Command Reference

TCP/IP for VSE

Version 2 Release 2

TCP/IP is a communications facility that permits bi-directional communication between VSE-based software and software running on other platforms equipped with TCP/IP.

This manual contains detailed explanations of the commands available for configuring TCP/IP FOR VSE.

Published October 2017

CSI INTERNATIONAL

"Delivering what the competition can only promise"

www.csi-international.com • info@csi-international.com • 800.795.4914

TCP/IP FOR VSE Command Reference
Version 2 Release 2
October 2017

Published by CSI International

Phone: 800-795-4914

Fax: 740-986-6022

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

Product questions: info@csi-international.com

Technical support: support@csi-international.com

Review comments: documentation@csi-international.com

CSI International 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
Email:	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.

Telephone:	Toll Free in the USA	800-795-4914
	Worldwide	740-420-5400

CSI International provides support to address each issue according to its severity.

Updates to This Manual

The following table describes updates to this manual. Updates may be identified by a fix number in CSI International's support database.

October 2017

ID	Command	Change Description	Page
	DEFINE FTPD	Added TLS 1.2 protocol support. Added the SSLMODE= operand.	68
	DEFINE TLSD	Added TLS 1.2 protocol support.	127
	FIREWALL	Updated the REPORT operand description. Added the ALLOWED and the BLOCKED operands.	176

Table of Contents

CSI International Technical Support i
Updates to This Manual ii
1. Introduction 1
2. Syntax Summary 6
3. Command Descriptions 17
Summary List 18
ACCESS 26
ASECURITY 27
AUTOLOAD 31
CHECKSUM 32
CLOSE FILE 33
CONNECT_SEQUENCE 34
CONSOLE_HOLD 37
DEFINE ADAPTER 38
DEFINE ALTIP 41
DEFINE CGI 43
DEFINE EVENT 44
DEFINE FILE 54
DEFINE FILEIO 66
DEFINE FTPD 68
DEFINE GPSD 78
DEFINE HTTPD 87
DEFINE LINK 90
DEFINE LOG 97
DEFINE LPD 100
DEFINE MASK 103
DEFINE MENU 105
DEFINE NAME 106
DEFINE NTPD 107
DEFINE PUBLISHER 109
DEFINE ROUTE 111

DEFINE SOTRACE 118
DEFINE TELNETD....121
DEFINE TLSD....127
DEFINE TRACE 131
DEFINE TRANSLATION....133
DEFINE USER....135
DELETE ALTIP....139
DELETE CGI....140
DELETE EVENT 141
DELETE FILE 142
DELETE FILEIO....143
DELETE FTPD 144
DELETE GPSD....145
DELETE HTTPD 146
DELETE LINK 147
DELETE LPD 148
DELETE MASK....149
DELETE MENU 150
DELETE NAME 151
DELETE NTPD....152
DELETE PUBLISHER....153
DELETE ROUTE 154
DELETE TELNETD....155
DELETE TLSD....156
DELETE TRACE 157
DELETE USER....158
DIAGNOSE....159
DISCOVER 164
DOWNCHECK....166
DUMP....167
DUMPOPTION....170
EMAIL....171
EXECUTE....174
EXTPGVS....175
FIREWALL 176
FLUSH....178
FTPBATCH_FETCH....179
GATEWAY 180
IBBLOK....182
INCLUDE 184

IPSTAT 185
ISOLATION 186
LISTIDCAMS 187
LOCAL_DLBL 188
MESSAGE (or MSG) 189
MODIFY CONSOLE 191
MODIFY FILE 193
MODIFY LOG 204
MODIFY ROUTE 206
PING 211
PING_MESSAGE 213
PORTQUEUE 214
PORTRANGE 216
QUERY ACTIVE 217
QUERY ALL 219
QUERY ALTIPS 222
QUERY ARPS 223
QUERY CGIS 225
QUERY CONNECTIONS 226
QUERY DIAGNOSE 230
QUERY DUMPOPTIONS 231
QUERY EMAIL 232
QUERY EVENTS 233
QUERY EXTERNAL 235
QUERY EXTTYPES 236
QUERY FILEIO 238
QUERY FILES 239
QUERY FIREWALL 240
QUERY FRAGMENTS 241
QUERY FTPDS 242
QUERY GPSDS 244
QUERY HOME 245
QUERY HTTPDS 246
QUERY IBBLOKS 247
QUERY IPSTAT 248
QUERY ISTATISTICS 250
QUERY LINKS 251
QUERY LOCKS 252
QUERY LOGS 253
QUERY LPDS 254

QUERY MASKS....255
QUERY MENUS....256
QUERY NAMES....257
QUERY NTPD....259
QUERY OPENFILES....260
QUERY OPTIONS (or QUERY SET)....261
QUERY PORTQUEUE....262
QUERY PRODKEYS....263
QUERY PROGRAMS....264
QUERY PUBLISHERS....265
QUERY ROUTES....266
QUERY SECURITY....268
QUERY STATISTICS (or QUERY STATS)....269
QUERY STOR....273
QUERY TASKS....276
QUERY TELNETDS....278
QUERY TLSDS....279
QUERY TRACES....280
QUERY TRANSLATIONS....281
QUERY TRUSTED....282
QUERY USERS....283
QUERY VERSIONS....284
QUIESCE....285
RAPTRAC....287
RECORD....289
REDEFINE....290
RELOAD....292
RESUME....294
SDOPEN_EXTRA....295
SECURITY....296
SEGMENT....300
SEPARATOR_PAGES....301
SET AUTO_TIME....303
SET DEFAULT_DOMAIN....304
SET DIAGNOSE....305
SET DNS*n*....306
SET DNST*n*....308
SET FIXED_RETRANSMIT....309
SET IPADDR....311
SET LINK_RETRY....313

SET MASK 314
SET MAXIMUM_MESSAGES 316
SET MAX_EMAIL_EVENTS 317
SET MAX_FTP_EVENTS 318
SET MAX_LPR_EVENTS 319
SET MAX_SEGMENT 320
SET PAGE_COUNT 322
SET PASSWORD 323
SET POWERPASSWORD 324
SET POWERUSERID 325
SET PULSE_TIME 326
SET RETRANSMIT 328
SET TELNET_TRANSLATE 330
SET TELNETD_BUFFERS 331
SET TELNETD_BUFSIZE 332
SET WINDOW 333
SHUTDOWN 335
SINGLEDEST 337
SPINCHECK 339
START 340
STEALTH 341
STOP 342
SUSPEND 344
TRACERT 345
TRAFFIC 347
TRUST 348
UPCASE 349
VERIFY_MEMORY 350
WAITFOR 351

4. Deprecated Commands 352

1

Introduction

Before using the *TCP/IP FOR VSE Command Reference*, it is important to understand the conventions and syntax used. Each command's description includes the following information:

- Usage summary
- Syntax diagram
- Argument list
- Usage example showing output messages
- Related commands

Note: Message fields are explained in the *TCP/IP FOR VSE Messages*.

Command Syntax

The command syntax diagrams in this manual appear as follows:

```
COMMand  REQuired=value,{KEY1|KEY2|KEY3}
         [,OPTParm={Yes|No|number}] [,VALue=87]
         [,BIGval=4K] [,TIME=30s]
```

This example shows how uppercase and lowercase letters are used in command names. The uppercase letters are required. The remaining letters are optional and may be included for clarity. Whatever string you use on the command line is syntax checked and must be accurate.

This convention also applies to arguments and keyword values.

Arguments

The following argument conventions are used in the example above:

REQuired=

Required parameters are shown without square brackets. Uppercase and lowercase letters indicate the minimum required characters. Generic placeholders for values are italicized.

KEY

Where you can specify a keyword, it appears without an equal sign. The braces "{ }" indicate that you must choose one of the values in the list. Because the list is not enclosed in brackets, the parameter is required. If the parameter were optional, it would appear as

[KEY1|KEY2|KEY3]

where the underscored value (KEY2) is the default.

OPTParm=

An optional parameter appears within brackets. In the example, the available values are limited to the choices shown. Keywords are shown non-italicized, with the capital letters indicating the minimum abbreviation. Values in italics indicate a generic placeholder, in this case a numeric value. The default value is underscored.

VALue=

When an optional value can be specified, the default value appears as non-italicized text.

BIGVal=

Frequently, large numeric values need to be specified. These can be entered as normal numeric or with a "multiplier suffix." Here are some examples of how suffixes can be used:

4K = 4,096

4k = 4,000

1M = 1,048,576

4M = 4,194,304

4m = 4,000,000

TIME=

A time interval must be specified with many commands. Often, this interval is specified in 300th-second units. This unit of measure is common to VSE programming.

Because it is cumbersome to code hours, minutes, or seconds in 300th-second units, you can use a suffix of "s," "m," or "h" to cause the appropriate conversion.

Note that capital letters are the same as lower-case letters for time values.

`2s` = 600/300th

`2m` = 36,000/300th

`1h` = 1,080,000/300th

Specifying Values

Values can be specified as follows:

- Multiple values generally are separated with commas and enclosed in parentheses:

```
UNIT=040
UNIT=(040,041)
```

- If a range of values is supplied, the values are separated with a hyphen and enclosed in parentheses:

```
UNIT=(040-044)
```

- "Names" used by the TCP/IP FOR VSE stack generally are up to 16 characters long. The first character is alphabetic, and the remainder can be alphabetic or numeric.

Argument Placeholders

Each argument in a command has specific format requirements. Some formats are used repeatedly throughout the product. To eliminate redundant explanations of formats, standard types are represented by reserved names in the syntax diagrams. These types follow.

name64

This string is the same as *name16*, except that the maximum length is 64 characters.

name16

This represents an internal TCP/IP name. These names are not case sensitive and are stored in upper-case. Each name consists of 1 to 16 alphabetic characters (A–Z), digits (0–9), or special characters (_ . $ @ # & ¢). The first character must be alphabetic (A–Z).

name8

A 1- to 8-character name, case insensitive. It consists of alphabetic characters (A–Z), digits (0–9), or special characters (_ $ @ #). The first character must be alphabetic (A–Z).

id

This is a special use of a *name16* field. When called for, the value must be unique within the TCP/IP stack for a particular group. For example, you can have only one DEFINE LINK with a specific ID, but a DEFINE HTTPD could have the same name because it is in a different group. It is used to identify daemons, table entries, and other items that must be identified individually.

pubname

Public names are not case sensitive. Each pubname consists of from 1 to 6 segments of from 1 to 8 characters each. The segments are separated by periods (.), forward slashes (/), or backward slashes (\). Each segment must begin with a letter (A–Z). The remainder of the characters can be letters, digits, or special characters (_ - $ @ #).

member

This represents a member in a VSE library. It is not case sensitive and is 1 to 8 characters long, not counting the 1- to 8-character extension. Files to be transferred are located by indicating their file system location. Files that hold TCP/IP definitions and commands are located using the partition's search chain, not the TCP/IP file system.

ext

The *ext* parameter indicates a VSE library member's name extension. It is case insensitive and can be from 1 to 16 characters in length.

port

A TCP/IP port number. The UDP and TCP protocols use port numbers to establish communications with specific remote clients and servers. The numbers range from 1 to 65635. When "0" is used, the stack automatically substitutes an appropriate, unused number.

When specifying a port number as a parameter, do not use commas. Depending on the application, commas may have a special meaning. For example, in the FTP PORT command, the port number is specified as two decimal numbers (0–255) separated by a comma. To obtain the actual port number, these two fields are converted to one hexadecimal byte each, concatenated into a two-byte field, and then converted back to a single decimal number.

ip4addr

IPv4 addresses consist of four decimal numbers separated by periods. Each numeric field ranges from 0 to 255. You can include leading zeros in each field if you want.

ipaddr

An IPv4 address.

host

A host name conforming to DNS standards or an IP address.

num
A numeric value.

sec
Time in seconds.

sec300
Time in 300th-second units.

msec
Time in milliseconds.

min
Time in minutes.

2

Syntax Summary

The following table summarizes each command's syntax. You can use it for quick reference.

Command	Arguments
ACCess	{ Query | CLEAR | ALlow,IPaddress = *ip4addr* | PRevent,IPaddress = *ip4addr* }
ASECUrity	[,ICMP={YES|NO}] [,FTPD={YES|NO}] [,WEBL={YES|NO}] [,FTPC={YES|NO}] [,ARP={YES|NO}] [,IPAV={YES|NO}] [,BLOCKIP={YES|NO}] [,BLOCKCNt=*num*] [,SCAN={YES|NO}]
AUTOLoad	{ON|OFF]}
CHECKSum	{SOftware|HArdware|OFf}
CLOSe FILE	PUBlic=*pubname*
CONNECT_Sequence	{ON|OFF}
CONSOLE_Hold	{ON|OFF}
DEFine ADAPter	LINKid=*id* ,IPaddr=*ip4addr* ,TYPe={ETHERnet|TOKEN_ring|FDDI} [,NUMber=0] [,MTU=*num*]
DEFine ALTip	ID=*id* ,IPaddr=*ip4addr*
DEFine CGI	PUBlic=*member* ,TYPE={CGI|CGI-BAL|CGI-REXX}

Command	Arguments
DEFine Event	ID=*id* [,ACTion={Lpr|Ftp|EMAIL}] [,CLass=X] [,Queue={Lst|Pun|Rdr}] [,REtry=1] [,RETRY_Time=45s] [,HOSTname={USERinfo|DEST|ROOM|DEPT|BLDG|NONE}] [,SINGle={Yes|No}] [,USErid=$EVENT] [,PASSword=$EVENT] [,PRIOrity={Yes|No}] [,ORDEr={Yes|JOBNUMber|No}] [,POWERSYSid=*sysid*] [,JECLscript={YES|NO}] [,SCRIPTtype=L|*type*] [,SCRIPTName=*name*] [,NULLfile={Skip|Ignore|Process|Fail|Delete}] [,FCBPrefix=*string4*]
DEFine FILE	TYPe={ESDS|KSDS|SAM|LIBrary|ICCF|POWer| VSAMCAT|VTOC|HFS|DSPACE|BIM-EDIT|CONDOR| FALCON|TAPE|VOLLIE} [,DLBL=*name8*] ,PUBlic=*pubname* [,DRIVer=*member*] [,ALLOWsite={Yes|No} [,READonly={Yes|No}] [,CC={Yes|No}] [,TRcc={YES|NO}] [,CRlf={Yes|No}] [,RECFm=F|FB|V|VB|S|SV|SU|SB] [,LRECL=*num*] [,BLKsize=*num*] [,GID=*snum*] [,UID=*snum*][,TRANslate=*name16*] [,SITE={Yes|No}][,DBLOCKS=*num*] [,EXT=*name8*] [VOLid=*volser*] [,CIPHER={NULL-SHA1|SDESCBC-NULL| SDESCBC-SHA1|TDESCBC-NULL| TDESCBC-SHA1|AES128C-NULL| AES128C-SHA1|AES192C-NULL|AES192C-SHA1|AES256C-NULL| AES256C-SHA1|KEYMASTER}] [,CIPHERKEY=CIALHFSK]
DEFine FILEIO	TYPe={ESDS|KSDS|SAM|LIBrary|ICCF|POWer| VSAMCAT|VTOC| HFS|DSPACE|BIM-EDIT|TAPE} [,DRIVer=*phase-name*]
DEFine FTPd	ID=*id* [,PORT=21] [,MAXACTive=3] [,UNIX={Yes|No|Binary}] [,TRANslate=*name16*] [,TIMEOut=2m] [,BSize=64K] [,WELCOME=*member*] [,EXTtypes={Yes|No}] [,EXTRADATA={FAIL|WARN|IGNORE|ACCEPT}] [,DYNfiles={Yes|No}] [,ALLowabort={Yes|No}] [,HESitate=0] [,IDLEtimeout=0] [,SITELAST={Yes|No}] [,SSL={YES|NO|YESCLAuth}] [,SSLKEY=*member*] [,SSLVERsion={SSL30|TLS10|TLS11|TLS12}] [,SSLCIPHER={ALL|WEAK|STRONG|AES|DES|NULL|HARDware| MEDIUM}] [,SSLDATAconn={CLEAR|PRIVATE}] [,ZEROerr={Yes|No}] [,IPaddr=*ip4addr*] [,JOURnal={Yes|No}] [,UPPERcase={Yes|No}] [,SENDFast={Yes|No}] [,REXX={Yes|No}] [,SENDWack={Yes|No}] [,SSLMODE={IMPLICIT|EXPLICIT}]

Command	Arguments
DEFine GPSd	ID=*id* ,IPaddr=*host* ,TERMname=*lu* [,ALTlength=132] [,BRACKet_eject={Yes|No}] [,CMD*n*=*string*] [,CONTrol_order=NFU] [,DEBug={Yes|No}] [,EMULate={3287|Transparent}] [,INSerts=*member*] [,INSESsion={No|Yes}] [,LINELength=132] [,LOG={No|Yes}] [,LOGMode=*mode*] [,MAXChars=1m] [,MAXIdle=10s] [,MAXLines=10k] [,MAXPages=1k] [,NRT=60s] [,NRC=3] [,NOEJect={No|Yes}] [,OUTput={Lpr|Direct}[,PORT=515]] [,PRinter=*name*] [,Queuing={Disk|Memory}] [,STORage=*pubname*] [,TARGet=*appl*] [,TRANslate=*name16*] [,TYPE=VTAM] [,VRC=10] [,VRT=60s]
DEFine HTTpd	ID=*id* ,ROOT=*pubname* [,PORT=80][,CONfine={Yes|No}] [,TRANslate=*name16*] [,TIMEOut=5m] [,SECure={Yes|No}] [,LIBrary=*pubname* ,SUBlibrary=*name8*]
DEFine LINK	ID=*id* ,IPaddr=*ip4addr* ,TYpe={CLAW|LCS|CTCa|OSAX|IPNET} [,DEVices=*hexaddr*] [,SYSid=*num*] [,MTU=*num*] [,FORCe] [,OUTBuffers=4] [,HOSTName=*name8*] [,HOSTAppl=TCPIP] [,WSName=*name8*] [,WSAppl=*name8*] [,INFactor=4] [,OUTFactor=4] [,RETRY_Time=*time*] [,STOPPED] [,STOPLan={Yes|No}] [,SHUTdown={Yes|No}] [,VMReset={Yes|No}] [,ALTIP=(*ip4addr* [,*ip4addr*,...,*ip4addr*])] [,ROUTER=None|Primary|Secondary}] [,OSAPort={0|1] [,DATapath=*cuu*] [,PORTName=*name8*]
DEFine LOG	ID=*id* ,TYPE={PRinter} ,LOGICALUnit=*lunit* [,LINELength=*num*] [,TIMEstamp={Left|Right|None}] [,*routes*]
DEFine LPD	PRinter=*name16*, Queue=*pubname* [,LIBrary=*pubname1*] [,SUBlibrary=*name*] [,TRANslate=*name*] [,HEXdump={YES|NO}] [,USERid=$LPD] [,PASSword=$LPD]
DEFine MASK	NETwork=*ip4addr*, MASk=*ip4addr*
DEFine MENu	ID=*id*, MEMber=*member*
DEFine NAME	NAMe=*name16* {[,IPaddr=*ip4addr*] | [,SCRipt=*member*]}
DEFine NTPd	ID=*id* [,PORt=37] [,PRotocol={Udp|Tcp}] [,GMT=*snum*] [,ADJustment=*snum*]
DEFine PUBlisher	ID=*id* ,IMODlist=*member*

Command	Arguments
DEFine ROUTe	ID=*id* ,LINKid=*name16* [,ADAPter=0] ,IPaddr=*ip4addr* [,GATEway=*ip4addr1*] [,AFTer=*id*] [,MTU=*num*] [,MSS=*num*] [,CRETran=*msec*] [,DRETran=*msec*] [,FIXRetran={Yes|No}] [,MINRetran=*msec*] [,MAXRetran=*msec*] [,PULse=*sec*] [,WINdow=*num*] [,RPAuse=*msec*] [,RETRY=*num*]
DEFine SOTRACe	ID=*id* [,IPaddr=*ip4addr*] [,PORT=*port*] [,SCOPE={All|Internal|External|Obsolete}] [,MAXData=60] [,PHASE=*member*] [,SIZe=500] [,KIND={TCP|UDP|FTP|CLIENT|TELNET}]
DEFine TELnetd	ID=*id* ,TARget=*name8* ,TERMname=*luname* [,COunt=*num*] [,BASE=1] [,TN3270E={Listener|Effector}] [,PORT=23] [,MENu=*name16*] [,LOGMode=S3270] [,LOGMODE3=D4B32783] [,LOGMODE4=D4B32784] [,LOGMODE5=D4B32785] [,IPaddr=0.0.0.0] [,BUFfersize=*num*] [,CLOSE={Always|Seldom}][,GENeric=*name16*] [,GRoup=*name16*] [,POOL={Yes|No}] [,DRIVer=TELNETD]
DEFine TLSd	ID=*id* ,CERTLibrary=*name8* ,CERTMember=*name8* ,CERTSublibrary=*name8* [,CIPher=09] [,MINVers={0300|0301|0302|0303}] [,PORT=443] [,PASSport=80] [,TYPE={1|2}]
DEFine TRACe	ID=*id* [,IPaddr=*ip4addr*] [,PORT=*port*] [,SIZe=500] [,KIND={TCP|UDP|ICMP|ALL}]
DEFine TRANslation	MEMber=*member* [,ENTry=*name*] [,NAMe=*name16*] [,DEFault=*name16*]
	TYpe=Double ,MEMber=*member* ,AScii=*name* ,EBcdic=*name* ,NAMe=*name16* [,DEFault=*name16*]
	DEFault=*name16*
DEFine USEr	ID=*name16* [,PASSword=*name16*] [,DATA=*any*] [,GID=*snum*] [,UID=*snum*] [,MAILbox=*str*] [,FTP={YES|NO}] [,LPR={YES|NO}] [,WEB={YES|NO}] [,TELNET={YES|NO}] [,ROOT=*path*]
DELete ALTIP	ID=*id*
DELete CGI	PUBLIC=*pubname*
DELete EVENT	{ID=*id*|ALL[,FORCE]}
DELete FILE	PUBlic=*pubname*
DELete FILEIO	ID=*name*

Command	Arguments
DELete FTPd	ID=*id*
DELETE GPSD	ID=*id*
DELete HTTPd	ID=*id*
DELete LINK	ID=*id*
DELete LPD	ID=*id*
DELete MASK	NETwork=*ip4addr*
DELete MENu	ID=*id*
DELete NAME	NAME={*name*|DYNAMIC}
DELete NTPd	ID=*id*
DELete PUBLisher	ID=*id*
DELete ROUTe	ID=*id*
DELete TELnetd	ID=*id*
DELete TLSd	ID=*id*
DELete TRACe DELete TRACES	ID=*id*
DELete USEr	NAMe=*id*
DIAGnose	{OFF|[-]*keyword*}
DISCover	*host*
DOWNcheck	{ON|OFF}
DUMP	{ALL|BUFfers|BUS|CONnects|DIRectory|Events|EXITS| FRAgments|FREquests|FTPds|GIVesockets|GPSds|HTTpds|IVBlok| LINks|LPDs|MASks|MENus|NAMes|PARTITION|PROGrams|ROUtes| SOCkets|STAts|TASks|TELnetds|TIMERS|TLSds|TRACes| TRANslations|USErs|VERsions}
DUMPOption	FILeio={Csi|Ibm|None}

Command	Arguments
EMAIL	[,SMTPd=*ip4addr*] [,ATsign=7C] [,FRom=*string64*] [,DESTination=*string64*] [,REPlyto=*string64*] [,RPORT=25] [,SUBject=*string64*] [,USERid=$EMAIL] [,PASSword=$EMAIL] [,LUSERid=$EMAIL] [,LPASSword=$EMAIL] [,TRANslation=*name16*] [,TRAttachments=*name16*] [,TRMail=*name16*] [,CHECKname={Yes|No}] [,GMT=*snum*]
EXECute	*member*
EXTPGVS	{ON|OFF}
FIREWALL	{ON|OFF|LOAD [PHASE=*phase-name*]|WARN|FAIL|MSGON|MSGOFF| DEBUGON|DEBUGOFF|REPORT|ALLOW|BLOCK}
FLUSH	*ip4addr* [,*port*]
FTPBatch_fetch	{ON|OFF}
GATEway	{ON|OFF}
IBBLok	[RELEase|RESET|CLear] [,{SIZE=*num*|,MTU=*num*}] [,FREE=*num*] [,STORage=*num*|PERcent=*percent*]
INCLude	*member* [,DELAY]
IPSTAT	{ON|OFF}
ISOLAtion	{ON|OFF}
LISTIDCAMS	{ON|OFF}
LOCAL_DLBL	{ON|OFF}
MESsage (or MSG)	MSGID=*msgid* [Console={Yes|No}] [Log={Yes|No}] [SCROLlable={Yes|No}]
MODify CONsole	[,LINELength=*num*] [,TIMEstamp={Left|Right|None}] [,*routes*]

Command	Arguments
MODify FILE	PUBlic=*pubname* [,TYPe={ESDS|KSDS|SAM|LIBrary|ICCF|POWer|VSAMCAT|VTOC| HFS|DSPACE|BIM-EDIT|CONDOR|FALCON|VOLLIE|TAPE}] [,DLBL=*name8*] [,DRIVer=*member*] [,ALLOWsite={Yes|No}] [,READonly={Yes|No}] [,CC={Yes|No}] [,TRcc={YES|NO}] [,CRlf={Yes|No}] [,RECFm=F|FB|V|VB|S|SV|SU|SB] [,LRECL=*num*] [,BLKsize=*num*] [,GID=*snum*] [,UID=*snum*] [,TRANslate=*name16*] [,SITE={Yes|No}] [,DBLOCKS=*num*] [,EXT=*name8*] [VOLid=*volser*] [,CIPHER={NULL-SHA1| SDESCBC-NULL|SDESCBC-SHA1|TDESCBC-NULL|TDESCBC-SHA1| AES128C-NULL|AES128C-SHA1| AES192C-NULL|AES192C-SHA1| AES256C-NULL|AES256C-SHA1| KEYMASTER}] [,CIPHERKEY=CIALHFSK]
MODify LOG	ID=*id* [,LINELength=*num*] [,TIMEstamp={Left|Right|None}] [,*routes*]
MODify ROUTe	ID=*id* [,LINKid=*name16*] [,NUMber=0] [,IPaddr=*ip4addr*] [,GATEway=*ip4addr1*] [,AFTer=*id*] [,MTU=*num*] [,MSS=*num*] [,CRETran=*msec*] [,DRETran=*msec*] [,FIXRetran={Yes|No}] [,MINRetran=*msec*] [,MAXRetran=*msec*] [,PULse=*sec*] [,WINdow=*num*] [,RPAuse=*msec*] [,RETRY=*num*]
PING	*host*
PING_Message	{ON|OFF}
PORTQueue	PORT=*port* [,TIMEOUT=*sec*] [,DEPTH=*num*]
PORTRange	LOw=*port* ,HIgh=*port*
Query ACTive	[,SYSLST] TYPe={TELnetd|FTPd|LINK|ALL}
Query ALL	[,SYSLST]
Query ALTips	[,SYSLST]
Query ARPs	[,SYSLST] [,IPaddr=*ip4addr*]
Query CGIs	[,SYSLST] [,PUBlic=*pubname*]
Query CONnections	[,SYSLST] [,IPaddr=*ip4addr*] [,FPort=*port*] [,LPort=*port*] [,STate={Listen|Established}] [,EXTended] [,Trace]
Query DIAGnose	[,SYSLST]
Query DUMPoptions	[,SYSLST]
Query EMAil	[,SYSLST]

