data. ON tells the PDF converter to strip all CC data from column 1 of the report and not use the data at all. This is useful if you want to eliminate form feeding other than what you indicate using MAXLINESPERPAGE. The default is OFF.</p> <p>Note: “STRIPCC=ON” forces the CC value to “OFF.”</p> <p>See also “CC={ON OFF}” (page 8-8).</p>										
TOPBORDER	<p>The size (in inches) of the border at the top of the page. The default is 1.</p> <p>You can use decimal values (such as .5, 1.5, 1.55) with up to two digits to the right of the decimal point. For example, 1.66 is a valid value, but 1.651 is not.</p>										
VERTICALSPACE	<p>The amount of leading (in points) between text lines. For example, if you specify a FONTSIZE smaller than 10 points, you may want to reduce the VERTICALSPACE value to 6. The default is 12 (1/6 inch).</p>										
<p>The following parameters set field values in the Properties dialog of the PDF file. These fields can be changed when the file is opened in a PDF editor.</p>											
AUTHOR	Name of the person or machine, for example, that created the PDF. Enclose the string in quotes.										
SUBJECT	Subject of the PDF. Enclose the string in quotes.										
KEYWORDS	A string of one or more keywords related the subject of the PDF. Enclose the string in quotes.										

Calculating the Page Layout

This section uses an example to describe how to fit a report to a specific page size.

First, *leading* (the vertical space between the bottom of one line of text and the top of the next line of text) and *font sizes* are measured in points. There are 72 points to an inch.

Example

Suppose you have an 8½" by 11" piece of paper that you want to print in landscape mode, and you want a ½" border around the page. This gives you a printing area 7½" high by 10" wide. If you convert these values to points, you have an area that is 540 points high and 720 points wide.

Calculating Height

To fit 50 lines on a page, divide 540 points by 50 lines. This results in 10.8 points. If you used a 10-point font size, the lines would be nearly on top of one another. To provide a little more space between lines, therefore, you could use an 8-point font size with a vertical space of 10 points.

Calculating Width

Dividing the 720-point width by 8 points (the font size) results in 90 characters across. This is the maximum number of characters across. If your print lines are wider, they will be truncated at the 90th character. If you have 120 characters to print on each line, you can either reduce the font size and vertical space further, or use a larger paper size such as a legal-size sheet.

You can define any page size you want. The only reason for adjusting the proper fit for a specific page size is that the recipient is actually going to print the data you send.

Conversion Examples

Example 1

This example automatically sends an e-mail to *bob@here.com* with an attachment that is automatically converted to a PDF file before it is sent.

First, define an automated script as you would do for any DEFINE EVENT process. This script must contain all the necessary commands to pass to the EMAIL client. The DEFINE NAME command defines the script name to TCP/IP FOR VSE.

```
DEFINE NAME,NAME=EMSCRIPT,SCRIPT=EMSCRIPT
```

Next, define the EMAIL automated EVENT client. Scan the POWER queue for all CLASS=K reports that have a script name in the "DEST=" field of the POWER LST statement that generated the report.

```
DEFINE EVENT, ID=MAIL1, ACTION=EMAIL, CLASS=K, HOST=DEST, RETRY=1
```

Anything going into class *K* is sent with the SCRIPT file specified in the DEST field of the POWER LST card, as follows.

```
* $$ LST DISP=D, CLASS=K, DEST=(, PAYSCR)
```

Finally, the script *PAYSCR.L* (cataloged to a VSE library) contains the following commands:

```
SET PDF=PDFCHECK
SET TO=bob@here.com
SET NEWNAME=&PWRNAME..PDF
TEXT
Here are the weekly payroll checks for your review!
/+
```

All checks are delivered directly to *bob@here.com*.

Remember that you need to have a PDF configuration file set up correctly with the appropriate font sizes and paper layout so that the print area matches the output forms.

Example 2

This example uses FTPBATCH to send data from POWER to a VSE library. The default PDF configuration file is used.

```
// EXEC FTPBATCH

LOPEN
LUSER lname
LPASS lpassword
OPEN remote_VSE_host_ipaddress
USER rname
PASS rpassword
LCD PRD2.HTML
LSITE RECFM S
LSITE PDF ON
CD POWER.LST.X
QUOTE SITE CC ON
GET REPORT1.01234 BOBINFO.PDF
QUIT
```

Example 3

This example uses FTPBATCH to transfer a POWER file to a remote host and convert it to a PDF file at the same time.

```
*   FTP a file and convert it to a PDF file
*
// JOB BATFTPJ
// EXEC FTPBATCH,SIZE=FTPBATCH
LOPEN
LUSER tstuser
LPASS tstpass
OPEN aaa.bbb.ccc.ddd
USER tstuser2
PASS tstpass2
LSITE PDF LSBPDFL
LCD \
LCD POWER.LST.Z
LSITE RECFM F
PWD
CD TSTDIR/JOE
PUT LSVCAT LSBFTPT.PDF
LCLOSE
CLOSE
QUIT
/*
/&
* $$ E0J
```

In this example, the LSITE command enabling PDF conversion specifies a custom configuration file named *LSBPDFL*. This file was cataloged by running the following job.

```
// JOB LIBRCAT
// EXEC LIBR
ACC SUB=BIMLIB.CSICNFG
CATALOG LSBPDFL.L                      REPLACE=YES
FONTNAME=COURIER
FONTSIZE=9
STRIPCC=YES
VERTICALSPACE=9
FONTTYPE=NORMAL
PAPERNAME=LETTER, LANDSCAPE
TOPBORDER=.4
LEFTBORDER=.4
MAXLINESPERPAGE=64
/+
/*
/&
* $$ EOJ
```