Command	Arguments
Query Events	[,SYSLST] [,ID=*id*] [,DETail]
Query EXTERNAL	[,SYSLST]
Query EXTtypes	[,SYSLST]
Query FILEIo	[,SYSLST]
Query Files	[,SYSLST] [,PUBlic=*pubname*]
Query FIREWALL	[,SYSLST]
Query FRAGments	[,SYSLST]
Query FTPds	[,SYSLST] [,ID=*id*] [,EXTended]
Query GPSD	[,SYSLST] [,ID=*id*]
Query HOME	[,SYSLST]
Query HTtpds	[,SYSLST] [,ID=*id*]
Query IBbloks	[,SYSLST]
Query IPSTAT	[,SYSLST]
Query ISTATistics	[,SYSLST]
Query LINks	[,SYSLST] [,ID=*id*]
Query LOCKs	[,SYSLST]
Query LOGs	[,SYSLST]
Query LPds	[,SYSLST] [,ID=*id*]
Query MASKs	[,SYSLST]
Query MENus	[,SYSLST]
Query NAMes	[,SYSLST] [,TYPe={ALL|STATIC|DYNAMIC}]
Query NTPd	[,SYSLST] [,ID=*id*]
Query OPENfiles	[,SYSLST]
Query OPTions	[,SYSLST]

Command	Arguments
Query PORTqueue	[,SYSLST]
Query PRODkeys	[,SYSLST] [,ALL]
Query PROGrams	[,SYSLST]
Query PUBLishers	[,SYSLST] [,ID=*id*]
Query ROUTes	[,SYSLST] [,ID=*id*|IPaddr=*ip4addr*]
Query SECURity	[,SYSLST]
Query SET	[,SYSLST]
Query STATistics	[,SYSLST]
Query STOR	[,SYSLST] [SAVe|TRend|MAXimum]
Query TASKs	[,SYSLST] [,ID=*hexnum*|NAMe=*name*] [,EXTENDED]
Query TELnetds	[,SYSLST] [,ID=*name*]
Query TLSDs	[,SYSLST] [,ID=*id*]
Query TRACes	[,SYSLST]
Query TRANslations	[,SYSLST]
Query TRUSted	[,SYSLST]
Query USERs	[,SYSLST] [,NAMe=*name*]
Query VERsions	[,SYSLST]
QUIESce	{ON|Off}
RAPTRAC	START|REPORT|STOP
RECord	{ON|OFF}
REDefine	START
RELOad	{EXTtypes|HACKlist|TERMtypes}
RESume	*tasknum*

Command	Arguments
SECURity	[ON|OFF] [,BATCH={ON|OFF}] [,PHASE=*member*] [,XDATA=*string*] [,ADATA=*string*] [,ASMDATE=*string*] [,ASMTIME=*string*] [,VERSION=*string*] [,AUTO={ON|OFF}] [,EXIT={ON|OFF}] [,ARP={ON|OFF}] [,MODE={WARN|FAIL}] [,LOGGING={ALL|FAIL|NONE}] [,DUMP=ALL|FAIL|NONE}] [,LOCK]
SEGment	[NEW]
SEParator_pages	{OFF|ON|[-]Email|[-]FTp|[-]HTtp|[-]LPr}
SET	AUTO_TIME=*sec300*
SET	CONSOLE_PORT=*port*
SET	DEFAULT_DOMAIN=*string*
SET	DIAGNOSE {NOCONSOLE|CONSOLE}
SET	DNS*n*=*ip4addr*
SET	DNST*n*=*sec300*
SET	FIXED_RETRansmit {ON|OFF}
SET	IPaddr=*ip4addr*
SET	LINK_retry=*time*
SET	MASK=*ip4addr*
SET	MAX_Email_events=*num*
SET	MAX_Ftp_events=*num*
SET	MAX_Lpr_events=*num*
SET	MAX_Segment=*num*
SET	MAXIMUM_MESSAGES=*num*
SET	PAGE_COunt=*num*
SET	PASSword=*string*
SET	POWERPassword=*string8*
SET	POWERUserid=*string8*

Command	Arguments
SET	PULse_time=*sec300*
SET	RETransmit=*time*
SET	TELNET_Translate=*name16*
SET	TELNETD_BUFFers=*num*
SET	TELNETD_BUFSize=*num*
SET	WINdow=*num*
SHUTdown	[Immediate]
SINGledest	{ON|OFF}
SPINcheck	{ON|OFF}
START	[LINKid=*id*]
STEALTH	{ON|OFF}
STOP	[LINKid=*id*]
SUSPend	*Taskid*
TRACert	*host*
TRAFFic	{ON|OFF|FULL}
TRUSt	{ADD|DELete} ,IPaddress=*ip4addr*
Upcase	{ON|OFF}
VERIFY_MEMORY	{ON|OFF}
WAITfor	{Vtam|TIME=*nn*}

3

Command Descriptions

The pages of this chapter contain the full reference information for TCP/IP FOR VSE commands.

A commands summary table precedes the first command. This table contains a brief description of each command.

Summary List

The TCP/IP FOR VSE commands are summarized in the following table.

Command	Description
ACCESS	Controls access to VSE by IP address
ASECURITY	Configures the Automatic Security Exit
AUTOLOAD	Determines automatic loading of file I/O drivers when files are defined
CHECKSUM	Controls how checksums are computed
CLOSE FILE	Closes an open file
CONNECT_SEQUENCE	Controls whether connection requests are allocated by IP address pattern checking
CONSOLE_HOLD	Maintains a console command prompt
DEFINE ADAPTER	Creates an adapter definition within the scope of a DEFINE LINK
DEFINE ALTIP	Causes the stack to monitor and respond to ARP requests for additional home addresses
DEFINE CGI	Loads a CGI program and makes it available for use
DEFINE EVENT	Monitors a POWER class for automatic report distribution
DEFINE FILE	Defines a file in the TCP/IP file system and associates it with a file I/O driver
DEFINE FILEIO	Loads a file I/O driver phase for a file type into storage
DEFINE FTPD	Creates a file transfer protocol daemon
DEFINE GPSD	Creates a general print server daemon
DEFINE HTTPD	Creates a hypertext transfer protocol (Web server) daemon
DEFINE LINK	Creates a link between TCP/IP and a network or to a directly connected stack
DEFINE LOG	Creates a system log file
DEFINE LPD	Creates a line printer daemon
DEFINE MASK	Creates a subnet mask for a particular network

Command	Description
DEFINE MENU	Loads a menu file and makes it available for use by telnet daemons
DEFINE NAME	Associates a TCP/IP name with an IP address or a script file
DEFINE NTPD	Creates a network time server daemon
DEFINE PUBLISHER	Creates a publishing daemon
DEFINE ROUTE	Add an entry to the TCP/IP routing table
DEFINE SOTRACE	Starts a socket trace
DEFINE TELNETD	Creates a TN3270 or TN3270E daemon
DEFINE TLSD	Creates an SSL/TLS daemon
DEFINE TRACE	Starts a datagram trace
DEFINE TRANSLATION	Loads and controls ASCII/EBCDIC translation tables
DEFINE USER	Creates a user ID and password
DELETE ALTIP	Removes an alternate home address
DELETE CGI	Removes a CGI program from storage
DELETE EVENT	Terminates monitoring of a POWER class
DELETE FILE	Removes a file from the TCP/IP file system
DELETE FILEIO	Removes a file I/O driver phase from storage
DELETE FTPD	Terminates a file transfer protocol daemon
DELETE GPSD	Terminates a general print server daemon
DELETE HTTPD	Terminates a hypertext transfer protocol (Web server) daemon
DELETE LINK	Removes a link between TCP/IP and a network or to a directly connected stack
DELETE LPD	Terminates a line printer daemon
DELETE MASK	Deletes a subnet mask for a particular network
DELETE MENU	Removes a TN3270 menu file from memory
DELETE NAME	Removes a TCP/IP symbolic name

Command	Description
DELETE NTPD	Terminates a network time server daemon
DELETE PUBLISHER	Terminates a publisher daemon
DELETE ROUTE	Removes an entry from the network routing table
DELETE TELNETD	Terminates a TN3270 or TN3270E daemon
DELETE TLSD	Terminates an SSL/TLS Daemon
DELETE TRACE	Terminates a trace and free its storage
DELETE USER	Removes a user ID and password entry
DIAGNOSE	Controls diagnostic display options
DISCOVER	Determines the “best” MTU size to a remote host
DOWNCHECK	Controls the safety prompt for SHUTDOWN command
DUMP	Performs a formatted dump of various TCP/IP control blocks
DUMPOPTION	Controls the format of TCP/IP-initiated storage dumps
EMAIL	Sets global default options for the EMAIL client
EXECUTE	Executes an operator command script
EXTPGVS	Controls whether external-partition socket requests are allocated in 31-bit private partition storage or in 31-bit system GETVIS storage.
FIREWALL	Controls and monitors the Firewall Shield optional feature
FLUSH	Terminates all processing with a specific remote host
FTPBATCH_FETCH	Determines automatic use of the FTPBATCH program when the FTP batch program is invoked
GATEWAY	Controls forwarding of datagrams not intended for the VSE stack
IBBLOK	Controls allocation and retention of Internet Buffer Blocks
INCLUDE	Includes a library member in the initialization parameter set
IPSTAT	Controls whether an ISBLOK is allocated for every IP address that accesses the stack
ISOLATION	Prevents inbound connection requests from being honored

Command	Description
LISTIDCAMS	For VSAMCAT: controls whether the results of DELETE/DEFINE CLUSTER operations from a calling IDCAMS are output to SYSLST
LOCAL_DLBL	Controls whether internal servers (DEFINE FTPD) allow remote FTP users to specify VSE files using a local DLBL
MESSAGE	Controls message suppression
MODIFY CONSOLE	Controls how messages are displayed on the console; it is a shortcut for MODIFY LOG,ID=CONSOLE
MODIFY FILE	Changes parameters and characteristics of an entry in the TCP/IP file system
MODIFY LOG	Changes the characteristics of a system log file
MODIFY ROUTE	Change values on an existing entry in the route table
OPENCHECK	Enables a periodic check for files left open by various processes
PING	Issues an ICMP echo (PING) request
PING_MESSAGE	Controls the “ping request received” console message
PORTQUEUE	Controls how inbound connection requests are queued for an application
PORTRANGE	Controls range for dynamic port assignment
QUERY ACTIVE	Displays the status of active daemons
QUERY ALL	Issues all QUERY commands
QUERY ALTIPS	Displays all alternate IP addresses
QUERY ARPS	Displays the current content of the ARP table
QUERY CGIS	Displays all currently available CGI programs
QUERY CONNECTIONS	Displays the status of one or more connections
QUERY DIAGNOSE	Displays current Diagnose settings
QUERY DUMPOPTIONS	Displays options for TCP/IP-produced dumps
QUERY EMAIL	Displays EMAIL client settings
QUERY EVENTS	Displays the status of automation processing

Command	Description
QUERY EXTERNAL	Displays information about partitions external to TCP/IP
QUERY EXTTYPES	Displays the contents of the External Types table
QUERY FILES	Displays the contents of the TCP/IP file system
QUERY FILEIO	Displays the status of the file I/O driver programs
QUERY FIREWALL	Displays the current firewall settings being enforced.
QUERY FRAGMENTS	Displays the status of fragmented datagram reassembly
QUERY FTPDS	Displays the status of the file transfer protocol daemons
QUERY GPSDS	Displays the status of the general print server daemons
QUERY HOME	Displays all IP addresses in the “Home Address” table
QUERY HTTPDS	Displays the status of the hypertext transfer protocol (Web server) daemons
QUERY IBBLOKS	Displays IBBLOK settings and statistics
QUERY IPSTAT	Displays IP address statistics stored in ISBLOKs.
QUERY ISTATS	Displays statistics detailing internal stack functions
QUERY LINKS	Displays the status of network links
QUERY LOCKS	Displays the current lock status
QUERY LOGS	Displays available consoles and logs, along with their properties
QUERY LPDS	Displays the status of line printer daemons
QUERY MASKS	Shows all defined subnetwork masks by network number
QUERY MENUS	Displays menus available for TN3270 use
QUERY NAMES	Displays TCP/IP names and their associated values
QUERY NTPDS	Displays status of NTP daemons
QUERY OPENFILES	Displays a list a files that are currently open
QUERY OPTIONS	Displays the current values of modifiable parameters
QUERY PORTQUEUE	Displays statistics of queued connection requests
QUERY PUBLISHER	Displays the status of the publisher daemon

Command	Description
QUERY PRODKEYS	Displays the TCP/IP product keys being used, the names of the licensed products, and the expiration dates of the keys
QUERY PROGRAMS	Displays the program phases being used by TCP/IP, their characteristics, their memory locations, and the library from which each was loaded
QUERY ROUTES	Displays the content of the network routing table or the route taken to reach a specific address
QUERY SECURITY	Displays current security settings
QUERY SET	Is a synonym for QUERY OPTIONS
QUERY STATISTICS	Displays a summary of stack-related information
QUERY STOR	Displays detailed information on memory use
QUERY SUSPENDED	Displays a list of suspended tasks
QUERY TASKS	Displays a list of pseudo tasks
QUERY TELNETDS	Displays TN3270 and TN3270E daemons
QUERY TLSD	Displays currently active TLS daemons
QUERY TRACES	Displays a list of currently running traces
QUERY TRANSLATES	Displays a list of available translate tables
QUERY TRUSTED	Displays the currently “trusted” IP addresses
QUERY USERS	Displays a list of defined user IDs
QUERY VERSIONS	Displays the versions and maintenance levels of stack components
QUIESCE	Prevents new connections while permitting existing connections to continue
RAPTRAC	Outputs process-tracking data for CSI Technical Support
RECORD	Controls recording of pseudo task statistics on SYSLST
REDEFINE	Permits redefinition of a daemon based on previously entered parameters
RELOAD	Reloads an internal table
RESUME	Resumes processing of a suspended pseudo task

Command	Description
SDOPEN_EXTRA	Controls extra tests to ensure that a SAM file was successfully opened
SECURITY	Controls TCP/IP security functions
SEGMENT	Segments the SYSLST and log files, making them available for printing
SEPARATOR_PAGES	Controls the generation of POWER separator pages
SET AUTO_TIME	Determines when automation is to check POWER queues
SET DEFAULT_DOMAIN	Establishes a domain name to be appended automatically to unqualified names
SET DIAGNOSE	Enables displaying DIAGNOSE messages on the console
SET DNS*n*	Defines a domain names server (DNS) for name resolution
SET DNST*n*	Controls the time-out value for the indicated DNS
SET FIXED_RETRANSMIT	Controls the default setting for the FIXRETRAN= parameter on DEFINE ROUTE
SET IPADDR	Establishes the default home address for the stack
SET LINK_RETRY	Determines the default time interval between attempts to reinitialize a failed network link
SET MASK	Establishes a default subnet mask
SET MAX_EMAIL_EVENTS	Establishes the maximum number of simultaneous EMAIL events for automation processing
SET MAX_FTP_EVENTS	Establishes the maximum number of simultaneous FTP events for automation processing
SET MAX_LPR_EVENTS	Establishes the maximum number of simultaneous LPR events for automation processing
SET MAX_SEGMENT	Controls the default setting for the inbound Maximum Segment Size (MSS= parameter on DEFINE ROUTE)
SET MAXIMUM_MESSAGES	Sets the maximum number of queued (not printed) system messages to SYSLOG
SET PASSWORD	Establishes a password for entering console commands
SET POWERPASSWORD	Establishes the password for POWER access

Command	Description
SET POWERUSERID	Establishes the user ID for POWER access
SET PULSE_TIME	Sets the default value for the interval between probes of inactive connections (PULSE= on DEFINE ROUTE)
SET RETRANSMIT	Sets the default value for the CRETRANS= and the DRETRANS= parameters on DEFINE ROUTE
SET TELNET_TRANSLATE	Sets the name of the translate table to be used with Telnet (not TN3270) connections
SET TELNETD_BUFFERS	Sets the size of the buffer pool shared by TN3270 daemons
SET TELNETD_BUFSIZE	Determines the size of individual TN3270 buffers
SET WINDOW	Sets the default size for the TCP inbound window (WINDOW= parameter on DEFINE ROUTE)
SHUTDOWN	Terminates processing and shuts down the stack
SINGLEDEST	Determines how automation processing handles multiple reports queued for the same destination (host).
SPINCHECK	Controls how TCP/IP monitors for runaway pseudo tasks
START	Starts TCP/IP dispatching engine or a “stopped” network link
STEALTH	Controls if a rejected connection request is reset or ignored
STOP	Stops the TCP/IP dispatching engine
SUSPEND	Halts processing by a pseudo task
TRACERT	Displays each “hop” in a route along with the time required to reach it
TRAFFIC	Controls which network traffic is allowed entry to the stack
TRUST	Establishes that an IP address is trusted and suspicious activity is to be ignored
UPCASE	Determines whether console and log messages are displayed in mixed case or forced to uppercase characters
VERIFY_MEMORY	Controls the storage monitor
WAITFOR	Causes initialization processing to wait for a specific event

ACCESS

The ACCESS command can be used to restrict or allow access to VSE based on the remote host's IP address. It also displays blocked addresses.

Syntax

ACCESS {Query|CLEAR|ALlow,IPaddress = *ip4addr*|
PRevent,IPaddress = *ip4addr*}

Arguments

Query
Displays addresses currently in "prevent" mode.

CLEAR
Clears all addresses currently in "prevent" mode.

ALlow
Removes the specified address from "prevent" mode.

PRevent
Adds the specified address to the table of addresses from which traffic will not be accepted.

Example

```
access prevent,ip=192.168.1.66
IPI107I All traffic with 192.168.1.66 will be prevented

access clear
IPI110I All traffic with 192.168.1.66 will be allowed
IPI108I Access CLEAR complete
```

Notes

The following notes apply to this command:

- Some internal TCP/IP processes automatically add an address to the "prevent" table if a hacking attempt is detected.
- The security exit can specify that an address be added to the "prevent" table.
- Before an address can be automatically added to the "prevent" table, the address must make a connection attempt.

Related Commands

ASECURITY
Configures the Automatic Security Exit

SECURITY
Controls TCP/IP security functions.

STEALTH
Controls whether a rejected connection request is reset or ignored.

ASECURITY

The Automatic SECURITY command provides many of the features of a custom-written security exit without the need to program an exit.

Syntax

```
ASECUrity [,ICMP={YES|NO}] [,FTPD={YES|NO}]
          [,FTPC={YES|NO}] [,ARP={YES|NO}]
          [,IPAV={YES|NO}] [,BLOCKIP={YES|NO}]
          [,BLOCKCNt=count] [,SCAN={YES|NO}]
          [,WEBL={YES|NO}]
```

Arguments

ICMP=

This parameter controls how the stack responds to PING requests.

YES

Allows normal responses to ICMP ECHO (ping) requests.

NO

Prevents VSE from responding to incoming ICMP PING requests. This is useful to stop "ping sweeps," which are commonly used to find active machines on a TCP/IP network. This setting does not affect ping requests that originate on VSE.

FTPD=

This parameter controls how attempts to start an FTP session are handled.

YES

Allows normal connection to FTP daemons.

NO

Prevents new FTP sessions. Already-established sessions continue unaffected. Controls connection requests to the FTP daemon and blocks the 220 system welcome message. This can be used to temporarily stop new FTP sessions.

FTPC=

As with FTPD=, this parameter permits establishing an FTP connection, but it causes commands to be rejected with a "500 Command rejected" message.

YES

Commands are processed normally.

NO

The following commands are refused: USER, PASS, ACCT, QUIT, REIN, SYST, HELP, NOOP, PBSZ, PROT, and AUTH.

ARP=

This parameter controls how the stack responds to ARP requests. It could be useful for stopping all current inbound activity due to an ARP attack

YES

Allows normal ARP response.

NO

Requires SECURITY ARP=ON to already be in effect. TCP/IP does not respond to ARP requests.

IPAV=

This parameter controls all inbound IP traffic (IP datagrams).

YES

Allows normal IP processing.

NO

Requires SECURITY IP=ON to already be in effect. Specifying "NO" immediately prevents processing of all incoming IP datagrams. This is a drastic step, but one that might be useful if an Internet attack is in progress.

BLOCKIP=

Enables automatic blocking of an IP address after it reaches an allowed number of security violations. This number is set by BLOCKCNt=. The ACCESS command can be used to reset the block for an IP address.

YES

Automatically block a specific IP address after BLOCKCNt security violations. This is the default.

NO

Do not automatically block access by IP address.

BLOCKCNt=

This is the value used by BLOCKIP to determine the maximum number of security violations before blocking that IP address. The count can range from 1 to 255. The default is zero (never).

SCAN=

This is the HTTPD Scanblock Request.

YES

If a remote user attempts to open a connection to the HTTP daemon, even partially (a half open), and then ends the connection without sending any data, TCP/IP FOR VSE tracks this event and blocks the user's IP address after a predetermined number of such calls. The maximum number of such calls is always 3 unless BLOCKIP=YES (the default) and BLOCKCNt is set to either 1 or 2. See also BLOCKIP=. An HTTP908W

message is always issued when a half-open connection is attempted. The ACCESS command can be used to reset the block for an IP address. YES is the default.

NO
IP addresses are not blocked after the predetermined number of half-open calls.

WEBL=
This is the Web Logon Screen Request.

YES
The HTTP daemon maintains a minimal level of access security based on the network address. To do this, the daemon maintains a table of "active" IP addresses, which are ones that have recently accessed a specific web page, eliminating the need to log in at every link. When a request is received from an address not in the table, the daemon automatically displays a page that requests a user ID and password. These values are checked through the standard TCP/IP FOR VSE mechanisms.

If valid, the IP address is added to the table and the original request is transmitted. The IP address is removed from the table when explicitly requested (a request made for "BLANKING.HTML") or when the HTTPD inactivity timer (TIMEOUT=) expires.

NO
No automatic security checking is performed.

Example

See the *TCP/IP FOR VSE Messages* manual for information on each message in this example.

```
asecurity icmp=yes,blockip=yes,blockcnt=10
IPN759I Security status change: Auto security changed ICMP=Y
IPN759I Security status change: Auto security changed BLOCKIP=Y
IPN473I Auto Security blocking by IP address Enabled
IPN474I Auto Security blocking by IP address after 10 violations
```

Notes

The following notes apply to this command:

- The use of the Automatic Security Exit is controlled by the SECURITY command. Once you have selected options with ASECURITY, you must still enable the exit with SECURITY.
- When blocking IP addresses, remember that users may be behind a router that causes them to "share" a single IP address.
- If you use the Automatic Security feature, be sure that any user IDs (DEFINE USER) have the correct values in the DATA= field.

Related Commands

ACCESS
Controls access to VSE by IP address.

DEFINE USER
Creates a user ID and password.

DELETE USER
Removes a user ID and password entry.

ISOLATION
Prevents inbound connection requests from being honored.

PING_MESSAGE
Controls the "ping request received" console message.

QUERY ARPS
Displays the current content of the ARP table.

QUERY SECURITY
Displays current security settings.

QUERY USERS
Displays a list of defined user IDs.

SECURITY
Controls TCP/IP security functions.

AUTOLOAD

The AUTOLOAD command controls file driver loading. When a file is defined, TCP/IP normally checks to see that the appropriate file I/O driver is loaded and initialized. By disabling automatic loading, file I/O drivers can only be loaded under DEFINE FILEIO.

Syntax

AUTOLoad [ON|OFF]

Arguments

ON
File I/O drivers are loaded and initialized when a file of the appropriate type is defined. This is the default.

OFF
File I/O drivers are not automatically loaded and initialized.

Example

```
autoload on
IPN268I AutoLoad now set to On
```

Related Commands

DEFINE FILE
Defines a file to the TCP/IP FOR VSE file system and associates it with a file I/O driver.

DEFINE FILEIO
Defines a file I/O driver for a file type.

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY FILEIO
Displays the status of the File I/O driver programs.

CHECKSUM

The CHECKSUM command controls whether TCP/IP FOR VSE computes and verifies checksums on incoming traffic.

Syntax

CHECKSum [SOftware|HArdware|OFf]

Arguments

SOftware
The checksums of incoming datagrams are computed and validated using an internal software algorithm. This is the default.

HArdware
The checksums of incoming and outgoing datagrams are computed and validated using the hardware CHECKSUM instruction. The CHECKSUM instruction is available on most CPUs.

OFf
The checksums of incoming datagrams are ignored. The checksums of outgoing datagrams continue to be computed using the last-specified technique (hardware or software).

Example

```
checksum off
IPN386I Inbound Checksum set OFF
IPN300I Enter TCP/IP Command

checksum on
IPN385I Inbound Checksum set ON
```

Exposition

Checksums are a vital part of the TCP/IP protocol. No production system should ever operate without them. You should turn off checksum computation only when you are debugging an installation or when you are trying to determine which network device is transmitting flawed data.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

CLOSE FILE

The CLOSE FILE command enables you to close a file that was opened by the TCP/IP FOR VSE file system. This typically is the result of an abnormal condition where processing did not complete, causing a VSAM file, for example, to be "owned" by the TCP/IP partition.

This command should be used only in extraordinary circumstances.

Syntax

CLOSe FILE PUBlic=*pubname*

Arguments

PUBlic=
The name of the file to close.

Example

```
close file,pub=ewf.save.t999.print
IPN579I A close request has been queued for: EWF.SAVE.T999.PRINT
```

Notes

The following notes apply to this command:

- Be very careful when you issue the CLOSE FILE command. No checking is performed to determine whether the file being closed is actually in use. If you close an active file, the results are unpredictable and undesirable. In general, the only time you should use CLOSE FILE is when an FTP process has failed and all other cleanup methods have not succeeded.
- If multiple processes are using a file, CLOSE FILE acts on the first instance it encounters. You need to issue the command multiple times to close all instances of the file.

Related Commands

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY OPENFILES
Displays a list of files that are currently open.

CONNECT_SEQUENCE

The CONNECT_SEQUENCE command controls how TCP/IP FOR VSE services session requests based on IP address.

This command affects TelnetD, FTPD, and potentially any TCP server application that issues a listen socket request on VSE. See Notes below.

Syntax

CONNECT_Sequence {ON|OFF}

Arguments

ON

When a client requests a session, a corresponding daemon is assigned on a best-fit basis, depending on the IP address.

OFF

When a client requests a session, a daemon is assigned on a first-fit basis irrespective of the IP address. This is the default.

Exposition

Whenever a remote client such as Telnet or FTP requests a session, the session manager must assign the request to a daemon. This is done by matching the IP address of the requestor with the IP address coded on the daemon definition.

IP addresses can match based on a complete specification, a match on subnet, or a match on network number. A daemon with a 0.0.0.0 specification will match any request.

When CONNECT_SEQUENCE is set to OFF, the first daemon that matches any test is assigned. The IPADDR= is completely ignored, and any client can connect without any IP address matching.

When CONNECT_SEQUENCE is set to ON, several passes can be made through the available daemon list. The match criteria, in order, are as follows:

- An exact match of the IP address by any available daemon.
- A match of an IP address with the network number and the subnet number of the incoming connection.
- A match of an IP address with the network number of the incoming connection.
- Any daemon with an IP address of 0.0.0.0, which is the default. Any inbound connection can match this value.

The SET MASK command has no effect on the selection process.

The following Telnet example assumes that the following partial definitions exist with CONNECT_SEQUENCE ON:

```
DEFINE TELNETD,ID=AAAA,COUNT=3,  IP=10.108.34.10
DEFINE TELNETD,ID=BBBB,COUNT=40, IP=10.32.0.0
DEFINE TELNETD,ID=CCCC,COUNT=100,IP=10.0.0.0
DEFINE TELNETD,ID=DDDD,COUNT=200,IP=0.0.0.0
```

Incoming client requests would be serviced in the following sequence:

- Clients with IPADDR 10.108.034.10 would connect to ID=AAAA.
- Clients with IPADDR 10.32.*nnn.nnn* would connect to ID=BBBB.
- Clients with IPADDR 10.*nnn.nnn.nnn* would connect to ID=CCCC, other than the 10.108.034.101 and 10.32.*nnn.nnn* addresses that are serviced by ID=AAAA and ID=BBBB.

TCP/IP FOR VSE also considers the total count per daemon:

- If the number of clients exceeds the ID=AAAA count (3), then the client with IPADDR 10.108.034.010 would connect into ID=BBBB.
- If the number of clients exceeds the ID=BBBB count (40), then clients with IPADDR=10.32.*.* would connect into ID=CCCC.
- If the number of clients exceeds the ID=CCCC count (100), then clients would connect into ID=DDDD.

If there were no ID=DDDD defined that allowed any clients to connect to it—it has the default setting of IPADDR=0.0.0.0—the client connection requests that did not match the ID=AAAA, ID=BBBB, or ID=CCCC IPADDR= settings would be rejected.

Example

```
connect_seq on
IPN569I Full connect sequence set to on
```

Notes

The following notes apply to this command:

- Another method for assigning specific daemons to incoming client session requests is to use multiple ports. Depending on your requirements, this method may be easier to administer and more efficient to operate.
- Any TCP server application that issues a listen socket request on VSE can control the remote IP addresses allowed to connect into it by specifying the FOIP= on the passive socket open request. But this is limited to TCP applications using the assembler macro interface, and CONNECT_SEQUENCE ON must be active.

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon

CONSOLE_HOLD

The CONSOLE_HOLD command controls the display of the command prompt on the VSE operator's console.

Syntax

CONSOLE_Hold {ON|OFF}

Arguments

ON

TCP/IP FOR VSE displays the console prompt immediately at startup and maintains the prompt regardless of operator action.

OFf

TCP/IP FOR VSE initializes without a console prompt. Messaging the partition is required before the operator can enter commands. The prompt is redisplayed until a null entry is made. This is the default.

Example

```
console_hold off
IPN482I TCP/IP for VSE reply ID will not be maintained

console_hold on
IPN482I TCP/IP for VSE reply ID will be maintained
```

Notes

The following notes apply to this command:

- To enter commands to TCP/IP FOR VSE, you must message the partition and wait for the command prompt to appear. Once you enter a command, the command prompt is redisplayed until a null entry is made. Setting CONSOLE_HOLD ON maintains the prompt on the screen regardless of null entries.
- If you include CONSOLE_HOLD ON in your initialization deck, the command prompt is automatically displayed at TCP/IP FOR VSE initialization.
- Previous versions of TCP/IP FOR VSE required considerable 24-bit storage to load the command parser. This requirement has been eliminated, and there is no memory savings achieved by deleting the command prompt.
- You can also enter commands by using the DATA= parameter of the VSE MSG command.

DEFINE ADAPTER

The DEFINE ADAPTER command identifies a specific adapter within a 3172, OSA, or 3172-compatible communications controller. This command is valid only when it follows a previously issued DEFINE LINK command that defines a 3172 or an OSA.

Syntax

DEFine ADAPter LINKid=*id* [,IPaddr=*ip4addr*]
,TYPe={ETHERnet|TOKEN_ring|FDDI} [,NUMber=0]
[,MTU=*num*]

Arguments

LINKid=
This field must specify the ID from the DEFINE LINK command that defines the 3172 or OSA containing this adapter.

IPaddr=
Specifies the IP address to be used with this adapter. This is essential for multi-homing because TCP/IP FOR VSE's IP address must be consistent with the network to which it is connected. If IPaddr= is omitted, the IP address specified in the SET IPADDR command is assigned to the adapter.

TYPE=
Specifies the type of network to which this adapter is physically connected.

ETHERnet
This is an Ethernet adapter.

TOKEN_ring
This is a Token_Ring adapter.

FDDI
This is an FDDI adapter.

NUMber=
Specifies the adapter's position within the 3172. Adapter numbers range from 0 through 99. Most devices use values of 0 through 3. The IBM 2216 N-WAYS control unit supports up to 16 adapters.

MTU=
Specifies the Maximum Transmission Unit size to be used with this adapter. An MTU size of 576 is always valid. This is the minimum value, and its acceptance is required of all TCP/IP implementations as part of the standard. This value is also the least efficient one. The recommended values are shown in the MTU Sizes table below. Keep in mind that the value you choose must be supported by each and every device on the network.

MTU Sizes

The following table shows the maximum and default MTU sizes for the different adapter types.

Adapter Type	Maximum MTU Size	Default MTU Size
Ethernet	1,500	1,500
Token Ring: 4-megabit network	4,000	1,500
Token Ring: 8-megabit network	8,000	1,500
FDDI	2,000	1,500

Example

See the *TCP/IP FOR VSE Messages* manual for information on each message in this example.

```
define link,id=link3172,type=osa2,dev=(032,33)

define adapter,linkid=link3172,number=0,type=ethernet, -
ip=192.168.1.161,mtu=1500

IPT100I Internet Link Level (ILL) Processor LCS starting
IPL491I OSA link LINK3172 started on devices 0032 – 0033
IPL491I OSA link LINK3172 started adapter 0 as 192.168.1.161
```

Notes

The following notes apply to this command:

- This command must refer to a 3172 or OSA control unit that has already been defined with a DEFINE LINK command.
- You must issue this command from the initialization library member. It cannot be issued from the console except under special circumstances; see the example above.
- Once a 3172 or OSA has been defined, all related DEFINE ADAPTER commands must be issued before TCP/IP is permitted to process (other than initialization).
- To issue a set of DEFINE LINK and related DEFINE ADAPTER commands from the console, first issue the STOP command. This prevents TCP/IP from prematurely completing the DEFINE LINK processing before all adapter information is specified. After your definitions are complete, issue the START command to resume processing.

- If you have problems communicating with one or more hosts on your network, try reducing the MTU size.
- Regardless of MTU settings, TCP/IP FOR VSE accepts incoming datagrams of any size.

Related Commands

DEFINE LINK
Creates a link between TCP/IP and a network or a directly connected stack.

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

EXECUTE
Executes an operator command script.

QUERY LINKS
Displays the status of network links.

SET IPADDR
Establishes the default home address for the stack.

START
Starts the TCP/IP dispatching engine or a “stopped” network link.

STOP
Stops the TCP/IP dispatching engine.

DEFINE ALTIP

The DEFINE ALTIP command identifies additional IP (network) addresses to which TCP/IP FOR VSE will respond when it receives an ARP request.

Syntax

DEFine ALTip ID=*id* ,IPaddr=*ip4addr*

Arguments

ID=
Is an ID to identify the table entry.

IPaddr=
Is an address for which TCP/IP FOR VSE will respond to ARP requests.

Example

```
define altip,id=sys2,ipaddr=64.10.5.2
IPN380I Alternative IP address, ID: SYS2 IPAddr: 64.10.5.2
```

Exposition

The TCP/IP routing rules require that hosts with the same network or subnetwork number reside on the same physical network (Ethernet, Token Ring, and so on). Hosts with different network or subnetwork numbers may not reside on the same physical network. There are techniques that permit two logical networks to share a single physical network (and vice versa). This is not easy, however, and it is not supported. This means that if you want to have one or more TCP/IP FOR VSE partitions connect to your network using a single adapter-owning partition, these auxiliary partitions need IP addresses on a separate network or subnetwork.

To reduce the proliferation of networks, DEFINE ALTIP causes the adapter-owning partition to serve as a proxy for the auxiliary. This means that each auxiliary partition appears on the physical network as if it was actually cabled to it.

The DEFINE ALTIP command makes it possible for routers on the same physical network to know about the alternate IP addresses.

Notes

The following notes apply to this command:

- Datagrams matching an “altip” address are automatically routed, regardless of the GATEWAY setting.
- Alternate IPs are useful only in conjunction with DEFINE LINKs that specify either TYPE=IPNET or TYPE=CTCA.

Related Commands

DEFINE LINK
Creates a link between TCP/IP and a network or to a directly connected stack.

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

DELETE ALTIP
Removes an alternate home address.

GATEWAY
Controls forwarding of datagrams not intended for the VSE stack.

QUERY ALTIPS
Displays all alternate IP addresses.

DEFINE CGI

The DEFINE CGI command adds a CGI phase or REXX program to TCP/IP FOR VSE. Once added, the CGI can be invoked with a web browser request.

Syntax

DEFine CGI PUBlic=*member*,TYPE={CGI|CGI-BAL|CGI-REXX}

Arguments

PUBlic=
The name of the program to be invoked.

TYPE=
Is one of the following values:

CGI
An "old-style" CGI routine

CGI-BAL
An assembler routine using the current interface definition.

CGI-REXX
A CGI routine coded in REXX.

Example

```
define cgi,public='addrbk',type=cgi-rexx
IPN694I Entry defined
```

Notes

The following notes apply to this command:

- See the *TCP/IP for VSE Programmer's Guide* for more information about writing CGI programs.
- The response in the above example refers to a file being defined. This is normal.

Related Commands

DELETE CGI
Removes a CGI program from storage.

QUERY CGIS
Displays all currently available CGI programs.

DEFINE EVENT

The DEFINE EVENT command permits you to establish actions that are triggered when specified events occur on your VSE system.

Syntax

```
DEFine EVent ID=id [,ACTion={Lpr|Ftp|EMAIL}] [,CLass=X]
        [,FCBPrefix=string4]
        [,HOSTname={USERinfo|DEST|ROOM|DEPT|BLDG|NONE}]
        [,JECLscript={YES|NO}]
        [,NULLfile={Skip|Ignore|Process|Fail|Delete}]
        [,ORDEr={Yes|JOBNUMber|No}] [,PASSword=$EVENT]
        [,PRIOrity={Yes|No}] [,POWERSYSid=sysid]
        [,Queue={Lst|Pun|Rdr}] [,REtry=1]
        [,RETRY_Time=45s] [,SCRIPTName=name]
        [,SCRIPTType=L|type] [,SINGle={Yes|No}]
        [,USErid=$EVENT]
```

Arguments

ID=

Is a unique name to identify the event.

ACTion=

Determines the action that will be taken for this event, as follows:

Lpr

Event processing follows the LPR protocol. This means that an LP daemon must be provided on the remote host. This is the default.

Ftp

Event processing follows the FTP protocol. This means that an FTP daemon must be provided on the remote host.

Email

Event processing follows the SMTP protocol. This means that an SMTP daemon must be provided on the remote host to process outgoing email.

CLass=

The VSE/POWER class in which the event will occur. The default class is “X”.

FCBPrefix=

If supplied, this 1- to 4-character value is used in a “SET FCBPREFIX=” command passed to the EMAIL and LPR clients. FTP events are unaffected. By default, no “SET FCBPREFIX=” command is issued.

HOSTname=

Specifies a JECL field to be inspected for the remote host's IP address or the name of a script file. Allowable values are USERinfo, DEST, ROOM, DEPT, BLDG, and NONE. NONE specifies to ignore these JECL fields and use the value of either the SCRIPTName= parameter or the SCRIPT= JECL field, if it is set. The default for HOSTNAME= is USERinfo.

The flow chart below shows how the auto client determines the script name/IP address to use. The precedence of the JECL field specified by HOSTNAME= depends on whether the action is set to FTP or LPR/EMAIL.

If a field is missing from the JECL and SCRIPTName= is omitted from the DEFINE EVENT statement, the report is treated as an error and is not processed. The report then goes into DISP=Y. If a field is specified and it contains a value, then SCRIPTName= is ignored.

Note: ICCF will override some JECL settings and insert a user name or a SAS name in either the DEST= or the USER= JECL field. You can do a PDISPLAY LST,ALL,FULL after the report is created to see how a particular site is configured.

JECLscript=

Sets whether to override the default script name set by SCRIPTName= with a different name set in the JECL.

YES

Use the value set by "SCRIPT=*value*" in the LST JECL, if this field is set. If it is omitted, use the value set by SCRIPTName=. Example 2, page 51, shows how to use JECLscript=YES.

NO

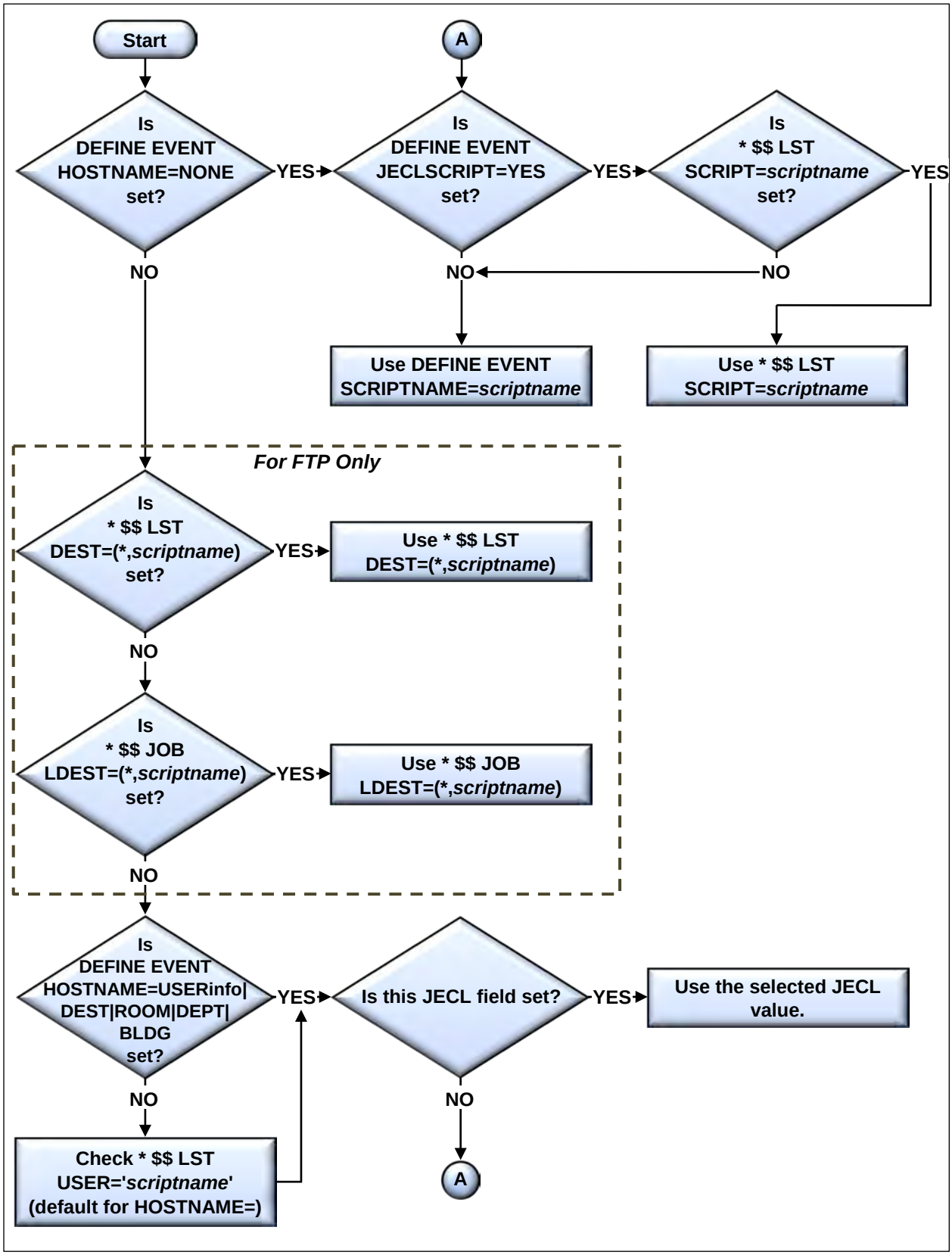
Do not override the name set by SCRIPTName=. This is the default.

Note: In addition to setting the JECLscript=YES parameter, you will need to modify your VSE/POWER startup to include the second DEFINE statement below:

```
DEFINE L,CICSDATA,3F00,1,255,*
DEFINE L,SCRIPT,FFDC,1,8,C
```

Insert that line after the "CICSDATA" line that normally comes with the default startup. To check whether the proper entry is defined to VSE/POWER as part of the VSE/POWER initialization, you can issue a PDISPLAY AUSTMT command. The response shows all of the user-defined fields. If "SCRIPT" is among them, then adding SCRIPT= to your LST JECL will work. If not, then you will need to modify your VSE/POWER startup.

How Auto Client Determines the Script Name or IP Address



NULLfile=

This parameter specifies how empty VSE/POWER files (no records) are to be handled.

Skip

The file will NOT be processed.

Ignore

This value is synonymous with SKIP.

Process

The file will be processed normally. Depending on the type or transfer, destination, and other factors, this may result in other errors occurring. This value is the default.

Fail

The report will be sent to DISP=Y.

Delete

The report will be deleted from the VSE/POWER queue.

ORDEr=

This parameter applies to FTP and LPR operations. It controls the order in which VSE/POWER queue entries are processed. The ORDER setting affects only the event you are defining, which is for a specific queue and class. The values are as follows.

Yes

The Auto-EVENT client processes entries in the order they are returned by a "PDISPLAY *queue*,FREE,CCLASS=*class*" command. Single-threaded delivery is enabled (SINGLe=YES is forced) to ensure that this order is maintained. YES is the default. For more information on the PDISPLAY command, see the IBM manual *VSE/POWER Administration & Operation*.

JOBNUMber

The Auto-EVENT client sorts the entries returned by a "PDISPLAY *queue*,FREE,CCLASS=*class*" command before processing them. Entries are sorted by VSE/POWER job number and then by VSE/POWER segment number. Single-threaded delivery is enabled (SINGLe=YES is forced) to ensure that this order is maintained.

No

Entries are processed in no particular order. Deliveries will be multi-threaded depending on how related parameters are set (SINGLe=YES|NO as well as MAX_FTP_EVENTS and MAX_LPR_EVENTS values). See also the SINGLEDEST command for related information.

Because VSE/POWER queue entries can be segmented, JOBNUMBER ordering may differ from the default ordering in

some cases. You can compare the results to determine which setting is most appropriate for your environment.

After an event is defined, QUERY EVENTS will show the ORDER= setting as follows for ORDER=NO, ORDER=JOBNUMBER, and ORDER=YES, respectively:

```
T1 0101 IPN427I    Priority: Yes, Order: None, Script: L
T1 0101 IPN427I    Priority: Yes, Order: Jobnum, Script: L
T1 0101 IPN427I    Priority: Yes, Order: Yes, Script: L
```

PASSword=

A 1- to 16-character uppercase password that will be used to logon to the local FTP or LPR clients. The default is "$EVENT."

POWERSYSid=

If you operate in a shared-spool environment, then you must specify the POWER SYSID of your "local" system. Because Automation processing cannot access files that belong to systems other than the local one, this parameter limits what the event "sees." If you specify a value in a non-shared environment, or if you specify an incorrect value, then no automation processing will occur for this event.

PRIOrity

This field indicates whether the value of the JECL PRI= field is to be used in determining the order of event processing.

Yes

The PRI value will be considered when determine the order in which events are selected for processing. This is the default.

No

The PRI value will be ignored.

Queue=

Keyword. This is the POWER queue to be monitored.

Lst

The POWER LST queue will be monitored. This is the default.

Pun

The POWER PUN queue will be monitored.

Rdr

The POWER RDR queue will be monitored.

REtry=

Numeric, 0 through 9.

If the remote server refuses a connection with the automation daemon, the attempt is retried the specified number of times before the operation is abandoned and the listing placed in DISP=Y status. The default is 1.

RETRY_Time=

Numeric, 0 through 99999, in 300th-second units.

Specifies the time interval between retry attempts for a refused connection. The default is 13500 (45 seconds).

SCRIPTName=

This is the script name that will be used if HOSTname=NONE is specified to disallow inspecting JECL fields. The name must be defined to TCP/IP FOR VSE using DEFINE NAME and can be from 1 to 8 characters long. It will be applied to every report being processed. It must not contain a file type. There is no default.

SCRIPTtype=

This is the file name extension that will be added to the script name before fetching it from a library. This value follows normal Librarian rules and can be from 1 to 8 characters long. The default is 'L'.

SINGle=

This parameter applies to FTP and LPR operations only. It controls whether the output is single threaded, which means that items bound for the same destination are sent serially. This control is useful for destinations that cannot process more than one file at a time.

Yes

Items are output single threaded across all destinations. YES is forced when ORDER=YES (the default) or ORDER=JOBNUM. It is optional when ORDER=NO.

Note: The SINGLEDEST ON command is different. This command tells the automation process to single thread items bound to the same destination and associated with the same script name. But it allows other items, bound to different destinations (different script names), to be sent at the same time. See the SINGLEDEST command for more information.

No

Multiple concurrent sends can occur if the items are not ordered (ORDER=NO) and the number of items is less than the MAX event setting for that client. For example, if MAX_LPR_EVENTS is set to 3 and three sends occur, sending stops until the maximum active number is less than 3. NO is the default when ORDER=NO.

USErid=

A 1- to 16-character uppercase user ID that will be used to logon to the local FTP or LPR client. The default is "$EVENT".

Example 1 This is an example of Auto FTP.

```
set auto_time=1500
IPN268I AUTO_TIME now set to 1500 300th sec
set max_ftp=3
IPN268I MAX_FTP now set to 3
define event,id=autoftp1,type=power,queue=lst,class=x,action=ftp
define event,id=autoftp2,type=power,queue=pun,class=x,action=ftp
define event,id=autolpr1,type=power,queue=lst,class=y, -
  action=lpr,hostname=dest
TCP900I Startup ClientD PowerSysid:N PnetNodeId: Addr:00633050

 query events
IPN253I << TCP/IP Events >>
IPN587I Event Pause Interval: 1500 (5 Seconds)
IPN273I Maximum LPR Events: 5 Current Events: 0
IPN273I Maximum FTP Events: 3 Current Events: 0
IPN273I Maximum Email Events: 1 Current Events: 0
IPN426I Event ID: AUTOLPR1
IPN428I Class: Y, Queue: LST, Action: LPR, POWER SYSID:
IPN427I Priority: Yes, Order: Jobnum, Script: L
IPN429I Host field: DEST, User ID: $EVENT, Single: N
IPN430I Action: LPR, Retries: 1, Time: 45 sec
IPN426I Event ID: AUTOFTP2
IPN428I Class: X, Queue: PUN, Action: FTP, POWER SYSID:
IPN427I Priority: Yes, Order: Jobnum, Script: L
IPN429I Host field: USER, User ID: $EVENT, Single: N
IPN430I Action: FTP, Retries: 1, Time: 45 sec
IPN426I Event ID: AUTOFTP1
IPN428I Class: X, Queue: LST, Action: FTP, POWER SYSID:
IPN427I Priority: Yes, Order: Jobnum, Script: L
IPN429I Host field: USER, User ID: $EVENT, Single: N
IPN430I Action: FTP, Retries: 1, Time: 45 sec
```

Example 2

This example shows how to define an automatic LPR client and use it. First, here is a sample DEFINE EVENT for an LPR client scanning CLASS=K, with no script information contained in any field. It sets a default script name and permits the user to override that default.

```
DEFINE EVENT,ID=ALPRLSTK,TYPE=POWER,CLASS=K,QUEUE=LST,ACTION=LPR, -
       HOSTNAME=NONE,ORDER=NO,JECL=YES,SCRIPTNAME=LPRSCRD
```

After the EVENT is defined, you can issue a QUERY EVENT against it to see the new settings. If these settings were not defined, you would see the default values of these fields. And if that is the case, then JECLscript would be set to NO (the default).

Here is sample output from the query.

```
T1 0110 IPN426I Event ID: ALPRLSTK LST(K) LPR
T1 0110 IPN428I Class: K, Queue: LST, Action: LPR, POWER SYSID:
T1 0110 IPN427I Priority: Yes, Order: No, Script: L
T1 0110 IPN429I Host field: NONE, User ID: $EVENT, Single: N
T1 0110 IPN430I Action: LPR, Retries: 1, Time: 45 sec
T1 0110 IPN434I    JECLScript: Yes, Scriptname: LPRSCRD
```

You then could run the following jobs:

```
* $$ JOB JNM=AUTOLPR1,CLASS=0
* $$ LST CLASS=K,DISP=D
// JOB AUTOLPR
// EXEC LIBR
ACC S=BIMLIB.CSICNFG
LISTDIR *.*
/*
/&
* $$ EOJ
```

```
* $$ JOB JNM=AUTOLPR1,CLASS=0
* $$ LST CLASS=K,DISP=D,SCRIPT=LPRSCR2
// JOB AUTOLPR
// EXEC LIBR
ACC S=BIMLIB.CSICNFG
LISTDIR *.*
/*
/&
* $$ EOJ
```

In these two cases, the first job will be delivered with LPRSCRD as the default script name. The second job will be delivered with LPRSCR2 as the script name, thus overriding the default name, because SCRIPT= was added to the $$ LST statement. SCRIPT was defined to VSE/POWER.

Notes

The following notes apply to this command:

- LPR is a spooling protocol. This means that the remote daemon must accept the entire transmission and store it in some manner. Some printers pretend to be daemons without sufficient capacity to accept and buffer an entire transmission. When the printer runs out of paper or its buffer is exhausted, it either closes its transmission window (and LPR hangs until it can complete the transmission) or it simply stops acknowledging TCP/IP's transmissions. In the latter case, LPR eventually times out and the error recovery procedures are performed.
- According to the published standards for LPR/LPD, if a transmission fails for any reason, retransmission starts at the beginning of the file.
- If a remote LPD refuses a connection, it is considered to be a communications error and error recovery procedures are initiated.
- To prevent degrading service to other network users, auto LPR is single threaded. This is needed because a properly configured LP daemon accepts data as fast as it can be transmitted. Concurrent LPR operations run the risk of swamping a physical network. See also the SINGLEDEST command.
- Pseudo LPD devices that cannot speedily and reliably receive complete transmissions can seriously reduce throughput.
- Any report that fails to process after the requisite retries have been made is set to DISP=Y. This prevents continuous reprocessing of the same file.
- See the *TCP/IP FOR VSE User Guide* for more information about the DEFINE EVENT command and using the automatic FTP client, the automatic LPR client, and the automatic EMAIL client.
- Be sure that the USERID and PASSWORD have been created with DEFINE USER and that the entry has sufficient authority to use either the FTP or LPR clients, as appropriate (EMAIL does not require this).
- If a userid/password is required by VSE/POWER for spool access, then it can be specified by the SET POWERUSERID and the SET POWERPASSWORD commands, respectively.

Related Commands

ASECURITY FTPC
Sets whether certain FTP commands are refused.

DEFINE USER
Creates a user ID and password.

DELETE EVENT
Terminates the monitoring of a VSE/POWER class.

FLUSH
Terminates all processing with a specific remote host.

QUERY EVENTS
Displays the status of automation processing.

SET AUTO_TIME
Determines the interval for automation to check the POWER queues.

SET MAX_EMAIL_EVENTS
Establishes the maximum number of simultaneous EMAIL events for automation processing.

SET MAX_FTP_EVENTS
Establishes the maximum number of simultaneous FTP events for automation processing.

SET MAX_LPR_EVENTS
Establishes the maximum number of simultaneous LPR events for automation processing.

SET POWERPASSWORD
Establishes the password for VSE/POWER access.

SET POWERUSERID
Establishes the user ID for VSE/POWER access.

SINGLEDEST
Determines how automation processing handles multiple reports queued for the same destination (host).

DEFINE FILE

The DEFINE FILE command adds a file to the TCP/IP FOR VSE file system. The file can be a VSAM file, a VSE library, the ICCF library set, the VSE/POWER queues, a sequential file, an HFS repository, or a dataspace.

Syntax

```
DEFine FILE TYPe={ESDS|KSDS|SAM|LIBrary|ICCF|POWer|
        VSAMCAT|VTOC|HFS|DSPACE|BIM-EDIT|CONDOR|
        FALCON|VOLLIE|TAPE} [,DLBL=name8]
        ,PUBlic=pubname [,DRIVer=member]
        [,ALLOWsite={Yes|No}] [,READonly={Yes|No}]
        [,CC={Yes|No}] [,TRcc={YES|NO}]
        [,CRlf={Yes|No}] [,RECFm=F|FB|V|VB|S|SV|SU|SB]
        [,LRECL=num] [,BLKsize=num] [,GID=snum]
        [,UID=snum] [,TRANslate=name16]
        [,SITE={Yes|No}] [,DBLOCKS=num] [,EXT=name8]
        [VOLid=volser] [,CIPHER={NULL-SHA1|
        SDESCBC-NULL|SDESCBC-SHA1|TDESCBC-NULL|
        TDESCBC-SHA1|AES128C-NULL|AES128C-SHA1|
        AES192C-NULL|AES192C-SHA1|AES256C-NULL|
        AES256C-SHA1|KEYMASTER}] [,CIPHERKEY=CIALHFSK]
```

Arguments

TYPE=

Specifies the type of dataset being defined.

ESDS

A VSAM ESDS dataset. FTP read requests result in the entire file being transmitted.

KSDS

A VSAM KSDS dataset. FTP read requests result in the entire file being transmitted. Write requests are processed as VSAM INSERT operations.

SAM

A Sequential Access Method dataset.

LIBrary

A VSE Library. Performing a directory listing on the public name returns a list of the sub libraries. Further qualifying the public name with a sublibrary name (such as *public.name.sublib*) returns a list of the members. FTP read requests retrieve the contents of a member. Write requests create or replace the contents of a member.

ICCF

An ICCF library structure. Directory listing is not supported. To retrieve data, use a fully qualified name consisting of the public name, the library number, and the member name (*pubname.num.member*). Writing to an ICCF library is not supported.

Note that to access an ICCF library using FTP, your FTP userid and password MUST be identical to your ICCF userid and password.

POWer

A VSE/POWER queue. To retrieve information from VSE/POWER, specify the public name qualified with LST, RDR, or PUN, followed by class, followed by job name. Two additional qualifiers, job number and job suffix, may also be appended as needed. Writing to VSE/POWER creates a file on the specified queue.

VSAMCAT

A VSAM catalog. A directory list returns the names of all files in that catalog. Supported operations include reading, writing, creating, deleting, renaming, and appending. A full list of supported operations is provided in the *TCP/IP FOR VSE User Guide*, chapter 2, "FTP."

VTOC

An entire VSE volume. You can perform limited operations on a file defined with TYPE=VTOC. A list of operations supported by TYPE=VTOC is in the *TCP/IP FOR VSE Installation Guide*.

HFS

CSI's Hierarchical File System. This file type supports a PC or Linux-type file system with multiple subdirectory levels and long file names.

DSPACE

Causes a dataspace to be allocated and used as a file. See the *TCP/IP FOR VSE Installation Guide* for more information.

BIM-EDIT

CSI International's BIM-EDIT product. This type requires a BIM-EDIT license, and you must link the .OBJ file to make the I/O driver module usable. You must use DEFINE FILEIO to load the BIM-EDIT I/O driver module into storage. See DEFINE FILEIO for a note about BIM-EDIT and the TCP/IP startup sequence.

CONDOR, FALCON, VOLLIE

File types associated with non-CSI products. For each type, you must obtain a vendor-provided I/O driver module and install it. See "CA Vollie™ File Example," page 64, for more information.

TAPE
A tape drive file. You must use DEFINE FILEIO to load the associated TAPE I/O driver module into storage.

DLBL=
A DLBL name. This specifies the DLBL statement to be used to open the file. It must be accessible to the TCP/IP FOR VSE partition. Files of type "POWER" do not require or use a DLBL.

PUBlic=
A unique name that identifies this dataset to users. This name follows the same rules as VSE dataset names, so you can set the public name equal to the actual dataset name. For a discussion of public names, see the *TCP/IP FOR VSE Installation Guide*, chapter 1, "Fundamentals of TCP/IP."

DRIVer=
If specified, this phase is loaded from the library search list when the dataset's driver is initialized. In general, you should let this value default to the phase provided for the file type.

If the phase you need is not listed by QUERY FILEIO, you must use DEFINE FILEIO to load the phase before you can specify it in DEFINE FILE.

READonly=
This setting can be used to selectively limit FTP access to this file.

No
FTP users may write to and update this file, subject to other security procedures. This is the default.

Yes
Users do NOT have write or update access to this file, regardless of other security procedures.

ALLOWsite=
FTP SITE commands that are specific to a file type (for example, SITE PALTER) are passed to the file I/O driver of the file currently being accessed. Coding NO for this parameter overrides whatever may be specified in the FTPD definition and can be used to prevent unauthorized commands from being issued.

No
FTP users may NOT pass SITE commands to the file I/O driver associated with this file.

Yes
FTP users may pass SITE commands to the file I/O driver associated with this file.

EXT=

For TYPE=VSAMCAT, specifies a "suffix" to be appended to the file name. For example, EXT=".TXT" would ensure that references to the files in the catalog would all be assumed to be "text."

CC=

Provides a default value for the FTP SITE CC command. If not specified, the value is supplied by the client or user.

Note: This CC setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

Yes

The first byte of each record is assumed to be a carriage control byte, and this byte is retained as part of the data.

No

During downloads (from VSE), the first byte of each record is discarded. During uploads (to VSE), a blank character is prefixed to each record.

TRcc=

Provides a default value for the FTP SITE TRCC command. If not specified, the value is supplied by the client or user.

Note: This TRCC setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

No

No special processing is performed to simulate carriage control.

Yes

ANSI carriage control codes (+, 0, -, 1) cause simulation of the CC character. Forms control characters (CR, LF, FF) are added to the output records as needed.

CRlf=

Provides a default value for CRLF record delimiter processing. If not specified, the value is supplied by client or user.

Note: This CRLF setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

Yes

Each record uploaded to VSE must be ended by an appropriate delimiter, generally CR/LF. On download, the appropriate delimiter is added to each record.

No

No delimiter(s) is added on download and no delimiter(s) is expected on upload. For incoming records, the data stream is divided based on the LRECL value.

RECFm=

Provides a default value for the FTP SITE RECFM command.

Specify a record format of F, FB, V, VB, or S. (The string format is valid only with Librarian files.) This value does NOT override the RECFM value specified on the DLBL definition. If not specified on either the DLBL or DEFINE FILE, then it can be supplied using an FTP SITE command.

Note: This RECFM setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

LRECL=

Provides a default value for the FTP SITE LRECL command.

This is the logical record length. It is a numeric and must be consistent with the value used when the file was created. The default is 80. See the tables that follow for information about LRECL selection. This value is not obtained from DLBL information and must be provided in a DEFINE FILE or with an FTP SITE command.

Note: This LRECL setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

BLKsize=

Provides a default value for the FTP "SITE BLKSIZE" command.

This value is the block size used in the dataset. This information must be consistent with the value used when the file was created. There is no default. See the tables in the sections below for information about BLKSIZE selection. This value is not obtained from DLBL information and must be provided by DEFINE FILE or with an FTP SITE command.

Note: This BLKSIZE setting can be overridden. See the "RECFM, LRECL, BLKSIZE" section, page 60, for details.

TRANslate=

Provides a default value for the FTP SITE TRANSLATE command when the file must be translated between ASCII and EBCDIC mode. If omitted, the default is the TCP/IP FOR VSE default translate table.

To be valid, the named translation table must already be loaded at the time the file is opened rather than when the DEFINE FILE is issued.

The DEFINE TRANSLATION command explains the definition and loading of translate tables.

SITE=

SITE commands are processed hierarchically. The FTP daemon handles most commands itself. Any command that is not understood by the daemon is passed to the file I/O driver of the currently selected file, as determined by the last-issued CD command.

No

FTP SITE commands that are not recognized by the FTP daemon are not passed to the file-specific driver routine.

Yes

FTP SITE commands that are not recognized by the FTP daemon are passed to the file I/O driver routine for possible processing. This is the default.

GID=

Signed numeric, -9999999 through +9999999.

Defines this file as part of a group. TCP/IP FOR VSE does not use this field but passes it to the TCP/IP FOR VSE security exit. A GID value may also be assigned with DEFINE USER.

UID=

Signed numeric, -9999999 through +9999999.

Associates this file with a UNIX-style user ID. TCP/IP FOR VSE passes this field to the TCP/IP FOR VSE security exit. A UID value may also be assigned with the DEFINE USER command.

DBLOCKS=

For TYPE=DSPACE, specifies the amount of space to be allocated to the virtual file.

VOLid=

For TYPE=VTOC, specifies the volume ID of the disk.

CIPHER=

For TYPE=HFS, indicates that files are to be stored in an encrypted form using the specified method. If this parameter is omitted, then no encryption or decryption is performed.

CIPHERKEY=

For encrypted TYPE=HFS files, this keyword provides the name of the phase that contains information on the encryption keys. The default phase is shipped with TCP/IP and contains sample keys. You should create your own phase with your own keys before using this feature in a production environment. More information is available in the *TCP/IP FOR VSE Programmer's Guide.*

Example

```
define file,public='power',type=power
IPN264I File defined, Dataset: POWER

define file,public='tcpip',dlbl=tcpip,type=library
IPN264I File defined, Dataset: TCPIP

define file,public='prd1',dlbl=prd1,type=library
IPN264I File defined, Dataset: PRD1

define file,public='prd2',dlbl=prd2,type=library
IPN264I File defined, Dataset: PRD2

define file,public='ijsysrs',dlbl=ijsysrs,type=library
IPN264I File defined, Dataset: IJSYSRS

define file,public='analyze',dlbl=analyze,type=sam, -
recfm=fb,lrecl=4096,blksize=4096
IPN264I File defined, Dataset: IJSYSRS
```

RECFM, LRECL, BLKSIZE

Acceptable values for RECFM, LRECL, and BLKSIZE depend on how you originally defined the file, the access method you use, and the mode of access (reading or writing). 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:

If EXTTYPES processing is NOT in effect and SITELAST=YES in FTP, then SITE commands take precedence over the following DEFINE FILE parameters: BLKSIZE, CC, CRLF, LRECL, RECFM, and TRCC. See also chapter 6, “Configuring FTP Daemons,” in the *TCP/IP FOR VSE Installation Guide*.

The sections that follow contain tables that show acceptable values for RECFM, LRECL, and BLKSIZE for specific file types. The “Input” headings indicate that you are reading from disk, and the “Output” headings indicate that you are writing to disk. These terms do not indicate whether you are using the FTP client or daemon.

Sequential Disk File and VSAM-Managed SAM File Considerations

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.
- Although IBM's VSAM-managed SAM files will appear to be ESDS files when performing an IDCAMS LISTCAT of the VSAM catalog, it is recommended that you read them as SAM files rather than ESDS files. This is because the IBM routines that perform the SAM output of the file often will not correctly update the catalog after the file is closed. This may result in incomplete transfers when using certain graphic FTP clients.
- RECFM SU is interpreted as a spanned unblocked file. RECFM SB is interpreted as a spanned blocked file.

The following table shows appropriate values for this file type.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
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
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

VSAMCAT File Considerations

For VSAMCAT files, note the following information.

- For output files, if the file does not already exist, then the SITE command will establish the parameter values used in the DEFINE CLUSTER command that will be passed as a subtask to the IBM IDCAMS utility prior to writing to the file.

  If the output file already exists, then the SITE commands will be ignored and the IDCAMS utility will not be invoked. This means that the SITE commands you use must match the expected attributes.

- For input files, your SITE commands need not match the attributes of the existing file. This is true for all VSAM file types.

- Fixed-length records are padded if necessary when writing to the VSE/POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with binary zeros.

- If you use the "blocked" type (VB or FB) for output, then the "(*nnnn*)" parameter of the DEFINE CLUSTER RECFM command passed to the IDCAMS utility will be provided, where *nnnn* is the record length. Otherwise, use the "F" or "V" record format.

The following table shows appropriate values.

	Input		**Output**	
recfm	***lrecl***	***blksize***	***lrecl***	***blksize***
F	record size	N/A	N/A[1]	N/A[1]
V	maximum record size	N/A	N/A[1]	N/A[1]

[1] Depends on whether the output file exists. SITE command parameters are passed to the IDCAMS utility only upon file creation.

ESDS, KSDS File Considerations

For ESDS and KSDS VSAM files, note the following information.

- Fixed-length records are padded if necessary when writing to the VSE/POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with zeros.

The following table shows appropriate values.

RECFM	**Input LRECL**	**Input BLKSIZE**	**Output LRECL**	**Output BLKSIZE**
F	record size	N/A	N/A	N/A
V	maximum record size	N/A	N/A	N/A

TAPE File Considerations

For TAPE files, note the following information:

- Fixed-length records are padded when necessary. When padding occurs, tape 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.

The following table shows appropriate values for TAPE files.

	Input		**Output**	
recfm	***lrecl***	***blksize***	***lrecl***	***blksize***
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
UN	maximum record size	N/A	maximum record size	N/A

VSE/POWER File Considerations

For VSE/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 for this file type.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	Record size	N/A	N/A	N/A
V	Maximum record size	N/A	N/A	N/A

ICCF File Considerations

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 this file type.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	N/A	N/A	N/A	N/A

Library File Considerations

For Library files, note the following information:

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

The following table shows appropriate values for this file type.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	80	N/A	80	N/A
SV	Variable	N/A	Variable	N/A
S	N/A	N/A	N/A	N/A

CA Vollie™ File Example

The following example shows how to define a CA Vollie file.

1. Locate the IPNFVOLL.OBJ in OEM.TCPIP (a CSI sublibrary).
2. LINK the object file into a phase, place it in OEM.CONFIG (a sublibrary CSI searches), and then catalog the phase as OLLFILE.

3. Cycle TCP/IP FOR VSE, then issue a DEFINE FILE:

```
DEFINE FILE,TYPE=VOLLIE,DLBL=OLLFILE,PUBLIC='OLLFILE'
```

The TYPE=VOLLIE file can now be used. For example, you could use TCP/IP FOR VSE's EMAIL client to attach a CA Vollie member to an email.

Related Commands

AUTOLOAD
Determines automatic loading of file I/O drivers when files are defined.

DEFINE FILEIO
Loads a file I/O driver phase into storage.

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

DELETE FILE
Removes a file from the TCP/IP FOR VSE file system.

MODIFY FILE
Changes parameters and characteristics of an entry in the TCP/IP FOR VSE file system.

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY FILEIO
Displays the status of the file I/O driver programs.

SECURITY
Controls TCP/IP FOR VSE security functions.

DEFINE FILEIO

The DEFINE FILEIO command loads a file I/O driver phase into storage. See the notes below for guidance in using this command.

Syntax

DEFine FILEIO TYPe=*type* [,DRIVer=*phase-name*]

Arguments

TYPE=

A label that appears in QUERY FILEIO output. See the DEFINE FILE command for a list of valid file types.

Note: This command does not apply to vendor-supplied driver modules for non-CSI file types: CONDOR, FALCON, and VOLLIE. See "CA Vollie™ File Example," page 64, for the steps you use to install such a module.

DRIVer=

If specified, this phase is loaded from the library search list when the dataset's driver is initialized. If it is not specified, a default phase is used.

Example

```
define fileio,type=BIM-EDIT
IPN264I File driver defined, Dataset: BIM-EDIT

 define fileio,type=LIBRARY,PHASE=MYFILEIO
IPN264I File driver defined, Dataset: LIBRARY
```

Notes

The following notes apply to this command:

- A file I/O driver is an external program that interfaces with the TCP/IP FOR VSE stack and controls the acquisition and delivery of data. Driver modules for the most common file types, such as LIBR and POWER, are automatically loaded when the stack initializes. (AUTOLOAD must be set to ON.) The DEFINE FILEIO command is needed to load any driver module that is not automatically loaded. Once a module is loaded, that specialized access program will be available to TCP/IP FOR VSE applications. For example, you must use DEFINE FILEIO to load a driver module for type BIM-EDIT.
- Use DEFINE FILEIO to load these module types:
  - A driver module for a file type associated with a CSI International product such as BIM-EDIT. This type requires a BIM-EDIT license, and you must link the .OBJ file to make the driver module usable. See also "**For BIM-EDIT**" below.
  - A user-defined driver module.

- All file types have a default driver phase. If you override the default by specifying a driver phase when you define a file (using DEFINE FILE), you may need to use DEFINE FILEIO to load that phase into storage. Use QUERY FILEIO after system initialization to check which drivers are loaded. Multiple driver phases may be associated with a given file type.

- In the example above, one LIBRARY-type entry is associated with the phase MYFILEIO. Another LIBRARY-type entry is associated with the default phase IPNFLIBR, which is loaded at startup.

- DEFINE FILEIO is not normally needed. You should use it only if you are having trouble using a specific file-access method, such as BIM-EDIT, to access a file system. Use QUERY FILEIO to first check whether the needed driver, with the designated file type, is loaded. If it is not loaded, use DEFINE FILEIO. If it is already loaded, however, then you know that a missing driver is not the problem.

- **For BIM-EDIT:** The order of the startup of TCP/IP and BIM-EDIT is important for the BIM-EDIT interface to work properly. TCP/IP must follow BIM-EDIT. If TCP/IP was started before BIM-EDIT, then it may be necessary to issue the following statements:

```
DELETE FILEIO,ID=IPNFBIME
DEFINE FILEIO TYPE=BIM-EDIT
```

  even though it may appear that the file IO driver for BIM-EDIT was loaded at TCP/IP startup time.

  You then need a DEFINE FILE statement such as

```
DEFINE FILE,PUBLIC='BIMEDIT',DLBL=BIFLIB,TYPE=BIM-EDIT
```

Related Commands

AUTOLOAD
Controls automatic loading of file I/O drivers.

DEFINE FILE
Defines a file in the TCP/IP file system and associates it with a file I/O driver.

DELETE FILEIO
Removes a file I/O driver from storage.

QUERY FILEIO
Displays the status of the file I/O driver programs.

QUERY FILES
Displays the contents of the TCP/IP file system.

DEFINE FTPD

The DEFINE FTPD command initiates an FTP daemon (server).

Syntax

```
DEFine FTPd ID=id [,PORT=21] [,MAXACTive=3]
        [,UNIX={Yes|No|Binary}] [,TRANslate=name16]
        [,TIMEOut=2m] [,BSize=64K] [,WELCOME=member]
        [,EXTtypes={Yes|No}]
        [,EXTRADATA={FAIL|WARN|IGNORE|ACCEPT}]
        [,DYNfiles={Yes|No}] [,ALLowabort={Yes|No}]
        [,HESitate=0] [,IDLEtimeout=0]
        [,SITELAST={Yes|No}] [,SSL={YES|NO|YESCLAuth}]
        [,SSLKEY=member]
        [,SSLVERsion={SSL30|TLS10|TLS11|TLS12}]
        [,SSLCIPHER={ALL|WEAK|STRONG|AES|DES|NULL|
          HARDware|MEDIUM}]
        [,SSLMODE={IMPLICIT|EXPLICIT}]
        [,SSLDATAconn={CLEAR|PRIVATE}]
        [,ZEROerr={Yes|No}] [,IPaddr=ip4addr]
        [,UPPERcase={Yes|No}] [,SENDFast={Yes|No}]
        [,REXX={Yes|No}] [,SENDWack={Yes|No}]
```

Arguments

ID=

A unique name to identify this daemon.

PORT=

Specifies the TCP/IP port number to be monitored by this FTP daemon. The default port is 21. You can specify any value between 0 and 65535, but you should avoid values below 4096 to prevent collision with ports that have standard uses. FTP uses the specified port to establish the control connection. When data is transferred, another port is assigned for the data connection.

BSize=

The size of the buffers that FTP will use. The default size is 65536. You may override to any value between 4096 and 131072.

EXTtypes=

Most GUI FTP clients “lie” about file types. They always specify “binary” transfers. Because this will not work if the file contains text, the External Types Table maps files to specific file types by checking the extension. Default transfer values for that file type are then used.

Yes
(Default) The FTP daemon consults the EXTTYPES.L member (loaded by RELOAD EXTYPES), looks for a file type entry that matches any part of the file name, and uses assigned transfer values if a match is found. These values override other settings.

No
The FTP daemon does not consult the EXTTYPES.L member. All processing options must be specified explicitly. See also SITELAST= for more information on specifying options.

UPPERcase=
The RFCs that define FTP behavior state that the client can only rely on the numeric value of each reply. The text is for human use only. If translation issues require it, then this parameter ensures that text is sent in uppercase only.

Yes
Causes responses to the client to be in uppercase.

No
(Default) Responses to the client are in mixed case.

SENDWack=
Normally, the FTP daemon transmits data as quickly as it can be written to the network. Depending on file system overhead, however, this can monopolize the CPU and/or network. This parameter can be used to cause the daemon to pace outbound transmission.

Yes
Causes the daemon to wait for a response from the remote client after each transfer buffer is sent. This helps pace transmissions to prevent monopolizing the CPU and network.

No
(Default) The daemon does not automatically wait for each buffer to be acknowledged before filling another one.

REXX=
It is possible for FTP users to execute REXX procedures in the TCP/IP partition. Doing so can expose the stack to delays and instability, however, so use caution if you set this option to YES.

Yes
Permits users to execute REXX execs using the SITE REXX command. Be aware that allowing this feature exposes the stack to potential security, performance, and integrity issues.

No
(Default) The SITE REXX command is not permitted.

SENDFast=

Normally, the FTP daemon fills one buffer while the other is being transmitted. This generally is sufficient to keep data flowing at its maximum speed. However, if system configuration is such that this provides inadequate buffering, an alternate method allows additional buffering.

Yes

Transfer buffers are filled and queued until the total bytes queued reaches a value equal to four transfer buffers.

No

(Default) Two transfer buffers are filled and queued for outbound transmittal. A buffer is not refilled until the remote stack acknowledges receipt of the data.

HESitate=

This value is used to pace data transmission when there is concern that the data transfer rate may exceed what the network and system can support. When in effect, a pause of the specified length occurs following each SEND and RECEIVE. Allowable values are 0 through 99999 (300^{th}-second units).

A value of 0 (the default) prevents pacing.

IDLEtimeout=

If non-zero, this specifies the maximum time (in 300^{th}-second units) that an FTP session can remain idle before being terminated. Idle time does not accrue while a data connection is open. Allowable values range from 0 (default) to 999999 (~56 minutes).

SSL=

When defining an FTP daemon, you must choose whether connections will be normal (unencrypted, clear text) or secure (encrypted and authenticated at some level). Use of SSL requires that the client and remote host also support encryption. When SSL is enabled, it is a normal practice to use a port other than the standard value of 21.

NO

(Default) Encrypted connections are not supported.

YES

Connections will be encrypted.

YESCLAUth

Connections will be encrypted and the client must provide an authorization certificate.

SSLKEY=

The *library.sublibrary.member* from which SSL processing will obtain members of types “.prvk”, “.cert”, and “.root.”

SSLVERsion=

This option specifies the minimum version of the TLS/SSL protocol that clients must use when connecting.

TLS12

Synonym: 0303
TLS12 is the most secure, but not all clients can support it.

TLS11

Synonym: 0302
TLS11 is more enhanced.

TLS10

Synonyms: 0301, TLSV1
The TLS protocol contains significant security corrections and enhancements over SSL30.

SSL30

Synonyms: 0300, SSLV3

SSLCIPHER=

For SSL/TLS sessions, this parameter determines which ciphers will be available. The value depends on the SSL/TLS protocol version used.

The valid values for the **SSL30, TLS1.0,** and **TLS1.1** protocol versions are as follows:

ALL

(Default) Use the suite(s) set by the SSLCIPH= option in $SOCKOPT.PHASE. In the default phase, the SSLCIPH= option is set to 'A', which allows all supported cipher suites to be used. For more information, see "Appendix A: $SOCKOPT Options Phase" in the *TCP/IP FOR VSE Programmer's Guide*.

WEAK

The following weak ciphers are permitted:

- 09 RSA_DESCBC_SHA
- 08 RSA_DES40CBC_SHA1

MEDIUM

The following medium strength ciphers are permitted:

- 2F RSA_AES128CBC_SHA
- 0A RSA_3DESCBC_SHA
- 09 RSA_DESCBC_SHA

STRONG

The following strong ciphers are permitted:

- 35 RSA_AES256CBC_SHA
- 2F RSA_AES128CBC_SHA

— 0A RSA_3DESCBC_SHA

AES
The following ciphers are permitted:

— 35 RSA_AES256CBC_SHA

— 2F RSA_AES128CBC_SHA

DES
The following ciphers are permitted:

— 0A RSA_3DESCBC_SHA

— 09 RSA_DESCBC_SHA

— 08 RSA_DES40CBC_SHA1

NULL
The following null ciphers are permitted:

— 02 RSA_NULL_SHA1

— 01 RSA_NULL_MD5

HARDware
When HARDware is specified, the Crypto Assist hardware will be queried for all available CP Assist for Cryptographic Function (CPACF) assists. Use caution when selecting this option to ensure that your site's CPU supports the KMC query instruction.

The valid values for the **TLS 1.2** protocol version are as follows. These suites do not support the DES algorithm. TLS 1.2 support also requires IBM's hardware CPACF feature.

ALL
(Default) Use the suite(s) set by the SSLCIPH= option in $SOCKOPT.PHASE. In the default phase, the SSLCIPH= option is set to 'A', which allows all supported cipher suites to be used. For more information, see "Appendix A: $SOCKOPT Options Phase" in the *TCP/IP FOR VSE Programmer's Guide*.

WEAK
The following weak cipher is permitted:

— 2F RSA_AES128CBC_SHA160

MEDIUM
The following medium-strength ciphers are permitted:

— 35 RSA_AES256CBC_SHA160

— 3C RSA_AES128CBC_SHA256

STRONG
The following strong cipher is permitted:

— 3D RSA_AES256CBC_SHA256

SSLMODE=
Sets the negotiation mode. The values are as follows.

IMPLICIT
(Default) The SSL/TLS negotiation is performed immediately when the connection is established (before the 220-welcome message is sent).

EXPLICIT
The SSL/TLS negotiation is delayed until an AUTH command is received. This is somewhat less secure because the initial 220-welcome message is sent in clear text.

SSLDATAconn=
This option controls encryption on the data connection.

CLEAR
(Default) Causes data to be transmitted in the clear, without encryption.

PRIVATE
Causes data to be transmitted in an encrypted state.

SITELAST=
This option determines a user's ability to override certain values that may have been specified on individual DEFINE FILE commands.

Yes
SITE commands take precedence over the following DEFINE FILE parameters: BLKSIZE, CC, CRLF, LRECL, RECFM, TRCC.

Note: The default transfer values assigned by the EXTTYPES table for a matched file type always take precedence when EXTTYPES=YES.

No
(Default) Parameters specified on DEFINE FILE commands take precedence over a user's SITE command settings.

DYNfiles=
In some cases you may want to allow users to specify files by DLBL rather than restrict them to using public names defined in the TCP/IP file system.

Yes
(Default) Users can access files outside of the TCP/IP file system (autonomous files) by specifying a file's DLBL. For details, see the *TCP/IP FOR VSE User Guide,* chapter 2, "FTP," subsection "VSE File Names."

No
Specifying NO prevents access to files by DLBL, restricting access to only those files defined by the DEFINE FILE

command. This is the recommended setting to prevent unintended access to files.

ALLowabort=

Once data transmission begins, the control connection remains idle until it completes. An exception is the "abort" command. If enabled, the daemon monitors the control connection during data transfer and allows the client to abort the transfer.

Yes

(Default) The client can send an abort command (ABRT) to cause the current data transmission to stop.

No

The daemon ignores any input on the control connection until the data transfer completes.

EXTRADATA=

This option controls what is done when an incoming text file has extra data at the end. "Extra data" (which can only occur in a text file) is defined as a text string that is not correctly ended with a CR/LF or other valid delimiter.

FAIL

(Default) The transfer fails and the file is not stored. An FTP343W message occurs, and a 5*xx* failure code is sent to the client.

WARN

A warning message is generated, but the file is stored. A normal code is sent to the client. The extra data is discarded.

IGNORE

No messages are generated. The file is stored and the extra data is discarded.

ACCEPT

No messages are generated. The file is stored with the extra data as if it were a correctly delimited, complete record.

MAXACTIVE=

Each FTP daemon supports multiple concurrent sessions. This parameter allows you to set the maximum number of user sessions that can be active at any given time. Values range from 1 to 32767. The default is 3.

UNIX=

In general, GUI FTP clients interpret and display information related to the FTP session. In order for this to happen, the information provided by the daemon must be in a form that is expected by the client. The expected format is that produced by a Unix daemon. The valid values are as follows.

Binary
When the FTP client specifies “BINARY” (TYPE I), a true binary transfer takes place. The DIR command returns a directory list in the standard Unix format that is understood by GUI clients.

Yes
The FTP daemon mimics Unix operation. This is important if you use graphical (GUI) FTP clients because many of these clients cannot recognize a VSE-style directory list.
To discontinue operating in Unix emulation mode, the user can enter the FTP “CD \” command or use “SITE UNIX OFF.”

No
The FTP daemon operates in VSE mode. This is the default.

Directory lists are returned in “VSE” format with information pertinent to mainframe-based files. To enter Unix mode, the user can enter a “CD /” command or use “SITE UNIX ON.”

WELCOME=
This option specifies the VSE member that contains site-specific text that is sent to the user along with the 220 message at the start of each new session. This member must be cataloged. For example, if the member name is “GREETING,” then “GREETING.L” must be catalogued in the LIBDEF chain as part of the TCP/IP initialization. No text in columns 73 through 80 is used.

TRANslate=
Specifies the name of an optional translate table for ASCII/EBCDIC translation.

This specification overrides the SITE TRANSLATE command. It does not override the TRANSLATE= parameter of the DEFINE FILE command associated with the transfer. If you want FTP users to be able to specify a translate table, omit this parameter on the DEFINE FTPD command.

TIMEOut=
Specifies a time-out interval for the data connection in 300th-second units. Although the FTP protocols do not recognize the concept of a time-out, it is essential from a practical standpoint. When the time interval is exceeded without a response of any kind from the remote host, the FTP session is terminated. This occurs only during actual data transmission; the daemon will wait for the IDLETIMEOUT value on a new command. The default value is 36000 (2 minutes); the maximum value is 999999 (~56 minutes).

ZEROerr=
This parameter determines how the daemon reacts when it is asked to process an empty (null) file.

Yes
(Default) A 500-level error message (a fatal error) is generated when a user attempts to transfer an empty file (one with zero data bytes).

No
The transfer of an empty file is considered acceptable.

IPaddr=
If specified, the remote IP address in an inbound connection request is compared against this value. Depending on how the CONNECT_SEQUENCE command is set, selection will be either first fit or best fit.

Matching is not performed when both the client and the daemon execute under the same stack.

Example

```
define ftpd,id=ftp01,port=21,unix=yes,welcome=greet1,maxact=5
FTP900I FTP Daemon: FTP01 listening on 192.168.1.161,21
```

Notes

The following notes apply to this command:

- There is no MODIFY FTPD command. To change specifications, you must delete and redefine the FTP daemon.
- File transfers can easily monopolize your network interface and even the network itself. Running multiple concurrent file transfers can result in reduced transfer speed and can degrade network response time. Consider this carefully when determining the number of FTP daemons that you will define. See the *TCP/IP FOR VSE Installation Guide* for more information about defining FTP daemons.
- The FTPBATCH utility program can be configured to run as an FTP server. This allows you to adjust its processing priority to a lower value without affecting the time-dependent portions of the stack. See the *TCP/IP FOR VSE Installation Guide* for details on configuring an FTPBATCH server.
- See the *TCP/IP FOR VSE User Guide* for information about running in UNIX compatibility mode.
- See “SecureFTP for VSE” in the *TCP/IP FOR VSE Optional Features Guide* for more information about using the SSL=, SSLKEY=, SSLCIPHER=, SSLVERSION=, SSLMODE=, and SSLDATACONN= parameters.
- The default ciphers are set by the SSLCIPH keyword in $SOCKOPT. For more information, see the *TCP/IP FOR VSE Programmer’s Guide*, “Appendix A: $SOCKOPT Options Phase.”

Related Commands

ASECURITY FTPC
Sets whether certain FTP commands are refused.

CONNECT_SEQUENCE
Controls whether connection requests are allocated by IP address pattern-checking.

DEFINE FILE
Defines a file in the TCP/IP file system and associates it with a file I/O driver.

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

DELETE FTPD
Terminates a File Transfer Protocol daemon.

PORTRANGE
Controls the range for dynamic port assignment.

QUERY ACTIVE
Displays the status of active daemons.

QUERY FTPDS
Displays the status of the File Transfer Protocol daemons.

RELOAD
Reloads a control table.

DEFINE GPSD

This command is used to define and initiate processing by a General Print Server (GPS) daemon. The full syntax statement follows. See the individual parameter definitions for their application in specific situations.

Syntax

```
DEFine GPSd ID=id ,IPaddr=host ,TERMname=lu
        [,ALTlength=132] [,BRACKet_eject={Yes|No}]
        [,CMDn=string] [,CONTrol_order=NFU]
        [,DEBug={Yes|No}] [,EMULate={3287|Transparent}]
        [,INSerts=member] [,INSESsion={No|Yes}]
        [,LINELength=132] [,LOG={No|Yes}]
        [,LOGMode=mode] [,MAXChars=1m] [,MAXIdle=10s]
        [,MAXLines=10k] [,MAXPages=1k] [,NRT=60s]
        [,NRC=3] [,NOEJect={No|Yes}]
        [,OUTput={Lpr|Direct} [,PORT=515]]
        [,PRinter=name] [,Queuing={Disk|Memory}]
        [,STORage=pubname] [,TARGet=appl]
        [,TRANslate=name16] [,TYPE=VTAM] [,VRC=10]
        [,VRT=60s]
```

Arguments

ALTLength=

Specifies the line length to be used if an application uses the ERASE WRITE ALTERNATE command. If an application issues an ERASE WRITE ALTERNATE command, the 3270 printer is set to its alternate characteristics. See the LINELEN parameter description for more information about GPS and line lengths. This is not used with EMULATE=TRANSPARENT.

BRACKET_EJECT=

GPS normally interprets a VTAM “begin bracket” as causing a form feed. If this behavior is not appropriate in your environment, code NO. Not used with EMULATE=TRANSPARENT.

Yes

GPS performs a page eject whenever a VTAM “begin bracket” is received.

No

GPS ignores a VTAM “begin bracket.”

CMD*n*=

Three commands, CMD1=, CMD2=, and CMD3=, are provided to support intermediate releases of GPS. Parameter values and actions will vary with each release. Use this argument only when directed by CSI Technical Support.

CONTrol_order=

A string of 1 to 3 characters: U, F, and N.

During LPR/LPD processing, a control file is constructed and transmitted. This file contains commands such as “print” and “delete”. Although the RFCs are not clear as to a required order, the default value of “NFU” works with the vast majority of LPD daemons. If required, you can modify the order or omit one of the commands. This parameter is used only with OUTPUT=LPR.

N

The name of the print file in question.

F

The “print” command.

U

The “delete” command.

DEBug=

GPS provides a debugging mode that captures and saves additional data.

When debugging is active, raw VTAM transmissions are saved in the storage file. You can review the data to see precisely what GPS has to work with. When you are finished, you must manually delete these files. Note that the debug mode may require considerable library space, depending on the amount of data being printed.

Yes

Run this GPS daemon in debug mode. You must also include the STORAGE= parameter. Also note that coding YES causes each daemon to require an additional 100k (approx.) of 24-bit storage.

No

(Default) Run this daemon in normal mode.

EMULate=

This parameter controls the type of printer emulation performed.

3287

(Default) GPS emulates a 3287 printer to the extent practical.

Transparent

GPS transmits the data exactly as received from the VTAM connection. The command character and the Write Control Character (WCC) are removed. The remainder of the stream is treated as a “page.” No translation to ASCII is performed.

ID=

This is a unique identifier assigned to the GPS daemon. This identifier appears in messages and displays pertaining to this daemon.

INSERts=

If specified, this phase is loaded and its contents are included in the data sent to the printer. INSERTS data includes control information that can precede a report, follow a report, and follow each embedded form feed.

INSESsion=

Determines whether the daemon attempts an immediate connection with the TARGET= application or waits for an external connection request.

Note that if the application releases the bind, the daemon makes no additional bind attempts, regardless of this setting.

Yes

The GPS daemon is directed to attempt to bind with the application specified in the TARGET= parameter immediately at startup.

No

(Default) The GPS daemon waits for the application to initiate the bind request.

IPaddr=

This is the IP address of the remote host to which the data is to be transmitted. If a host name is specified, it may be up to 64 characters.

LINELength=

This sets the maximum printer line length. The 3287 emulation provides for a maximum line length of 132. When the line is filled, the printer forces a newline operation. If your application needs to print longer lines, you can increase the printer's maximum line length to as many as 255 characters. Not used with EMULATE=TRANSPARENT.

LOGMode=

This sets the VTAM LOGMODE that is to be used to negotiate a bind with the VTAM application. The default (and recommended) value is DSC2K. This is an IBM-supplied, non-SNA LOGMODE for printers. This parameter is effective only if you also specify the INSESSION=YES and TARGET= parameters.

LOG=

This option controls the use of a logging file.

Yes

Causes the GPS daemon to create a log file. The log file name is the value specified for TERMNAME with an extension of LOG. This file is overwritten each time the daemon restarts. The file is a simple text file, and you can use standard VSE facilities to view or print it. You must also include the STORAGE= parameter. Also note that coding YES will cause each daemon to require an additional 100k (approx.) of 24-bit storage.

No

(Default) No log file will be opened or used.

MAXChars=

The maximum characters that can accumulate before an LPR operation is forced. Because LPR is a spooled protocol and GPS is emulating a serial device, GPS uses the MAXCHARS parameter as one way of determining when to segment the report and transmit it using LPR. The disadvantage of triggering by character count is that extraneous page breaks are introduced. The default is 1,000,000 characters; the maximum is 999,999,999.

This parameter has effect only if OUTPUT=LPR is also specified.

MAXIdle=

When the LPR/LPD protocol is being used, this parameter specifies how much idle time (in 300^{th}-second intervals) can elapse before an LPR operation is forced. Because LPR is a spooled protocol and GPS is emulating a serial device, GPS uses the MAXIDLE parameter as one way of determining when to segment the report and transmit it using LPR. “Idle time” is defined to be the time between VTAM transmissions to GPS. The value should be set high enough that normal processing delays do not cause premature transmission of the report and extraneous page breaks.

If the direct socket interface is being used, this parameter indicates the elapsed idle time before the printer connection is closed. Note that the printer itself may drop the connection after a predetermined idle interval. If this happens, error-recovery procedures will occur. Typically, a printer’s time-out is between 45 and 90 seconds, so selecting a reasonable MAXIDLE value will avoid this.

The default is 3,000 (10 seconds). The maximum value is 99,999.

MAXLines=

Specifies the maximum text lines that can accumulate before an LPR operation if forced. Because LPR is a spooled protocol and GPS is emulating a serial device, GPS uses the MAXLINES parameter as one way of determining when to segment the report and transmit it using LPR. The disadvantage of triggering by line count is that extraneous page breaks are introduced. The default is 10,000 lines. The maximum permitted is 9,999,999.

This parameter has effect only if OUTPUT=LPR is also specified.

MAXPages=

Specifies the maximum pages that can accumulate before an LPR operation if forced. Because LPR is a spooled protocol and GPS is emulating a serial device, GPS uses the MAXPAGES parameter as one way of determining when to segment the report and transmit it using LPR. The advantage of triggering by page count is that no extraneous page breaks are introduced. The default is 1000 pages. The maximum permitted is 99,999.

This parameter has effect only if OUTPUT=LPR is also specified.

NRT=

This is the time interval that is used between retries of network operations, including LPR/LPD. Values range from 0 through 550 minutes. This value may also be specified as "NETWORK_RETRY_Time=". The default is 60s.

NRC=

This value controls how many times potentially retriable network failures will be retried, including LPR/LPD. Allowable values range from 0 through 99,999. This parameter may also be specified as "NETWORK_RETRY_Count=". The default value is 3.

NOEJect=

Many reports begin with a page eject character. GPS normally passes this character through to the printer. Depending on the printer or LP daemon, however, this may create an extra blank page. Note that GPS ends each report with a form feed to force printing of the final page.

Not used with EMULATE=TRANSPARENT.

Yes

Specifying YES suppresses the initial form feed character at the beginning of a listing.

No

Specify NO if you do not want to suppress initial form feed characters

OUTPut=

A keyword that controls how GPS connects with the remote host.

Lpr

(Default) GPS accumulates data into "reports" of appropriate length and then transmits these reports to a remote LPD using the LPR protocol.

Direct

GPS connects directly with a "network" printer. Data is transmitted as received and is not accumulated. This type of connection is supported by Hewlett-Packard JetDirect printers and many others. When DIRECT is specified, GPS opens the data connection upon receipt of the first VTAM request to deliver data. The connection remains open until the VTAM connection is idle for the time specified by MAXIDLE.

PORT=

This parameter designates the remote port where data should be sent. For LPR processing, this value must be 515, the default. For direct processing, this value must be as specified by the network printer manufacturer.

PRinter=

1 to 16 mixed-case characters.

Specifies a 1- to 16-character, mixed-case name of an LPD print queue. This value is sent to the LPD on the remote host to identify the target printer. You must know this name and specify it here. Otherwise, the LPD rejects the attempt to establish a session. This is a required parameter if OUTPUT=LPR is specified or implied. The parameter is ignored for OUTPUT=DIRECT.

Queuing=

A keyword that indicates where GPS is to store data until a complete "report" is accumulated.

Disk

Each report is stored in the storage disk file until it is processed as an LPR data stream. The disadvantage of disk-based queuing is that a sizable amount (100k) of 24-bit storage is required to support the I/O routines and buffers. The advantage is that the disk file is retained in the event of error.

Memory

Each report is stored in 31-bit GETVIS until it is processed as an LP data stream. The advantage is that no 24-bit GETVIS is required. The disadvantage is that, if the TCP/IP partition fails or is shut down before transmission completes, the data is lost. If the LPR transmission fails, the data is written to disk before the daemon shuts down.

STORage=
Specifies the library to be used for staging LPR data and for the optional logging and debug files. The specified file refers to a file defined to the TCP/IP file system. This is a required parameter if QUEUING=DISK, LOG=YES, DEBUG=YES, or OUTPUT=LPR is specified or implied.

TARGet=
Identifies the VTAM application name (APPL) that you want GPS to bind with immediately at startup. This parameter is effective only if you also specify INSESSION=YES. If you specify INSESSION=NO or if you omit this parameter, GPS does not attempt a bind and instead waits for an application to initiate a bind.

TERMname=
Specifies the VTAM application name that GPS uses to connect with VTAM. This name also is the LUNAME that GPS uses to communicate with other VTAM applications. This is a required parameter.

TRANslate=
Specifies the name of the translate table to be used when converting the EBCDIC data stream to ASCII. If not specified, the system default table is used.

This parameter has no meaning if EMULATE=TRANSPARENT is in effect.

TYPe=
Specifies the interface to be used for connecting with the application.

VTAM
The GPS daemon obtains data through a VTAM interface. This is the default and the only interface currently available.

VRC=
Numeric, 0 through 99999.

This value controls the number of times that a GPS daemon will retry any (retriable) VTAM operation before giving up. Acceptable values range from 0 (no retry) through 99,999. The default is 10. This parameter may also be specified as “VTAM_RETRY_Count=”.

VRT=
Used in conjunction with VRC=, this parameter controls how much time elapses (in 300th-second units) before any VTAM operations are retried. Acceptable values range from 0 (no retry) through 9,999,999. The default is 18000 (1 minute). This parameter may also be specified as “VTAM_RETRY_Time=”.

Process Overview

The GPS feature is designed to allow VSE applications to direct print streams to TCP/IP enabled printers. These printers may be supported by a Line Printer daemon (LPD) or may be directly attached to the network (Direct Socket).

To a VSE application, GPS appears to be a series of VTAM-attached printers, each with its own LUname. The printout is sent across the VTAM connection just as if the destination were a physical 3287 printer.

When GPS transmits the printout to the remote printer, it can use the LPD protocol. The LPD protocol requires the remote destination to be a server. Using this technique, GPS must buffer the VTAM stream, format it, and then send it to the LPD as a dataset. Although the VTAM-attached application considers the printout to be going directly to a printer, GPS must arbitrarily carve the stream into a discrete LPD transmission. GPS attempts to segment reports only on page boundaries. In this manner, the final printed version will appear to be one continuous transmission.

Note that the LPD protocol REQUIRES the remote host to completely receive the transmission before printing can begin. In general, this means that the LPD must run on a server with disk storage. Few “network printers” have this capacity. Instead, they simply print the data stream as it is received. The LPD control dataset is ignored and discarded. Unfortunately, this has three drawbacks:

- Because the control file is ignored, requests for multiple copies, banner pages, and other items are also ignored.
- Because the data prints in real time, the connection is tied up for the duration. Such a printer cannot be shared by multiple users (a busy printer appears to be offline to all other requestors).
- The LPD protocol for error recovery states that any batch that is not fully transmitted and acknowledged must be restarted from the beginning. This means that any interruption in printing will result in duplicate pages.

A better technique, when available, is the “direct socket” connection. This is supported by most “network” printers. When using the direct socket interface, no protocol is used. Data received by the printer is sent directly to paper. GPS does no buffering.

Controlling Segmentation

When using the LPD protocol, the print stream must be broken into discrete pieces for transmission. How and when this occurs is controlled by the “MAX” parameters. MAXPAGES controls the number of pages to be included in each report. This is a nominal value. Due to the way data is transmitted to GPS from the application, additional pages may be included. These pages seldom contain much data. Two other “MAX” values, MAXLINES and MAXCHARS, are included to handle reports that do not have embedded page breaks. These parameters ensure that GPS is not flooded by a data stream that it cannot segment.

If a report is broken by either MAXLINES or MAXCHARS, the break will probably be within a page.

Finally, MAXIDLE controls when remaining data should be sent to the printer after the VTAM connection becomes idle.

Segmentation does not occur with a direct socket connection, although MAXIDLE will cause the TCP/IP connection with the printer to be closed.

Example

```
define gpsd,id=gps1,storage=gps.save, ipaddr=rmt1,printer=prt1
GPS900I GPS1 GPS Daemon Starting
FPS917I GPS1 Waiting for BIND
```

Notes

There is no MODIFY GPSD command. To change any specification, you must delete and redefine the daemon.

Related Commands

DELETE GPSD
Terminates a General Print Server daemon.

QUERY GPSDS
Displays the status of the General Print Server daemons.

DEFINE HTTPD

The DEFINE HTTPD command initiates an HTTP (Web) daemon (server). You need only one daemon regardless of the number of Web sessions to be supported.

Syntax

```
DEFine HTTpd ID=id ,ROOT=pubname [,PORT=80]
        [,CONfine={Yes|No}] [,TRANslate=name16]
        [,TIMEOut=5m] [,SECure={Yes|No}]
        [,LIBrary=pubname ,SUBlibrary=name8]
```

Arguments

ID=

This ID uniquely defines this daemon.

ROOT=

Specifies the public name of a library and sublibrary to be used by the HTTP daemon. This library, by default, contains all the HTML, JPGS, and other objects that are served by the HTTP daemon.

PORT=

Specifies the TCP/IP port number to be monitored by the HTTP daemon. The default port is 80. You can specify any value between 0 and 65,535, but you should avoid values below 4096 to prevent collisions with ports that have standard uses.

CONfine=

Controls whether requests are confined to the ROOT= specification.

Yes

The ROOT= specification is *always* prefixed to any client request. If the file is not found, the request fails.

No

The ROOT= specification is prefixed to each client request. If the file is not found, then another attempt is made without the ROOT= prefix. This is the default.

TRANslate=

The HTTP daemon frequently performs EBCDIC-to-ASCII translation. This parameter specifies the name of a translate table (as defined using the DEFINE TRANSLATION command) to use when performing that translation. This table also is used by the daemon when performing ASCII-to-EBCDIC translation.

TIMEOut=

This parameter is effective only if SECURE=YES is specified. This value is the inactivity interval after which the user must resupply a user ID and password. The default is 90,000 (5 minutes). Allowable values range from 0 to 9,999,999.

SECure=

The HTTP daemon can maintain a minimal level of access security based on network address. To do this, the daemon maintains a table of active IP addresses. When a request is received from an address not in the table, the daemon automatically displays a page that requests a user ID and password. These values are checked through the standard TCP/IP FOR VSE mechanisms. If valid, the IP address is added to the table and the original request is transmitted.

The IP address is removed from the table when explicitly requested (a request made for "BLANKING.HTML") or when the inactivity timer (TIMEOUT=) expires.

Yes

The daemon's built-in security procedures are enabled.

No

(Default) Automatic security checking is not done.

LIBrary=

Specifies the library that contains the special security documents PASSWORD.HTML, VIOLATED.HTML, and BLANKING.HTML. If you omit this parameter, the security documents must reside in the root directory.

SUBlibrary=

This parameter, in conjunction with the LIB= parameter, permits you to specify the sublibrary that contains the special security documents PASSWORD.HTML, VIOLATED.HTML, and BLANKING.HTML.

Example

```
define http,id=web,root=prd2.html
HTT900I Daemon Startup HTTP ID:WEB Port:80
```

Notes

The following notes apply to this command:

- There is no MODIFY HTTPD command. To change any of the specifications, you must delete and redefine the HTTP daemon.
- In general, when an HTTP request is received, forward slashes are replaced with periods, the ROOT string is added as a prefix, and, if there is no file or CGI noted in the request, the string ".INDEX.HTML" is added as a suffix. Each node of the resulting name is located in the file system until either an actual file is found or the requested name is exhausted. If a file is found, the unused, extraneous portions of the name are discarded. If no file is found and CONFINE=NO is specified, the entire process is repeated without the ROOT string.

- HTML is not sensitive to line breaks. VSE library files are limited, however, to 80-character lines. If you need to break a line for continuation purposes, end it with an ampersand (&). The HTTP daemon removes the ampersand and appends the next line before transmitting the record.

- Running multiple HTTP daemons on the same port is not recommended. Performance is not enhanced, and there is no way to predict which daemon will be assigned to any given request.

- You can run multiple HTTP daemons on different ports. This is useful when you also specify different libraries or when you require multiple translate tables. This gives you the ability to host multiple web sites.

Related Commands:

ASECURITY WEBL
Sets access security based on the network address.

DEFINE CGI
Loads a CGI program and make it available for use.

DEFINE FILE
Defines a file in the TCP/IP file system and associates it with a file I/O driver.

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

DELETE HTTPD
Terminates a Hypertext Transfer Protocol (web server) daemon.

QUERY CGIS
Displays all currently available CGI programs

QUERY HTTPS
Displays the status of the Hypertext Transfer Protocol (web server) daemons

RELOAD
Reloads a control table.

SECURITY
Controls TCP/IP security functions.

DEFINE LINK

The DEFINE LINK command defines and initializes a network connection device.

Syntax

```
DEFine LINK ID=id ,IPaddr=ip4addr
        ,TYpe={CLAW|LCS|CTCa|OSAX|IPNET}
        [,DEVices=hexaddr] [,SYSid=num] [,MTU=num]
        [,FORCe] [,OUTBuffers=4] [,HOSTName=name8]
        [,HOSTAppl=TCPIP] [,WSName=name8]
        [,WSAppl=name8] [,INFactor=4] [,OUTFactor=4]
        [,RETRY_Time=time] [,STOPPED]
        [,STOPLan={Yes|No}] [,SHUTdown={Yes|No}]
        [,VMReset={Yes|No}]
        [,ALTIP=(ip4addr [,ip4addr,...,ip4addr])]
        [,ROUTER=None|Primary|Secondary}]
        [,OSAPort={0|1] [,DATapath=cuu]
        [,PORTName=name8] [,BUSY=nnn]
```

Arguments

TYpe=

This required parameter indicates the type of link being defined.

CLAW

Channel-connected CLAW interface to an RS/6000, Cisco, or other computer.

LCS

LAN Channel Station controller or any other 3172-compatible communications controllers. These include all forms of the IBM Open Systems Adapter (OSA) except when running in QDIO mode (for that case, use “OSAX”).

Allowable synonyms for LCS are “OSA”, “OSA2”, and “3172”.

OSAX

An IBM OSA Express running in QDIO mode

CTCa

A channel-to-channel adapter connected to another TCP/IP. The other TCP/IP can be running on VSE, MVS, or VM.

IPNET

A cross-partition link to another copy of TCP/IP FOR VSE running on the same VSE image.

DEVices=

Specifies the unit address at which the device resides. When the device requires multiple addresses, specify the lowest address. The system derives the additional addresses by repeatedly adding one until sufficient addresses are computed. For CTCA links, you can specify any two addresses, separated by commas and enclosed in parentheses.

SYSid=

For TYPE=IPNET only. Specifies an ID number identifying another TCP/IP for VSE running in another partition. This number is identical to the value specified for the ID= parameter in the EXEC card PARM field of the other TCP/IP FOR VSE.

IPaddr=

Specifies the TCP/IP network address to be used with this link. Specification allows the TCP/IP host to appear as a different address for purposes of multi-homing. If omitted, the address specified by SET IPADDR= is used. If there are associated DEFINE ADAPTER commands, the IP address should be specified there, and not on the DEFINE LINK.

MTU=

The Maximum Transmission Unit transmitted on this adapter. The maximum value for Ethernet is 1500. Other network types permit values up to 65535. In all cases, the minimum value is 576. The MTU size of individual datagrams may be adjusted by gateways as they pass between physical networks. This adjustment is done by fragmentation. Because fragmentation reduces efficiency, avoid an MTU size mismatch between your networks to the extent practical. You can use parameters on DEFINE ROUTE statements to make MTU adjustments based on path.

FORCe

When specified, directs the link driver to inspect the assigned PUB table entry. If the PUB entry is improperly defined, it is corrected. If FORCE is not specified, no inspection is performed. If the PUB entry is incorrect, the link driver may fail or function erratically. Exercise caution with this parameter. If you code the wrong device address, it is rendered unusable for other purposes until the next IPL.

RETRY_Time=

If link initialization fails, this parameter specifies the time interval before initialization is retried. If this parameter is omitted, the global value specified in the SET LINK_RETRY command is used. Permitted values range from 0 (no retry) to 500 min (30000 sec).

STOPPED
The link driver remains in the "stopped" state until the operator issues a START command. This is particularly useful for links that are not to be used immediately, such as TYPE=IPNET links used to communicate with test partitions running TCP/IP FOR VSE.

STOPLan=
When an LCS (or equivalent) is shut down, a STOPLAN command is issued to the device. If this causes problems with a shared device, or if the device does not support the STOPLAN command, then the STOPLAN can be prevented.

Yes
STOPLAN is issued. This is the default.

No
STOPLAN will not be issued during link shutdown. This setting should be used only if a problem is encountered.

SHUTdown=
When an LCS (or equivalent) is shut down, a SHUTDOWN command is issued to the device. If this causes problems with a shared device, or if the device does not support the SHUTDOWN command, then the SHUTDOWN command can be prevented.

Yes
A SHUTDOWN is issued. This is the default.

No
A SHUTDOWN is not issued during link shutdown. This setting should be used only if a problem is encountered.

VMReset=
This parameter controls whether or not a CP RESET command will be used in conjunction with a CTCA link in a VM environment.

Yes
TCP/IP FOR VSE issues CP RESET commands to virtual channel-to-channel adapters, when appropriate. This is the default.

No
The CP RESET command is not used. Use this setting only when directed by CSI Technical Support.

NOTE: The next seven parameters are for TYPE=CLAW only.

INFactor=
TYPE=CLAW only. This parameter controls the size of the input buffers for the CLAW interface. The value may range from 1 through 8 and represents the buffer size in kilobytes. The default and recommended value is 4. The optimum size for this value depends on the size of the data blocks. Setting too large a size wastes fixed storage. Setting too small a size increases the number of I/Os.

OUTFactor=
TYPE=CLAW only. This parameter controls the size of the output buffers for the CLAW interface. The value may range from 1 through 8 and represents the buffer size in kilobytes. The default and recommended value is 4. Setting too large a size wastes fixed storage.

OUTBuffers=
TYPE=CLAW only. The maximum number of output buffers that are chained together for the CLAW output operation. Specify a value between 1 and 16. The default is 4. Setting too large a size wastes fixed storage. Setting too small a size increase number of I/Os.

HOSTName=
TYPE=CLAW only. The 1- to 8-character name of the host computer system.

This value must match the value expected by the workstation (set during workstation configuration). If the value you code is not acceptable to the workstation, TCP/IP FOR VSE dynamically detects the correct name and retries the connection.

HOSTAppl=
TYPE=CLAW only. The 1- to 8-character name of the application running on the host computer system. This value must match the value expected by the workstation. The recommended value is `TCPIP`.

WSName=
TYPE=CLAW only. The 1- to 8-character name of the workstation. This value must match the value assigned to the workstation during the workstation's configuration. If the value you code is not correct, TCP/IP for VSE will correct it and retry the connection.

WSAppl=
TYPE=CLAW only. The 1- to 8-character name of the application running on the workstation. This value must match the value expected by the workstation. The correct value is probably TCPIP.

NOTE: The next six parameters are for TYPE=OSAX only.

ALTIP=
Valid with TYPE=OSAX only. If your z/VSE system is a "multi-homed" host (known by more than one IP address) or if the TCP/IP FOR VSE stack serves as a gateway to other stacks, you can assign up to nine additional IP addresses to this interface. Consult your OSA Express documentation for more information on the use of this parameter.

BUSY=
Valid with TYPE=OSAX only. It specifies the time to wait before retrying outbound transmissions following a busy indication from the adapter. Valid values range from 100 msec to 5000 msec; the default is 500 msec. During its processing, the OSA Express link driver passes datagrams to the adapter. If the adapter runs out of buffer space, it issues a "busy" return code. The link driver then waits a set period (BUSY= value) and retries the transmission. It does this repeatedly until the busy condition is cleared.

A datagram's "retransmit time" is unaffected by the link driver's delay. The retransmit time for each datagram begins when the adapter has actually accepted the datagram.

ROUTER=
Valid with TYPE=OSAX only. This parameter allows you to capture traffic addressed to "unknown" hosts. TCP/IP FOR VSE then processes it according to the routing parameters in effect. Consult your OSA Express documentation for more information on this parameter.

Primary
This link will be flagged as "primary." All inbound traffic for unknown hosts will be delivered to this link. If GATEWAY ON is in effect, any datagram not addressed to this stack will be passed through the routing table and then be routed accordingly.

Secondary
This link will be flagged as "secondary." All inbound traffic for unknown hosts will be delivered to this link if no other active link has been specified as "primary." If GATEWAY ON is in effect, any datagram not addressed to this stack will be passed through the routing table and then be routed accordingly.

None
This link will only receive datagrams specifically addressed to it. This is the default.

OSAPort=
Valid with TYPE=OSAX only. If your OSA Express supports two ports per CHPID, this parameter can be used to select the port to be used. The default is "0".

Consult your OSA Express documentation for more information on this parameter.

DATapath=
Valid with TYPE=OSAX only. This value indicates the OSA Express CUU address to be used for data transmission.

Consult your OSA Express documentation for more information on this parameter.

PORTName=
Valid with TYPE=OSAX only. This parameter allows you to specify the symbolic name of the OSA Express port to be used with this link. For most OSA Express adapters, this parameter is obsolete and is ignored.

Consult your OSA Express documentation for more information on this parameter.

Example

```
 define link,id=linklcs,type=lcs,dev=(032,33)

 define adapter,linkid=linklcs,number=0,type=ethernet, -
 ip=192.168.1.161,mtu=1500

IPT100I Internet Link Level (ILL) Processor LCS starting
IPL491I OSA link LINKLCS started on devices 0032 – 0033
IPL491I OSA link LINKLCS started adapter 0 as 192.168.1.161
```

Notes

The following notes apply to this command:

- CAUTION: If you delete a hardware link and need to define another connection as the new link, follow these steps:

  1. Delete all defined routes using DELETE ROUTE.
  2. Define the new link using DEFINE LINK.
  3. Define the routes again DEFINE ROUTE.

  If you do not follow this order, after the new link is defined traffic will not be correctly directed to the appropriate connecting hardware.

  As an alternative to these steps, you can cycle the TCP/IP stack.

- Special attention is needed when defining an LCS controller as this device can contain multiple adapters. You must identify these adapters by including at least one DEFINE ADAPTER command.

- All DEFINE LINK commands that require DEFINE ADAPTER commands should be included in the initialization library member. Following initialization, the link driver routines are eligible for immediate startup. This means that if you enter an additional DEFINE LINK command after initialization, the appropriate link driver starts before you can enter any related DEFINE ADAPTER commands. If you must define an LCS-type link following initialization, create a library member containing the necessary commands and use the EXECUTE command to process it.

- Use DISCOVER to find an appropriate MTU size between two hosts.

- The DEFINE ROUTE command can be used to reduce the MTU and MSS sizes for a specific destination or route. For example, you may want to use an MTU of 1500 for "local" devices and a smaller MTU for devices passing through a gateway onto the Internet.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

DISCOVER
Determines the "best" MTU size to a remote host.

EXECUTE
Executes an operator command script.

QUERY STATS
Displays OSAX link driver statistics when the LINKID= option is used.

SET IPADDR
Establishes the default home address for the stack.

SET LINK_RETRY
Determines the default time interval between attempts to reinitialize a failed network link.

DEFINE LOG

Messages are written to the console and one or more log files. Two logs, "CONSOLE" and "SYSLST," are automatically defined and configured for you. You can define additional logs with DEFINE LOG.

Syntax

```
DEFine LOG ID=id ,TYPE={PRinter} ,LOGICALUnit=lunit
        [,LINELength=num]
        [,TIMEstamp={Left|Right|None}] [,routes]
```

Arguments

ID=

An ID to be assigned to this entry. The recommended value is the one used for LOGICALUNIT=.

TYPE=

Indicates the type of log being defined. Currently, "PRINTER" is the only the value supported.

LOGICALUnit=

The logical device to be used, for example, SYS021. This option can be abbreviated as "LU=". By default, SYSLST is used if no logical unit is specified.

LINELength=

Numeric, 40–132.

This value indicates the maximum line length to be written. When a line exceeds this length, it is broken at a blank and a continuation character (>) is appended. This value can range from 40 through 132. If not specified, TCP/IP will select a value based on the device.

TIMEstamp=

A date and time stamp may be applied to each line. This can be on the left of each line, or on the right.

Left

(Default) This log has timestamps on the left of each line.

Right

This log has timestamps on the right of each line.

None

This log does not have timestamps.

routes

Optional values indicate message levels that added to or subtracted from the list of types to be logged. Note that the list is processed from left to right, so specifying "ALL, –NODIAG" ensures that everything but diagnostic messages is written to this log.

By default, all messages are written to “printer” logs, and all messages except diagnose are displayed on the console.

[NO]CRITical
Messages of a critical nature are logged/suppressed.

[NO]VITAL
Messages of vital nature are logged/suppressed.

[NO]WARNing
Warning messages are logged/suppressed.

[NO]IMPORTant
Important messages are logged/suppressed.

[NO]INFOrmational
General informational message are logged/suppressed.

[NO]RESPonse
Responses to commands are logged/suppressed.

[NO]DIAGnose
Messages produced in response to the DIAGNOSE command are logged/suppressed. Note that these messages can be both voluminous and rapid.

[NO]SECurity
Messages related to security are logged/suppressed.

ALL
All message types are logged.

NONE
No messages are logged.

Example

```
 define log,id=sys007,type=printer,lu=sys007,linel=132
 modify log,id=syslst,nodiag
 modify log,id=console,nodiag
 query logs
IPN253I << TCP/IP Console Logs >>
IPT240I CONSOLE on CONSOLE
IPT241I  Line Length: 65 Lines per page: 0 Timestamp: None
IPT242I  Total lines: 0 Total pages: 0
IPT243I  Logging: Critical Vital Warning Important Info Response Security
IPT240I SYSLST on SYSLST
IPT241I  Line Length: 132 Lines per page: 86 Timestamp: Left
IPT242I  Total lines: 300 Total pages: 0
IPT243I  Logging: Critical Vital Warning Important Info Response Security
IPT240I SYS007 on SYS007
IPT241I  Line Length: 132 Lines per page: 0 Timestamp: None
IPT242I  Total lines: 44 Total pages: 0
IPT243I  Logging: All
```

Notes

Make sure that all messages are recorded somewhere. Otherwise, when a problem occurs, it may be difficult to obtain the necessary diagnostic information.

Related Commands

MESSAGE
Controls message suppression.

MODIFY LOG
Changes characteristics of a system log file.

QUERY LOGS
Displays available consoles and logs and their properties.

DEFINE LPD

The DEFINE LPD command initiates a Line Printer daemon (server). LPDs permit remote hosts to send files to VSE/POWER queue entries or VSE files and library members. The remote host must be capable of transmitting the files using the LPR/LPD protocol (this is not FTP). You must provide a definition for each virtual printer to be supported.

Syntax

```
DEFine LPD PRinter=name16, Queue=pubname
        [,LIBrary=pubname1] [,SUBlibrary=name]
        [,TRANslate=name] [,HEXdump={YES|NO}]
        [,USERid=$LPD] [,PASSword=$LPD]
```

Arguments

PRinter=

1- to 16-character printer name, case sensitive. This value must be unique and is the name by which this printer is known to external clients.

Queue=

Specifies the location to receive the output routed to this daemon. This location must be a valid public name from the TCP/IP FOR VSE file system (see DEFINE FILE). See the sections that follow for additional information.

LIBrary=

Specifies a library to be used for temporary storage of a file. If you do not specify a library name, incoming data is temporarily stored in memory. This means that the largest file that can be handled is limited to available memory.

SUBlibrary=

Specifies a sublibrary for temporary storage of a file. This parameter has meaning only if you have specified a value for LIB=. If omitted, TCP/IP FOR VSE uses default sub library TEMP.

TRANslate=

Specifies the name of the table to be used for ASCII-to-EBCDIC translation.

HEXdump=

This is provided as a debugging option.

Yes

Inbound files are converted to hexadecimal dump format. The contents of the control file are included. This can be useful in debugging problems with LPR/LPD.

No

(Default) Files are stored normally.

USERid=
A 1- to 16-character user ID. The default is "$LPD". This value is passed to security processing.

PASSword=
A 1- to 16-character password. The default is "$LPD". This value is passed to security processing.

Example

```
 define lpd,printer=local,queue='power.lst.a'
 define lpd,printer=hex,queue='power.lst.a',hexdump=yes
LPD900I Daemon Startup LPD
```

Notes

The following notes apply to this command:

- A single LPD can handle several simultaneous requests. By default, the listings are buffered in memory until they are complete. When complete, the listing is sent to its final destination. We recommend that you specify a library and sublibrary to be used for temporary storage.
- The sublibrary used for temporary storage of incomplete files should not be used for other purposes and should specify REUSE=IMMEDIATE.

Writing to VSE/POWER

Writing to a VSE/POWER queue is possible only if the special dataset name POWER is assigned a public name using a DEFINE FILE command. The remainder of this discussion assumes that you have followed the recommended procedure of assigning the public name POWER to special dataset name POWER. In the DEFINE LPD command, you must provide a QUEUE specification that consists of public name POWER qualified with the POWER queue and class. For example, you might specify QUEUE=POWER.LST.A.

When a client LPR establishes a link to the daemon, it may pass a jobname. This is defined in the LPR protocol. If jobname meets the syntactic requirements of a POWER job name, it is used in that manner. Otherwise, a job name is constructed as "LPDFA*nnn*", where *nnn* is the three-digit job number transmitted by the LPR client (transmission of a job number is required by the protocol).

Writing to a VSAM KSDS

QUEUE= must specify a fully qualified public name from the TCP/IP FOR VSE file system (no directory structure is available for a KSDS). Any records sent by the LPR client are passed to VSAM as an INSERT. Except for special purposes, this type of operation has little use.

Writing to a VSAM ESDS

QUEUE= must specify a fully qualified public name from the TCP/IP FOR VSE file system (no directory structure is available for an ESDS). Records sent by the LPR client are appended to the end of the dataset.

Writing to a VSE Library

QUEUE= must specify the public name assigned to a VSE library, qualified with a sub library name (for example, if the public name is PRD2 and the desired sub library is LST, then code QUEUE=PRD2.LST). When a client LPR establishes a link to the daemon, it may pass a job name. The job name is truncated to eight characters. The member type is derived from the transmitted origin name (defined in the protocol). If no origin name is available or if it is unsuitable for use as a member type, the word LISTING will be used. If a duplicate member already exists, it is overwritten.

Related Commands

DEFINE FILE
Defines a file in the TCP/IP file system and associates it with a file I/O driver.

DEFINE USER
Creates a user ID and password.

DELETE LPD
Terminates a Line Printer daemon.

QUERY FILES
Displays the contents of the TCP/IP file system.

QUERY LPDS
Displays the status of Line Printer daemons.

DEFINE MASK

The DEFINE MASK command allows you to establish a table of subnet masks. This enables you to use different masks on different networks.

Syntax

DEFine MASK NETwork=*ip4addr*, MASk=*ip4addr*

Arguments

NETwork=

Although this parameter specifies a full IP address, only the network portion is used to identify the network to which the mask is to be applied. If the first byte is in the range of 1 through 127, the network number is one byte. If the first byte is in the range of 128 through 191, the network number is the first two bytes. If the first byte is in the range of 192 through 223, the network number is the first three bytes. In all cases, the host portion of the address is ignored.

MASk=

This is the value of the mask that should be applied to the network address to obtain the subnetwork number. This value is coded in the same manner as a TCP/IP network address *n.n.n.n*, where each instance of *n* is the decimal representation of one byte.

Example

```
 define mask network=192.168.30.4, mask=255.255.255.0

 query masks
IPN253I << TCP/IP Network Masks >>
IPN575I  Network: 192.168.30.0 (12,625,950) Mask: 255.255.255.0
IPN575I  Network: 127.0.0.0 (127) Mask: 255.0.0.0
IPN575I  Network: Default (--) Mask: 255.255.255.0
```

Notes

The following notes apply to this command:

- Only one subnet mask may be specified for any given network.
- The DEFINE MASK command does not return a response. You can issue QUERY MASKS to determine all masks in effect at the present time.
- See the discussion of IP addresses, masks, and subnetworking in the *TCP/IP FOR VSE Installation Guide*.
- If a mask is not defined for a particular network, the default mask (SET MASK=) is used.

Related Commands

DEFINE ROUTE

Adds an entry to the TCP/IP routing table.

DELETE MASK
Deletes a subnet mask for a particular network.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

DEFINE MENU

The DEFINE MENU command enables you to load a telnet menu from a library and make it available for use by telnet daemons.

Syntax

DEFine MENu ID=*id*, MEMber=*member*

Argument

ID=

The name to be assigned to the menu. This name must be unique and is referenced by the DEFINE TELNETD command.

MEMber=

The member name containing the menu definition. The specified value is appended with a ".L" suffix to form the complete name.

Example

```
 define menu,id=menuadm,member=menu1

 query menus
IPN253I << TCP/IP Menu Definitions >>
IPN391I   ID: MENUUSR   Member: MENU2
IPN391I   ID: MENUADM   Member: MENU1
```

Notes

The following notes apply to this command:

- Directions for creating a telnet menu are contained in the *TCP/IP for VSE Installation Guide*.
- The DEFINE MENU command does not return a response. You can issue the QUERY MENUS command to currently available menus.
- Menus are checked for errors and compiled during execution of this command. Any errors that are flagged suppress the define operation.
- Telnet daemons check for the existence of a menu only when a connection is attempted.

Related Commands

DEFINE TELNETD

Creates a TN3270 or TN3270E daemon.

DELETE MENU

Removes a TN3270 menu file from memory.

QUERY MENUS

Displays menus available for TN3270 use.

QUERY TELNETDS

Displays TN3270 and TN3270E daemons.

DEFINE NAME

The DEFINE NAME command allows you to create symbolic names. These names can be used by TCP/IP FOR VSE clients as a convenient way of referring to other items.

Syntax

```
DEFine NAME NAMe=name64
        {[,IPaddr=ip4addr]|[,SCRipt=member]}
```

Arguments

NAMe=
The 1- to 64-character, case-insensitive, symbolic name to be defined.

IPaddr=
An IPv4 network address. If this parameter is coded, references to the specified name will be mapped to this address.

SCRipt=
If this parameter is specified, the command creates a symbolic name that can be used to refer to a "script." Once defined, the script can be used with any client that uses domain name resolution, including Email, LPR, and Ping. The member name is suffixed with ".L" and is located using the standard search chain in the TCP/IP FOR VSE partition.

Example

```
define name,name=printer17,ipaddr=201.52.101.72
```

Notes

Names referring to IP addresses override DNS lookup.

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

DELETE NAME
Removes a TCP/IP FOR VSE symbolic name created by DEFINE NAME.

QUERY NAMES
Displays the contents of the symbolic names table created by DEFINE NAME.

DEFINE NTPD

The DEFINE NTPD command starts a Network Time Protocol daemon. This service allows other hosts on the network to synchronize with the VSE TOD clock.

Syntax

DEFine NTPd ID=*id* [,PORt=37] [,PRotocol={Udp|Tcp}]
[,GMT=*snum*] [,ADJustment=*snum*]

Arguments

ID=
A unique name that identifies the daemon.

PORt=
The port number to be monitored for time requests. The standard and default port is 37.

PRotocol=
A keyword, one of the following:

Udp
This daemon replies to requests using the UDP protocol. This is the default.

Tcp
This daemon replies to requests using the TCP protocol.

GMT=
Signed numeric, -99999 through +99999.

This is the number of hours to be added/subtracted to the VSE TOD clock setting before transmitting to a client. This is useful if your local clock does not have the proper time-zone offset or if daylight savings time is involved.

ADJustment=
Signed numeric, -99999 through +99999.

This is a signed integer number of seconds to be added/subtracted to the VSE TOD clock setting before transmitting to a client. Due to varying calculation of leap seconds and so on, it may be necessary to provide a fine adjustment. The best way to determine this value is to start an NTP daemon, have a client obtain the time, and determine the adjustment required, if any.

Example

```
define ntpd,id=ntp1
NTP100I NTP Daemon Running
```

Notes

The following notes apply to this command:

- The Network Time Protocol simply consists of transmitting the current time as the elapsed number of seconds since January 1, 1900 (the last year of the nineteenth century).
- In addition to leap days added each four years (2000 is a leap year), leap seconds have also been added at intervals since 1972.
- NTPD includes leap seconds in its calculations.
- Each remote host requires client software to ask for the time and set the local clock. One such client is Tardis. There is a shareware version that is available from http://www.kaska.demon.co.uk.
- NTP clients attempt to allow for network transmission delay. The accuracy of clocks maintained with NTP, however, will probably be only within +/- 2 seconds.

Related Commands

DELETE NTPD
Terminates a Network Time Server daemon.

QUERY NTPDS
Displays the status of NTP daemons.

DEFINE PUBLISHER

The DEFINE PUBLISHER command creates a Publisher daemon that can be used to notify external processes of completed events in the TCP/IP FOR VSE partition.

Syntax

DEFine PUBlisher ID=*id* ,IMODlist=member

Arguments

ID=

A unique name that will identify this daemon.

IMODlist=

Identifies an "L book" that will be inspected to determine a list of publishing event numbers and the corresponding IMOD to be executed under the control of CSI International's FAQS/ASO. The format of this file is explained in the "Exposition" section below.

Example

```
define publisher,id=pub01,imodlist=list01
IPN694I Publishing Daemon now active
```

Exposition

The Publisher is intended to notify external applications and processes of TCP/IP-related events, such as the successful transfer of a file. The Publisher issues a defined-format notification message that can easily be interpreted by other programs. When used with CSI-FAQS/ASO, IMODs can be scheduled to automate dependent actions. This facility is intended to eliminate the need to "screen-scrape" the operator console to determine event completion and to provide an unchanging format for automation.

In CSI International's production environment, we use the Publisher's event 1 (FTP'ed file received) to run an IMOD that issues a "refresh" command so that CSI International's Entrée will purge the "old" copy from its cache. The same event also initiates another FTP to copy the updated file to our hot-site backup.

The file specified by the IMODLIST= parameter will contain a mapping of "published event numbers" to the IMOD that is to be triggered. Four fields are coded, separated by one or more blanks:

1. The Published Event Number (see the example below)
2. The keyword "IMOD"
3. The IMOD name
4. Optional comments

Any line that begins with "*" is considered a comment.

Example

The following job catalogs a sample IMOD file.

```
// EXEC LIBR
ACC SUBL=PRD2.CONFIG
CATALOG IMODLIST.L  REPLACE=YES
* * This is the IMODLIST= member for the DEFINE PUBLISH,IMODLIST=
* * It is read by the Publishing Daemon to map Published Events
* * to specific IMODs.
* *
* * The format is:
* * event_num IMOD imodname comments
* * Fields are separated by 1 or more blanks
* *
001  IMOD $FTPFRCV EVENT-ITEM 1 = FTP FILE RECEIVED
002  IMOD $FTPFSNT EVENT-ITEM 2 = FTP FILE SENT
/+
/*
```

Notes

Additional information on using the Publisher can be found in the *TCP/IP FOR VSE Installation Guide*.

Related Commands

DELETE PUBLISHER
Terminates a Publisher daemon.

QUERY PUBLISHER
Displays the status of the Publisher daemon.

DEFINE ROUTE

The DEFINE ROUTE command is used to construct a routing table for the purpose of routing datagrams and to provide transmission characteristics based on path.

Syntax

```
DEFine ROUTe ID=id ,LINKid=name16 [,ACTion={Next|Stop}]
        [,ADAPter=0] ,IPaddr=ip4addr
        [,GATEway=ip4addr1] [,AFTer=id] [,MTU=num]
        [,MSS=num] [,CRETran=msec] [,DRETran=msec]
        [,FIXRetran={Yes|No}] [,MINRetran=msec]
        [,MAXRetran=msec] [,PULse=sec] [,WINdow=num]
        [,RPAuse=msec] [,RETRY=num]
```

Arguments

ID=

This is the character value that will be used when issuing a QUERY or a DELETE against *this* route.There is no default.

LINKid=

This is the same value that is specified in the ID= parameter of the DEFINE LINK command that will be the destination for this route table entry. There is no default.

ACTion=

Next

Indicates that the route compare that takes place will continue to the next route on a no-match condition.This is the default.

Stop

The route compare will stop at this point and not continue further when a match or a no-match condition is encountered.

ADAPter=

For links with adapters, this directs the route to the specific numbered adapter. The default is “0”. This parameter is required if the NUMBER= parameter of the target DEFINE ADAPTER is not “0”.

IPaddr=

A TCP/IP network address or “zero host” address. The default is 255.255.255.255. All messages destined for this address are sent on the associated link.

GATEway=

The full network address of a gateway to other networks. The default is 255.255.255.255. A match on this table entry causes the data packet to be sent to the specified gateway.

AFTer=

The value of the name parameter identifying the DEFINE ROUTE statement after which this one is to be inserted. If this parameter is omitted, the route entry is added to the end of the table. A special value of "TOP" can be coded to cause the route statement to be inserted at the top of the list.

Placement in the table is very important because the look-up procedure is a top-to-bottom search for the first match (except for "0.0.0.0," which is always matched last).

MTU=

The MTU value to be used with this route. There is no default. This is only meaningful if it is less than the value specified by the target DEFINE LINK or DEFINE ADAPTER. This parameter only controls the size of outbound datagrams. However, it also determines the largest value that can be used for maximum segment size (MSS).

MSS=

The Maximum Segment Size to be used with this route. The MSS value is sent to the remote host during negotiation of a TCP connection. It specifies the largest piece of data that may be sent in a single datagram. When selecting an MSS size, you must take into account that there are 40 bytes of header information beyond the data portion. Thus, for an MTU size of 1500, the maximum MSS is MTU-40, or 1460. When the MSS for a connection is determined, it is always reduced to MTU-40. If no value is specified for MSS, then it is set to either the default MSS or 40 less than the MTU, whichever is less. The MTU is the largest value that can be used without datagram fragmentation. The SET MAX_SEGMENT command sets the default MSS (32684 is the system default).

CRETran=

This specifies the number of milliseconds that TCP/IP will wait for an ACK in response to a connection request (SYN). Once this interval has elapsed, retransmission mode is entered. The default is 1000. The SET RETRANSMIT command sets this default value in $1/300^{th}$ seconds (300 is the system default for this command).

DRETran=

This specifies the number of milliseconds that TCP/IP will wait for an ACK in response to a datagram transmission on an established connection. Once this interval has elapsed, retransmission mode is entered. The default is 1000. The SET RETRANSMIT command sets this default value in $1/300^{th}$ seconds (300 is the system default for this command).

FIXRetran=

Yes

The values specified for DRETRAN= and RPAUSE= will remain constant for the duration of the connection.

No

The values for DRETRAN= and RPAUSE= will start out as specified but will be dynamically adjusted as network response is analyzed.

The system default is NO. The SET FIXED_RETRANSMIT command updates the default setting for this parameter.

MINRetran=

If FIXRETRAN=NO is specified, this is the minimum time (in milliseconds) that can be dynamically assigned to DRETRAN. The default is 500.

MAXRetran=

If FIXRETRAN=NO is specified, this is the maximum time (in milliseconds) that can be dynamically assigned to DRETRAN. The default is 2000.

RPAuse=

Once retransmit mode has been entered, this is the time (in milliseconds) that will elapse between retransmission attempts. The default is 500.

RETRY=

This parameter specifies the number of times that an unacknowledged datagram will be retransmitted before the connection is considered to be dead. The default is 50.

PULse=

This parameter specifies how long (in $1/300^{th}$ seconds) that a connection can be idle—no traffic of any kind—before a probe is made to determine whether the remote host is still active. The SET PULSE_TIME command sets the default for this parameter (the system default is 18000, which equals 60 seconds).

WINdow=

This value indicates the desired size of the Receive Window. The SET WINDOW command sets the default for this parameter (the system default is 65535).

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
DEFINE ROUTE,ID=LOCAL, LINKID=LINK3172, IPADDR=192.168.001.000, -
 ADAPTER=0,MSS=5000,MTU=5040

IPN448I  ID: LOCAL    Link ID: LINK3172
IPN449I     IP Address: 192.168.1.0 Mask: 255.255.255.0
IPN450I     Net: 192.168.1.0 Subnet: -- Host: --
IPN875I     MTU: 5040 Max Seg: 5000 Pulse: 60s
IPN876I     SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
IPN877I     Retran Min: 500ms Max: 2000ms
IPN882I     Retry Delay: 500ms  Retries 50
IPN884I     RWin: 65535

DEFINE ROUTE,ID=DEFAULT, LINKID=LINK3172, IPADDR=0.0.0.0, -
 ADAPTER = 0, GATEWAY=192.168.001.1, MSS=1400

IPN448I  ID: DEFAULT    Link ID: LINK3172
IPN449I     IP Address: 0.0.0.0 Mask: 255.255.255.0
IPN450I     Net: -- Subnet: -- Host: --
IPN537I     Gateway IP Address: 192.168.1.1
IPN875I     MTU: 0 Max Seg: 1400 Pulse: 60s
IPN876I     SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
IPN877I     Retran Min: 500ms Max: 2000ms
IPN882I     Retry Delay: 500ms  Retries 50
IPN884I     RWin: 65535
```

The table below shows the DEFINE ROUTE parameter name for some displayed parameter values.

Label	Parameter Name
`Max Seg:`	MSS
`SYN Retran:`	CRETran
`Data Retran:`	DRETran
`Fixed:`	FIXRetran
`Retran Min:`	MINRetran
`Max:`	MAXRetran
`Retry Delay:`	RPAuse
`Retries:`	RETRY
`RWin:`	WINdow

Notes

The following notes apply to this command:

- TCP/IP FOR VSE searches the route statements in the same order that they are entered (except for an entry with an all zero IP address). Using the AFTER= parameter ensures proper sequencing of the route table.
- To examine the order of search, use the QUERY ROUTES command. This displays the route table in search order (except for entries with an all-zero IP address).
- A route statement with an all-zero IP address is matched only after all other entries have been tested (in order).
- Once a route statement is matched by IP address, the designated link is checked for availability. If the link cannot be used, the search continues with the next route entry.
- When you issue a DELETE LINK command because
  - — you want to disable a link, or
  - — you want to change one or more attributes of a link, or
  - — you want to recover from a link that was deactivated by a hardware command,

  then all DEFINE ROUTE statements referring to that link are automatically deleted.
- If you want to redefine a link you deleted, you first need to wait for the DELETE command to complete and then confirm that the LINK no longer exists by issuing a QUERY LINKS command. After that, you need to define the link and then reestablish the routes using the DEFINE ROUTE command.

  For example, after issuing a DELETE LINK, you can

  1. Use QUERY LINKS to confirm that the link is gone
  2. Use DEFINE LINK to reestablish the link
  3. Use DEFINE ROUTE to reestablish the routes.

  Alternatively, to reestablish a link you deleted, you can simply cycle the TCPIP FOR VSE stack. The initialization definitions will be used.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ALTIP
Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE MASK
Creates a subnet mask for a particular network.

DELETE ALTIP
Removes an alternate home address.

DELETE LINK
Removes a link between TCP/IP and a network or a directly connected stack.

DELETE ROUTE
Removes an entry from the network routing table.

DISCOVER
Determines the “best” MTU size to a remote host.

GATEWAY
Controls forwarding of datagrams not intended for the VSE stack.

MODIFY ROUTE
Changes values on an existing entry in the Route Table.

QUERY ARPS
Displays the current content of the ARP table.

QUERY LINKS
Displays the status of network links.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

QUERY OPTIONS
Displays the current values of modifiable parameters.

SET IPADDR
Establishes the default home address for the stack.

SET MASK
Establishes a default subnet mask.

SET FIXED_RETRANSMIT
Sets the default for the FIXRetran= parameter on DEFINE ROUTE.

SET MAX_SEGMENT
Controls the default setting for the inbound Maximum Segment Size (MSS= on DEFINE ROUTE).

SET PULSE_TIME
Controls the default setting for the interval between probes of inactive connections (PULSE= on DEFINE ROUTE).

SET RETRANSMIT
Controls the length of the default interval before TCP/IP FOR VSE enters retransmit mode for unacknowledged data packets (CRETRANS= and DRETRANS= on DEFINE ROUTE).

SET WINDOW
Controls the default setting of the TCP inbound window (WINDOW= on DEFINE ROUTE).

TRACERT
Displays each “hop” in a route along with the time required to reach it.

DEFINE SOTRACE

The DEFINE SOTRACE command starts a tracing operation for Socket requests. Each Socket request is saved in memory until a corresponding DELETE TRACE command is issued. The DUMP command may be used at any time to output the accumulated trace buffers.

Syntax

```
DEFine SOTRACe ID=id [,IPaddr=ip4addr] [,PORT=port]
        [,SCOPE={All|Internal|External|Obsolete}]
        [,MAXData=60] [,PHASE=member] [,SIZe=500]
        [,KIND={TCP|UDP|FTP|CLIENT|TELNET}]
```

Arguments

ID=

A unique name that identifies this trace entry. This name is used in conjunction with the DELETE TRACE and QUERY TRACES commands.

IPaddr=

Limits tracing to a specific IP address. Only traffic involving the specified address is considered.

PORT=

Limits tracing to a specific port. Only traffic involving the specified port is considered.

PHASE=

Limits tracing to a specific phase. Only traffic involving the specified phase is considered. The phase name of a connection can be found with a QUERY CONNECTIONS command.

SCOPE=

Internal

Only socket requests made from within the stack partition are considered.

External

Only socket requests made from outside the stack partition are considered.

Obsolete

Only socket requests with obsolete parameter lists are considered. This may be useful when attempting to locate applications that may need to be recompiled.

All

(Default) All Socket requests are considered.

MAXData

The amount of data from each socket call that will be included in the trace. The default is 60 bytes per socket request. Acceptable values range from 0 to 64K.

SIZe=
Indicates the maximum message blocks that are to be retained during the trace. Old blocks are discarded in favor of newer blocks. The default is 500. Allowable values are 10 through 64K.

KIND=
This parameter can be used to limit the trace to specific types of Socket requests. By default, TCP requests are recorded.

TCP
Trace standard TCP requests. These are the requests most frequently associated with applications using TCP/IP for communication. This is the default.

UDP
Trace UDP requests. UDP requests are generally used when performance is of the essence. Under UDP, all acknowledgement, retransmission, and content verification are handled by the application.

FTP
Trace only socket requests that specify a type of FTP.

CLIENT
Trace socket requests associated with the TCP/IP FOR VSE client interface. This includes socket requests with types of PING and LPR.

TELNET
Trace only socket requests that specify a type of telnet.

Example

```
 define sotrace,id=trace1,phase=testserv
IPN210I Socket Trace ID TRACE1 defined and running.
```

Notes

The following notes apply to this command:

- Once trace data has been accumulated, the DUMP command can be used to dump the data.
- Be sure to terminate your trace with DELETE TRACE after you use the DUMP command. Because all traffic to a traced address is retained in memory, large amounts of virtual storage can be tied up.
- You can trace multiple addresses simultaneously by issuing multiple DEFINE TRACE commands. Each trace must have a unique ID.

Related Commands

DEFINE TRACE
Starts a datagram trace.

DELETE TRACE
Terminates a trace and frees its storage.

DUMP
Performs a formatted dump of various TCP/IP control blocks.

QUERY TRACES
Displays a list of currently running traces.

SEGMENT
Segments the SYSLST and log files, making them available for printing.

DEFINE TELNETD

The DEFINE TELNETD command defines and initializes one or more Telnet daemons (servers). Each daemon provides one TN3270 or TN3270E session.

Syntax

```
DEFine TELnetd ID=id ,TARget=name8 ,TERMname=luname
        [,COunt=num] [,BASE=1]
        [,TN3270E={Listener|Effector}] [,PORT=23]
        [,MENu=name16] [,LOGMode=S3270]
        [,LOGMODE3=D4B32783] [,LOGMODE4=D4B32784]
        [,LOGMODE5=D4B32785] [,IPaddr=0.0.0.0]
        [,BUFfersize=num] [,CLOSE={Always|Seldom}]
        [,GENeric=name16] [,GRoup=name16]
        [,POOL={Yes|No}] [,DRIVer=TELNETD]
```

Arguments

BASE=

When used in conjunction with the COUNT= parameter, BASE= specifies the first two-digit suffix to assign. The default is 1. Allowable values are 0 through 99.

BUFfersize=

When POOL=NO is in effect, each daemon allocates its own static buffers.

When appropriate, you may use the BUFFERSIZE= parameter to control the size of each buffer. If not specified, then the value established by the SET TELNETD_BUFSIZE= command is used. Valid values range from 8,192 through 65,535.

Reducing the default value may result in session failure or data loss. Increasing the default value should be done only in cases where very large and complex screens are incorrectly displayed.

CLose=

This parameter controls the method by which the Telnet daemon ends its session with the application. The following values can be coded:

Always

The Telnet daemon closes and re-opens its VTAM ACB each time a session is completed. This is the default.

Seldom

The Telnet daemon closes and re-opens its VTAM ACB only if the previous session ended with an error.

COunt=

Specifying any value for COUNT= causes one to 100 Telnet daemons to be defined using the other DEFINE TELNETD parameters as a pattern. A two-digit suffix, 00 to 99, is added to the ID= and TERMNAME= values. The starting value for this two-digit suffix is specified using the BASE= parameter. The default is to define a single daemon without the use of numeric suffixes.

DRIVer=

The name of the phase that will be loaded to provide Telnet daemon support. The default to TELNETD. Specifying any other value may void your warranty.

ID=

This is a unique identifier assigned to the Telnet daemon. This identifier can be used in later commands to single out a specific daemon. If COUNT= is also specified, then a two-digit suffix is added to each ID= value.

IPaddr=

If specified, this daemon only connects with requests issued from a remote host whose IP address matches the supplied pattern. If a full IP address is specified, then the match must be exact. If the address specifies network and subnetwork (host portion of the address is zero), then a generic match is performed.

Selection of an available daemon can be either first fit or best fit. This is controlled by the CONNECT_SEQUENCE command.

LOGMode=

This is the VTAM logmode that will be used to negotiate VTAM sessions for a model 2 terminal. This logmode must be non-SNA. The default and recommended value is S3270.

LOGMODE*n*=

Where "*n*" is 3 through 5 to provide the ability to specify logmodes for 3270 models 3 through 5. The logmode must be non-SNA. The default and recommended values are: D4B32783, D4B32784, and D4B32785, respectively.

GENeric=

A 1- to 16-character name; not case sensitive.

This parameter has meaning only when accompanied by TYPE=LISTENER.

This parameter has effect when a TN3270E client requests a session but does NOT specify an LUname preference. In this case, the first available effector daemon whose GROUP name matches the listener daemon's GENERIC name is assigned. If GENERIC is NOT specified, then all effector daemons are eligible.

The default is a null value. If you do not want "generic" sessions, specify a value that does not match any GROUP names.

GRoup=

A 1- to 16-character name. Not case sensitive.

This parameter has meaning only when accompanied by TN3270E=LISTENER or TN3270E=EFFECTOR.

When specified for a listener daemon, the user's request for an LUname will be honored only if the required effector daemon has the same GROUP name.

The default for this value is null. If no value is specified on a Listener daemon, there is no group restriction on which effector daemons can be used.

MENu=

1- to 16-character name.

This refers to a menu previously defined by a DEFINE MENU command. If specified, the menu contents are displayed to the user upon initial connection with the daemon.

If specified, this value overrides any specification by TARGET= (except that the TARGET= value is passed to the menu as the default application).

Following termination of a Telnet session, the connection with the remote client is always closed. The daemon does NOT return to the menu until a new connection is requested.

POOL=

This parameter determines the location of the daemon's data buffers.

Yes

This Telnet daemon shares a pool of buffers with other daemons specifying POOL=YES. A buffer is required to pass data through the daemon in either direction. The number of available buffers is controlled by the SET TELNETD_BUFFERS command. Buffer usage can be displayed with the QUERY STATS command.

No

This is the default. When NO is in effect, each daemon acquires its own buffers (31-bit). Although this can result in considerable storage use, less CPU is required to process the data. The BUFFERSIZE= parameter can be used to control the size of individual buffers.

PORT=

This is the "well-known" port remote clients will use to connect with a waiting Telnet daemon. The default (and standard) port is 23. One reason for selecting other ports is to create pools of Telnet daemons.

TARget=

This is the VTAM Application name of the application with which this daemon will connect. If MENU= is also supplied, then TARGET= is ignored. Either TARGET= or MENU= is required for all TN3270 daemons and TN3270E Effector daemons.

TERMname=

This is the VTAM Application name (ACB) that the Telnet daemon will use to connect with VTAM. Put another way, this value will be the LUname of the Telnet session. If COUNT= is also specified, then a two-digit suffix will be appended to the TERMNAME= value. TERMNAME= is required for all TN3270 daemons and TN3270E Effector daemons.

TN3270e=

TN3270E is a newer version of the TN3270 protocol that allows for more “advanced” support for 3270-type devices. One feature of TN3270E is client specification of LUname. Note that this applies only to 3270 and not printer sessions, which are handled by the General Print Server (GPS) optional feature.

To provide TN3270E support, you must define two types of daemons: listeners and effectors. A listener daemon opens a “listen” connection and waits for a client to request a session. Once a client is connected, the listener negotiates a TN3270E session (including LUname) and then passes the session to the appropriate effector daemon. The effector daemon owns the LUname and conducts the actual session. Once a session is negotiated, the listener daemon opens another “listen” connection and awaits further connection requests.

A frequent reason for using TN3270E is “security”. The idea is to assign a specific LUname to a particular user. Unfortunately, this scheme relies heavily on the “honor system.”

To provide some control over LUname assignment, you can assign an IP address (or subnet or network) to a listener daemon. Once an address (or pattern) is assigned, only those clients whose IP address matches that of the listener can obtain a connection to that particular listener.

Listener daemons can be limited in their choice of effector daemons. If the listener has a value specified for GROUP=, it can only pass sessions to effectors with an identical GROUP= specification.

Another listener daemon parameter, GENERIC=, provides for situations where the client does NOT specify an LUname preference. In this case, a random effector daemon is selected from the GROUP identified by the listener’s GENERIC= parameter.

Specify one of the following values to define a TN3270E daemon:

`Listener`
This daemon listens for a Telnet client to request a session and then negotiate a TN3270E session. Following successful negotiation, the session is passed to the appropriate effector daemon and the listener daemon returns to the listen mode.

`Effector`
This daemon has no capability to initiate a Telnet session. It accepts a pre-negotiated session from a Listener daemon and then continues the TN3270E session between the client and VTAM application using its predefined LUname.

Example

```
 define telnetd,id=tln,target=dbdcics,termname=tcp, base=0, count=5
TEL900I Daemon Startup Telnet Termname: TCP00 Port: 23
TEL900I Daemon Startup Telnet Termname: TCP01 Port: 23
TEL900I Daemon Startup Telnet Termname: TCP02 Port: 23
TEL900I Daemon Startup Telnet Termname: TCP03 Port: 23
TEL900I Daemon Startup Telnet Termname: TCP04 Port: 23
```

Notes

The following notes apply to this command:

- One Telnet daemon is required for each concurrent Telnet terminal session requested by a remote user. Outbound Telnet sessions, such as those initiated from a CICS transaction to another platform, use daemons provided at the remote end. Do not consider outbound sessions when determining the number of daemons you need.
- If Telnet daemons refuse to start or connect, check your VTAM storage and buffer values and your VSE/ESA dataspace values. Remember that the JCL-specified DSPACE parameter sets a maximum value. The actual amount of DSPACE available for all requests is set at IPL time.
- When setting up VTAM APPL IDs for telnet terminals, be sure to specify EAS=1. This reduces the amount of storage required to support each session.
- The TN3270 protocols are EBCDIC based. No ASCII translation is performed.

Controlling LUnames

When a TN3270 client opens a connection with one of TCP/IP's TN3270 daemons, the connection is established and the daemon is assigned prior to any negotiations with the client. Because each daemon has a preassigned LUname, each telnet session obtains its LUname in a random fashion.

If you must control LUname assignment, you can use one of the following methods:

- You can define multiple pools of telnet daemons. Each pool monitors a different port number. The downside is that the end user must know and specify the appropriate port when connecting.
- You can specify an IP address when you define an individual telnet daemon. Depending on how the CONNECT_SEQUENCE command is set, you can precisely control the assignment of telnet daemons based on the IP address of the requester.
- Do not mix TN3270 and TN3270E daemons using the same port number. You can use both daemon types if you assign different port numbers.

Related Commands

CONNECT_SEQUENCE
Controls whether connection requests are allocated by IP address pattern checking.

DEFINE MENU
Loads a menu file and makes it available for use by Telnet daemons.

DELETE TELNETD
Terminates a TN3270 or TN3270E daemon

DIAGNOSE
Controls diagnostic display options.

FLUSH
Terminates all processing with a specific remote host.

QUERY ACTIVE
Displays the status of active daemons.

QUERY STATISTICS
Displays a summary of stack-related information.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

SET PULSE_TIME
Controls the default setting for the interval between probes of inactive connections (PULSE= parameter on DEFINE ROUTE).

SET TELNETD_BUFFERS
Determines the size of the buffer pool shared by TN3270 daemons.

SET TELNETD_BUFSIZE
Determines the size of individual TN3270 buffers.

DEFINE TLSD

The DEFINE TLSD command initiates an SSL/TLS daemon to handle encryption and decryption.

Syntax

```
DEFine TLSd ID=id ,CERTLibrary=name8 ,CERTMember=name8
        ,CERTSublibrary=name8 [,CIPher=09]
        [,MINVers={0300|0301|0302|0303}] [,PORT=443]
        [,PASSport=80] [,TYPE={1|2}]
```

Arguments

ID=
A unique name to identify this daemon.

CERTLibrary=
Identifies the library name that contains the private key and certificates to be used by this daemon.

CERTMember=
Identifies the member name that contains the private key and certificates to be used by this daemon.

CERTSublibrary=
Identifies the sublibrary that contains the private key and certificates to be used by this daemon.

CIPher=
A string of hexadecimal values that indicate the acceptable cipher suites. When a connection is being negotiated, the client will be required to select a cipher suite from this list. The valid values for the **SSL30, TLS 1.0,** and **TLS 1.1** protocol versions are as follows:

01 — RSA_NULL_MD5

02 — RSA_NULL_SHA

08 — RSA_SDES040_SHA

09 — RSA_SDES056_SHA

0A — RSA_TDES168_SHA

2F — RSA_AES128_SHA

35 — RSA_AES256_SHA

When the **TLS 1.2** protocol version is used, the supported cipher suites are as follows. These suites do not support the DES algorithm.

2F — RSA_AES128CBC_SHA160

35 — RSA_AES256CBC_SHA160

3C — RSA_AES128CBC_SHA256

3D — RSA_AES256CBC_SHA256

MINVers=

This value specifies the minimum acceptable version of SSL/TLS.

0300

This is the "old" SSL standard (SSL30) and is less secure. Most clients will be able to provide this level of support.

0301

This value requires that clients adhere to the TLS 1.0 standard as set forth in IETF RFC2246. This version is more secure, and many clients support it.

0302

This is the TLS 1.1 version. It provides enhanced security.

0303

This is the TLS 1.2 version. It provides even more security, but not all clients can support it.

PASSport=

This is the unique port number that the SSL/TLS-enabled application will connect with.

If this value is identical to the PORT= value, then this indicates the application has directly implemented the SSL/TLS API.

PORT=

This is the unique port number that clients will connect with. Any port number, 1 through 65,535, may be specified.

TYPE=

One of two numeric values may be specified to indicate whether the client must provide authentication when connecting.

1

(Default) No client authentication is performed.

2

Client authentication is enforced.

Example

```
 define tlsd,id=tls01,port=992,passport=992,cipher=08090a2f35 -
   certlib=prodlib,certsublib=phase,certmember=sample01
 query tlsd
IPN253I << TCP/IP TLS Daemons >>
IPN617I  ID: TLS01 Cipher: 08090A2F35
IPN618I  Port: 992  Passport: 992 Type: Server
IPN619I  Driver: SSLD Minimum version: 0300
 define telnetd,id=teln01,tcpappl=telnlu01,menu=menu01,pool=yes,port=992
TEL900I Daemon Startup Telnet Termname: TELNLU01 Port: 992
 query telnet
IPN253I << TCP/IP Telnet Daemons >>
TEL920I  ID: TELN01 (Inactive)
TEL921I  Terminal: TELNLU01   Menu: MENU01
TEL922I  Port: 992 Match IP: 0.0.0.0
```

Notes

The following notes apply to this command:

- SSL/TLS servers must always provide a certificate to the client during negotiation. The client then uses the certificate to authenticate the server.
- Given the library, sublibrary, and member name specified, three members with the extensions of ".PRVK", ".ROOT", and ".CERT" must exist and contain valid information.
- Consult the *TCP/IP FOR VSE Optional Features Guide* for more information on configuring TCP/IP FOR VSE's SSL/TLS for VSE feature.
- Cipher suites 0A and 2F are sufficient for most applications. Note that suite 0A is a DES-based cipher that is not supported in TLS 1.2.
- Cipher suite 35 provides the strongest encryption for more sensitive applications for the SSL30, TLS 1.0, and TLS 1.1 protocol versions. Cipher suite 3D provides the strongest encryption for the TLS 1.2 protocol version.
- TLS 1.2 support also requires IBM's hardware CP Assist for Cryptographic Function (CPACF) feature.
- Cipher suites 01 and 02 provide no encryption and generally should not be included in the list.
- Coding CIPHER=08090A2F35 (concatenated values) provides the greatest flexibility for establishing an encrypted connection for the SSL30, TLS 1.0, and TLS 1.1 protocol versions. For the TLS 1.2 protocol version, coding CIPHER=2F353C3D provides the greatest flexibility.

- Using an encrypted connection protects the data ONLY during transmission. In most instances where data is stolen or forged, the act is performed at the endpoints, before the data is encrypted or after it is decrypted.

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE HTTPD
Creates a Hypertext Transfer Protocol (web server) daemon.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

DELETE TLSD
Terminates an SSL/TLS daemon.

QUERY TLSD
Displays currently active TLS daemons.

DEFINE TRACE

The DEFINE TRACE command starts a tracing operation for datagrams. IP traffic is saved in memory until a corresponding DELETE TRACE command is issued. The DUMP command may be used at any time to output the accumulated trace buffers.

Syntax

DEFine TRACe ID=*id* [,IPaddr=*ip4addr*] [,PORT=*port*]
[,SIZe=500] [,KIND={TCP|UDP|ICMP|ALL}]

Arguments

ID=
A unique name that identifies this trace entry. This name is used in conjunction with the DELETE TRACE and QUERY TRACES commands.

IPaddr=
Limits tracing to a specific IP address. Only traffic involving the specified address (inbound or outbound) is considered.

KIND=
This parameter selects the type of network traffic that is recorded.

TCP
Only datagrams containing TCP payloads are traced. This is the default.

UDP
Only datagrams containing UDP payloads are traced.

ICMP
Only ICMP (for example, PING) datagrams are traced.

ALL
All datagrams are included in the trace.

PORT=
Limits tracing to a specific port. Only traffic involving the specified port (outbound or inbound) is considered.

SIZe=
Indicates the maximum message blocks that are to be retained during the trace. Old blocks are discarded in favor of newer blocks. The default is 500. Allowable values are in the range of 10 through 64K.

Example

```
define trace,id=trace2,ipaddr=192.168.1.66,kind=udp
IPN210I Trace ID TRACE2 defined and running.

IPN213I TRACE2 tracing traffic between 192.168.1.161; 21 and 192.168.1.66; 2646
IPN213I TRACE2 tracing traffic between 192.168.1.161; 4109 and 192.168.1.66; 2647
```

Notes

The following notes apply to this command:

- Once trace data has been accumulated, the DUMP command can be used to dump the data.
- Be sure to terminate your trace with DELETE TRACE after you use the DUMP command. Because all traffic to a traced address is retained in memory, large amounts of virtual storage can be tied up.
- You can trace multiple addresses simultaneously by issuing multiple DEFINE TRACE commands. Each trace must have a unique ID.

Related Commands

DEFINE SOTRACE
Starts a Socket Trace.

DELETE TRACE
Terminates a trace and frees its storage.

DUMP
Performs a formatted dump of various TCP/IP control blocks.

QUERY TRACES
Displays a list of currently running traces.

SEGMENT
Segments the SYSLST and log files, making them available for printing.

DEFINE TRANSLATION

The DEFINE TRANSLATION command controls the availability of translate tables used by various TCP/IP FOR VSE clients and daemons. You can load, name, and provide for default translation.

Three forms of this command are available. The first defines single-byte translation tables. The second form defines double-byte character set (DBCS) translations. The third form establishes a previously loaded translate table as the system-wide default.

Syntax

```
DEFine TRANslation MEMber=member [,ENTry=name]
        [,NAMe=name16] [,DEFault=name16]

DEFine TRANslation TYpe=Double ,MEMber=member
        AScii=name ,EBcdic=name ,NAMe=name16
        [,DEFault=name16]

DEFine TRANslation DEFault=name16
```

Arguments

MEMber=

Specifies the name of the library member in which the source for one or more translation tables resides. The member is an ".L book" and must reside in a library in the partition's search chain. Information on member format and customizing translate tables is contained in the *TCP/IP FOR VSE Installation Guide*.

ENTry=

Single-byte translate table members can contain any number of translate tables. Specify this parameter to restrict processing to a single translate table. If you omit both this parameter and the NAME parameter, all translate tables in the member are loaded and are identified to the system by their entry names. This parameter is ignored if TYPE=DOUBLE is specified.

NAMe=

Specifies the name that is to be assigned to the translate table after it is loaded. It is required if TYPE=DOUBLE is specified. For single byte tables, NAME= defaults to the value of ENTRY=.

DEFault=

Specifying this parameter, either by itself or in conjunction with loading one or more translate tables, changes the system default table. This is the table that is used for ASCII/EBCDIC translation when the user does not override it.

TYpe=
Indicates the type of table being defined.

Single
A Single-Byte Character Set (SBCS) will be loaded. This is the default.

Double
A Double-Byte Character Set (DBCS) will be loaded.

AScii=
If you specify TYPE=DOUBLE, you must use this parameter to specify the name of the ASCII code page that is used to generate the translate table.

EBcdic=
If you specify TYPE=DOUBLE, you must use this parameter to specify the name of the EBCDIC code page that is used to generate the translate table.

Example

```
 define translate,mem=ipxlate,entry=os_02,default=os_02
IPN651I SBCS translation tables being loaded from IPXLATE.L
IPN654I Translation table created: OS_02
IPN656I SBCS Table loading complete for member IPXLATE
IPN657I Default SBCS translate table set to OS_02
```

Notes

The following notes apply to this command:

- Once loaded, a translate table cannot be deleted. If the same-named table is reloaded, the new table will replace the old.
- Double-byte character sets are used only with Chinese, Japanese, and Korean.
- To define a DBCS translate table, you must also define a corresponding SBCS translate table and assign them identical names.
- See the *TCP/IP FOR VSE Installation Guide* for more information about using translation tables.

Related Commands

QUERY TRANSLATES
Displays a list of available translate tables.

SET TELNET_TRANSLATE
Sets the name of the translate table that will be used with Telnet (not TN3270) connections.

DEFINE USER

The DEFINE USER command allows you to specify the user IDs of authorized TCP/IP FOR VSE users in the absence of a user-provided security exit.

Syntax

```
DEFine USEr ID=name16 [,PASSword=name16] [,DATA=any]
        [,GID=snum] [,UID=snum] [,MAILbox=str]
        [,FTP={YES|NO}] [,LPR={YES|NO}]
        [,WEB={YES|NO}] [,TELNET={YES|NO}] [,ROOT=path]
```

Arguments

ID=

This value will be used as the User ID. It will be converted to upper case.

PASSword=

If specified, the user must provide the matching value before logon is permitted.

Passwords are from 1 to 16 characters long and are always converted to upper case.

DATA=

This is an optional data string of up to 40 user-specified characters.

TCP/IP FOR VSE does not examine this field on input. Its contents are passed to the Automatic Security Exit (if active) and then to the installation-supplied security exit, if you provide one. No case conversion is performed on this field. If the field contains blanks or commas, it must be enclosed in single quotes.

When passed to the Automatic Security Exit, each position of the data string should contain either “Y” or “N” to indicate that functions are either allowed (Y) or disallowed (N). The positions within the data string correspond to the value passed in the SXBLOK DSECT's SXTYPE field. The values are shown below in the section “Automatic Security Exit,” page 137.

GID=

Signed numeric, -9999999 through +9999999.

The GID defines this user as part of a group. TCP/IP FOR VSE does not use this field but passes it to the security exit.

UID=

Signed numeric, -9999999 through +9999999.

The UID associates this user with a UNIX-style user ID. TCP/IP FOR VSE passes this field to the security exit.

FTP=
Determines whether the user is authorized to use FTP.

LPR=
Determines whether the user is authorized to use LPR.

WEB=
Determines whether the user is authorized to make HTTP requests. Note that the HTTP daemon must also be configured to accept user IDs.

TELNET=
Determines whether the user is authorized to access Telnet menus. For this to be effective, the menu must make provision to poll for a user ID and password.

ROOT=
If specified and the user is authorized for FTP, the FTP session will begin in this directory. The user will be able to change to lower-level subdirectories, but they will be prevented from accessing higher-level directories.

If this value contains special characters, it must be enclosed in apostrophes.

Example

```
 define user,id=don,password=republican, data='Spills Coffee'
 define user,id=leo, data='Drinks Coffee' ftp=yes, -
IPN237I ++SUPRESSED++

 query users
IPN253I << TCP/IP User IDs >>
IPN475I  User ID: LEO
IPN476I   Data: Drinks Coffee
IPN883I   Valid for: FTP
IPN475I  User ID: DON
IPN476I   Data: Spills Coffee
IPN883I   Valid for: *All*
```

Notes

The following notes apply to this command:

- User IDs and passwords are case insensitive.
- Special characters should be avoided because a user may have difficulty in providing exactly matching values from some platforms.
- If the password is not specified, any value provided by the user is accepted unless the security exit (if any) determines otherwise.
- To modify an entry, you must delete and redefine it.

- In case of duplicate entries, the one entered first is used.
- The FTP daemon, the Telnet daemon (if a menu is supplied containing the appropriate fields), and the HTTP daemon check user IDs and passwords. The HTTP daemon checks user IDs if the SECURITY=ON parameter is specified in the DEFINE HTTPD command.
- If no limitation is placed on the user ID by FTP=, LPR=, WEB=, or TELNET=, then the user ID is authorized to be used with all services.
- See the *TCP/IP FOR VSE Installation Guide* for more information about user IDs and security.
- Any input statement whose first character is "+" is not echoed on the TCP/IP log. This includes each line of a "continued" command line.

Automatic Security Exit

When activated by the SECURITY and ASECURITY commands, the Automatic Security Exit validates each user for the task being performed. User authorization is checked by examining specific field positions in the DATA= string supplied with DEFINE USER.

To permit a function, a "Y" should appear in its assigned position; to disallow the function, code "N". The following table shows column numbers and their assigned functions. The values are those found in the SXTYPE field of the SXBLOK mapping macro. For examples, see "Auto Security Manager" in chapter 9, "Security," in the *TCP/IP FOR VSE Installation Guide*.

1. Password Check
2. Read Check
3. Write Check
4. Update Check
5. Startup Security
6. Shutdown Security
7. Hardware Address Verify
8. IP Address Verify
9. SITE Command check
10. Delete check
11. Rename check
12. Create check
13. EXEC command check

14. APPEND check
15. OPDIR check
16. RDDIR check
17. CWD Check
18. SHELL Check
19. ICMP check
20. Daemon LOGIN request
21. RPC Request
22. Web Logon Screen Request
23. HTTPD SCANBLOCK request
24. Make directory
25. Remove directory
26. Last CWD
27. Auto exit startup
28. Auto exit shutdown
29. FTPD command

Related Commands

ASECURITY
Configures the Automatic Security Exit

DELETE USER
Removes a user ID and password entry.

QUERY USERS
Displays a list of defined user IDs.

SECURITY
Controls TCP/IP security functions.

DELETE ALTIP

The DELETE ALTIP command deletes IP addresses that were previously identified using the DEFINE ALTIP command.

Syntax

DELete ALTIP ID=*id*

Arguments

ID=

The ID= parameter that was specified in the DEFINE ALTIP command that created the entry.

Example

```
delete altip,id=sys2
IPN581I ALTIP SYS2 has been deleted.
```

Notes

Deleting an ALTIP entry does not immediately cancel its effects. This is because most hosts on your network issue an ARP request and then retain the results for a period of time. These values eventually expire (or the host reboots) and a new ARP request is issued.

Related Commands

DEFINE ALTIP

Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE LINK

Creates a link between TCP/IP and a network or a directly connected stack.

DEFINE ROUTE

Adds an entry to the TCP/IP routing table.

QUERY ALTIPS

Displays all alternate IP addresses.

DELETE CGI

The DELETE CGI command deletes a CGI program from storage. When you delete a CGI, you are only deleting the module from storage. You are not deleting the CGI definition. The next time the program is invoked it is loaded into storage again. This is useful when you want to refresh a CGI.

Syntax

DELete CGI PUBLIC=*pubname*

Arguments

PUBLIC=

Specifies the phase name from an Assembler program or the PROC name for a REXX CGI.

Example

```
 delete cgi,pub=procavg
IPN694I CGI Deleted
```

Notes

For more information about using CGIs with TCP/IP FOR VSE, see the *TCP/IP FOR VSE Programmer's Guide*.

Related Commands

DEFINE CGI

Loads a CGI program and makes it available for use.

QUERY CGIS

Displays all currently available CGI programs.

DELETE EVENT

The DELETE EVENT command allows you to remove events created by the DEFINE EVENT command.

Syntax

DELete EVENT {ID=*id*|ALL[,FORCE]}

Arguments

ID=

The ID parameter that was specified in the DEFINE EVENT command that created the entry.

ALL

If specified, all events are deleted and the Event task is terminated.

FORCE

If specified with ALL, this parameter forces an immediate termination of the auto Event task. Any in-flight transmissions are lost. If not specified with ALL, this parameter is ignored.

Example

```
 delete event,id=ftp_listen
IPN800I Event scheduled for deletion: FTP_LISTEN
TCP913I Event Deleted: FTP_LISTEN
```

Notes

The following notes apply to this command:

- The initial DEFINE EVENT creates a single automation daemon task that monitors the POWER queue and processes the files. This task remains until the last event is deleted with DELETE EVENT.
- Deleting an event may have no immediate effect. Issuing a delete flags the element for deletion. The automatic daemon task performs the actual delete when it completes any in-flight transmission.
- Using FORCE during transmission of a file (FTP or LPR) may cause the remote daemon to hang. Such an event is rare, and normal network timeout mechanisms should correct the hang condition within a few minutes.

Related Commands

DEFINE EVENT

Monitors a POWER class for automatic report distribution.

QUERY EVENTS

Displays the status of automation processing.

DELETE FILE

The DELETE FILE command removes a file from the TCP/IP FOR VSE file system.

Syntax

DELete FILE PUBlic=*pubname*

Arguments

PUBlic=

The public name of the file to be deleted.

Example

```
 delete file,public=bimlib
IPN581I File definition has been deleted.
```

Notes

The following notes apply to this command:

- Deleting a file has no effect on users who have already opened the file.
- When files are defined, a directory structure is created. Files exist only in the lowest level of this structure. For example, if you define a file as TEST.USERS.VSAM.FILE1, the file system contains four directories (root, /TEST, /TEST/USERS/, and /TEST/USERS/VSAM) and a file entry (in the lowest level directory). If you delete the file, only the file entry disappears. The /TEST/USERS/VSAM directory still exists but is empty. There is no way to delete the directory structure.

Related Commands

DEFINE FILE

Defines a file in the TCP/IP FOR VSE file system and associates it with a file I/O driver.

QUERY FILES

Displays the contents of the TCP/IP FOR VSE file system.

DELETE FILEIO

The DELETE FILEIO command removes a loaded I/O driver phase from storage.

Syntax

DELete FILEIO,PHASE=*name*

Arguments

PHASE=
A phase name listed in QUERY FILEIO output.

Example

```
 delete fileio,phase=IPNFBIME
IPN264I File driver deleted, Dataset: IPNFBIME
```

Notes

The following notes apply to this command:

- You must use DELETE FILEIO to remove a loaded phase name before you can reload that phase.
- The order of the startup of TCP/IP and BIM-EDIT is important for the BIM-EDIT interface to work properly. If TCP/IP was started before BIM-EDIT, then it may be necessary to issue the following statements

  ```
  DELETE FILEIO,ID=IPNFBIME
  DEFINE FILEIO TYPE=BIM-EDIT
  ```

  even though it may appear that the file IO driver for BIM-EDIT was loaded at TCP/IP startup time.

  You then need a DEFINE FILE statement such as

  ```
  DEFINE FILE,PUBLIC='BIMEDIT',DLBL=BIFLIB,TYPE=BIM-EDIT
  ```

- In general, you should only remove driver phases that were loaded using DEFINE FILEIO.

Related Commands

DEFINE FILEIO
Loads a file I/O driver phase into storage.

QUERY FILEIO
Displays the status of the file I/O driver programs.

DELETE FTPD

The DELETE FTPD command removes an FTP daemon (server) from the system.

Syntax

DELete FTPd ,ID=*id*

Arguments

ID=
The ID of the FTP daemon that is to be deleted.

Example

```
 delete ftpd,id=ftp01
FTP912I FTP01 not accepting connections on port 21
FTP908I Daemon Shutdown FTP Id: FTP01 Port: 21
```

Notes

The following notes apply to this command:

- There is no MODIFY FTPD command. To change any specification, you must delete and redefine the FTP daemon.
- To terminate a hung FTP session, consider using the FLUSH command.

Related Commands

ASECURITY
Configures the Automatic Security Exit

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE USER
Creates a user ID and password.

FLUSH
Terminates all processing with a specific remote host.

QUERY FTPDS
Displays the status of the File Transfer Protocol daemons.

QUERY CONNECTIONS
Displays the status of one or more connections.

QUIESCE
Prevents new connections while allowing existing connections to continue.

DELETE GPSD

The DELETE GPSD command terminates an active GPS Daemon. When you issue this command, processing stops immediately.

Syntax

DELETE GPSD ID=*id*

Arguments

ID=

The ID= parameter used to define the GPS Daemon.

Example

```
 delete gpsd,id=gpsd1
GPS922I GPSD1 GPS Shutting down
GPS924I GPSD1 GPS Shutdown complete
```

Related Commands

DEFINE GPSD

Creates a General Print Server daemon.

QUERY GPSDS

Displays the status of the General Print Server daemons.

DELETE HTTPD

The DELETE HTTPD command removes an HTTP daemon (server) from the system.

Syntax

DELete HTTPd ID=*id*

Arguments

ID=

The ID= value used to define the HTTP daemon that is to be deleted.

Example

```
 delete http,id=h1
HTT903I Daemon Shutdown HTTP
```

Notes

There is no MODIFY HTTPD command. To change specifications, you must delete and redefine the daemon.

Related Commands

DEFINE HTTPD

Creates a Hypertext Transfer Protocol (web server) daemon.

QUERY HTTPDS

Displays the status of the Hypertext Transfer Protocol (web server) daemons.

DELETE LINK

The DELETE LINK command permits you to eliminate a network access device or a driver from your current configuration.

Syntax

DELete LINK ID=*id*

Arguments

ID=

The ID= value used to define the link that is to be deleted.

Example

```
 delete link id=link3172
IPT101I Link Driver Processor LCS stopping
```

Notes

The following notes apply to this command:

- CAUTION**:**

  If you need to dynamically attach a new link, after you run DELETE LINK you must follow these steps:

  1. Delete all defined routes using DELETE ROUTE.
  2. Define the new link using DEFINE LINK.
  3. Define the routes again using DEFINE ROUTE.

  If you do not follow this order, after the new link is defined traffic will not be correctly directed to the appropriate connecting hardware. As an alternative to these steps, you can cycle the TCP/IP stack.

- When you use the DELETE LINK command, all ROUTE statements that refer to the link will be bypassed. These will reactivate if and when the link is redefined.
- When you use the DELETE LINK command, entries in the ARP table are discarded automatically.
- All adapter definitions associated with the link will also be deleted.

Related Commands

DEFINE ADAPTER

Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE LINK

Creates a link between TCP/IP FOR VSE and a network or to a directly connected stack.

QUERY LINKS

Displays the status of network links.

DELETE LPD

The DELETE LPD command removes an LPD (server) from the system. If the LPD is currently in use, it is flagged for deletion as soon as it finishes the current operation.

Syntax

DELete LPD ID=*id*

Arguments

ID=

The PRINTER= value used to define the LPD that is to be deleted.

Example

```
query lpds
IPN253I << TCP/IP Line Printer Daemons >>
IPN444I  Print Name: HEX UserID: $LPD Hexdump Mode
IPN445I  Queue: POWER.LST.A.
IPN446I  Library: MEMORY.
IPN444I  Print Name: LOCAL UserID: $LPD
IPN445I   Queue: POWER.LST.A.
IPN446I   Library: MEMORY.

delete lpd,id=local
LPD932I LPD queue scheduled for deletion: LOCAL
LPD935I LPD queue deleted: LOCAL

query lpds
IPN253I << TCP/IP Line Printer Daemons >>
IPN444I  Print Name: HEX UserID: $LPD Hexdump Mode
IPN445I  Queue: POWER.LST.A.
IPN446I  Library: MEMORY.
```

Related Commands

DEFINE LPD

Creates a Line Printer daemon.

FLUSH

Terminates all processing with a specific remote host.

DELETE MASK

The DELETE MASK command removes an entry in the subnet mask table.

Syntax

DELete MASK NETwork=*ip4addr*

Arguments

NETwork=

The network whose mask is to be deleted. The host portion, if not zero, is ignored.

Example

```
 query masks
IPN253I << TCP/IP Network Masks >>
IPN575I  Network: 192.168.30.0 (12,625,950) Mask: 255.255.255.0
IPN575I  Network: 127.0.0.0 (127) Mask: 255.0.0.0
IPN575I  Network: Default (--) Mask: 255.255.255.0

 delete mask network=192.168.30.0
IPN581I Mask 192.168.30.0 has been deleted.
```

Notes

The following notes apply to this command:

- For a discussion of addressing, see the section "Network Addressing" in the *TCP/IP FOR VSE Installation Guide*.
- The default subnet mask is unaffected by the DELETE MASK command. To change the default mask, use the SET MASK command.
- Deleting a subnet mask entry changes how your DEFINE ROUTE statements are interpreted.
- Deleting a subnet mask can affect how Telnet and FTP daemons are assigned, if their definitions use the IPADDR= parameter.

Related Commands

DEFINE MASK

Creates a subnet mask for a particular network.

QUERY MASKS

Shows all defined subnetwork masks by network number.

QUERY ROUTES

Displays the content of the network routing table or the route taken to reach a specific address.

DELETE MENU

The DELETE MENU command removes a telnet menu from use and frees the storage.

Syntax

DELete MENu ID=*id*

Arguments

ID=

The identifier from the DEFINE MENU command that created the menu you are deleting.

Example

```
 delete menu,id=menuadm
IPN581I Menu MENUADM has been deleted.
```

Notes

The following notes apply to this command:

- Menus are accessed by a telnet daemon only when a connection is requested.
- Deleting a menu while it is being displayed to a user causes unpredictable results.

Related Commands

DEFINE MENU
Loads a menu file and makes it available for use by Telnet daemons.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

QUERY MENUS
Displays menus available for TN3270 use.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

DELETE NAME

The DELETE NAME command removes an entry from the symbolic name table.

Syntax

DELete NAME NAME={*name*|DYNAMIC}

Arguments

NAME=

The symbolic name to be deleted. Case is ignored. A special reserved name, "DYNAMIC," can be specified to delete all cached entries that were created by a DNS lookup.

Example

```
 delete name,name=devpun
IPN581I Name DEVPUN has been deleted.
```

Related Commands

DEFINE NAME

Associates a TCP/IP FOR VSE name with an IPv4 address or a script file.

QUERY NAMES

Displays TCP/IP FOR VSE names and associated values.

DELETE NTPD

The DELETE NTPD command removes a Network Time Protocol daemon from the system.

Syntax

DELete NTPd ID=*id*

Arguments

ID=
The ID that identifies the daemon to be deleted.

Example

```
 delete ntpd,id=ntp1
NTP102I NTP Daemon shutdown complete
```

Related Commands

DEFINE NTPD
Starts an NTP daemon.

DEFINE NTPD
Creates a Network Time Server daemon.

QUERY NTPDS
Displays the status of NTP daemons

DELETE PUBLISHER

The DELETE PUBLISHER command removes a Publisher daemon from your configuration.

Syntax

DELete PUBLisher ID=*id*

Arguments

ID=

The identifier from the DEFINE PUBLISHER command that created the daemon you are deleting.

Example

```
 delete publisher,id=pub01
IPN694I Publishing Daemon shutting down
```

Related Commands

DEFINE PUBLISHER

Creates a Publishing daemon.

QUERY PUBLISHER

Displays the status of the Publisher daemon,

DELETE ROUTE

The DELETE ROUTE command removes routing information from the TCP/IP FOR VSE internal routing table.

Syntax

DELete ROUTe ID=*id*

Arguments

ID=

The ID parameter specified with the DEFINE ROUTE command that created the route entry.

Example

```
 delete route id=default
IPN581I Route DEFAULT has been deleted.
```

Related Commands

DEFINE ROUTE

Adds an entry to the TCP/IP FOR VSE routing table.

MODIFY ROUTE

Changes values on an existing entry in the routing table.

QUERY ROUTES

Displays the content of the network routing table or the route taken to reach a specific address.

DELETE TELNETD

The DELETE TELNETD command removes a Telnet daemon from your configuration.

Syntax

DELete TELnetd ID=*id*

Arguments

ID=

The identifier from the DEFINE TELNETD command that created the daemon you are deleting.

Example

```
 delete telnetd,id=lu05
TEL918I Daemon Shutdown Telnet Termname:TELNLG05
```

Notes

The following notes apply to this command:

- To clear an unwanted session, try using the FLUSH command. The user is dropped, and the daemon is freed for other users.
- If Telnet sessions are left hanging, consider a lower value for SET PULSE_TIME. This permits TCP/IP FOR VSE to detect and reset "orphaned" Telnet sessions automatically.

Related Commands

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon

FLUSH
Terminates all processing with a specific remote host.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

SET PULSE_TIME
Controls the default setting for the interval between probes of inactive connections (PULSE= parameter on DEFINE ROUTE).

DELETE TLSD

The DELETE TLSD command removes an SSL/TLS daemon from your configuration.

Syntax

DELete TLSd ID=*id*

Arguments

ID=

The identifier from the DEFINE TLSD command that created the daemon you are deleting.

Example

```
 delete tlsd,id=tlsdtnds
TLS903I Daemon Shutdown TLS Id:TLSDTNDS
```

Notes

The following notes apply to this command:

- Deleting a TLS daemon terminates any connections that are using it.
- The QUIESCE command may be useful to prevent new connections while existing ones complete.

Related Commands

DEFINE TLSD
Creates an SSL/TLS daemon.

FLUSH
Terminates all processing with a specific remote host.

QUERY CONNECTIONS
Displays the status of one or more connections.

QUERY TLSD
Displays currently active TLS daemons.

QUIESCE
Prevents new connections but permits existing connections to continue.

DELETE TRACE

The DELETE TRACE command terminates a running trace and frees all storage associated with it. This includes traces started both with DEFINE TRACE and DEFINE SOTRACE.

Syntax

DELete {TRACe ID=*id* | TRACES}

Arguments

ID=
The ID that identifies the trace entry to be deleted. This name was assigned by a corresponding DEFINE TRACE command.

Example

```
 delete traces
IPN581I Trace TRACE2 has been deleted.
IPN581I Trace TRACE1 has been deleted.
```

Notes

You must issue the DUMP command to dump the running trace before you issue the DELETE TRACE command because DELETE TRACE releases all trace-related storage.

Related Commands

DEFINE SOTRACE
Starts a Socket Trace.

DEFINE TRACE
Starts a Datagram Trace.

DUMP
Performs a formatted dump of various TCP/IP FOR VSE control blocks.

QUERY TRACES
Displays a list of currently running traces.

DELETE USER

The DELETE USER command removes a user ID entry from the user ID table.

Syntax

DELete USEr NAMe=*id*

Arguments

NAMe =
The user name to be deleted.

Example

```
 delete user,name=guest
IPN581I User GUEST has been deleted.
```

Related Commands

DEFINE USER
Create a user ID and password.

QUERY USERS
Displays a list of defined user IDs.

DIAGNOSE

The DIAGNOSE command allows you to gather diagnostic information specific to a given function. The command's output is primarily intended to aid CSI Technical Support in solving problems you might encounter.

To direct DIAGNOSE command output to both SYSLST and the TPC/IP console, see the SET DIAGNOSE command.

Syntax

DIAGnose {OFF|[-]*keyword*}

Arguments

keyword

Specifying one of the keywords below turns on the related diagnostic message. Specifying a keyword with a '-' prefix turns off the related diagnostics.

ARPs

Tracks address resolution protocol (ARP) requests.

AUTOmation

Displays the email scripts triggered by DEFINE EVENT commands as the scripts execute. It looks at what happens before the internal client gets control and also after the internal client has finished, such as the final disposition of a report. This can help solve problems related to automatic EMAIL, LPR, or FTP.

CHECksum

When in effect, datagrams rejected because of a failed checksum are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

CLEAnup

Displays the progress of the stack's periodic cleanup process.

CLOSe

Produces diagnostic messages related to the various stages for the TCP connection CLOSE process.

CONNect

This is an alias for the TCP keyword on page 162.

CONNReject | CONReject

Displays diagnostics for rejected connection requests for TCP connections.

CONTrol

Produces information about SOCKET CONTROL connection calls.

DEBug

Produces information on certain modules and their functions that is generally useful only to CSI Software Development, but it

may be requested by CSI Technical Support as well when debugging a problem that affects those modules.

DNC
Produces information relating to the Domain Name Client.

DTLOAD
Monitors phase loading and management.

EMAIL
Produces Email Client diagnostics.

FILEIO
Produces messages containing file system record counts.

FILEREQ
Produces information about file system requests.

FRAGment
Monitors datagram fragmentation/defragmentation.

FTP
Produces diagnostics related to internal FTP clients (not FTPBATCH).

FTPD
Produces additional information from the FTP daemon.

GETVIS
Shows high-water usages of system GETVIS when issued in the TCP/IP partition. This is different from CLEANUP, which also causes messages and dumps related to cleaning up abandoned connections **Note:** To show GETVIS usage by subpools, use the VSE command GETVIS SVA,DETAILS.

GPS
Produces GPS daemon diagnostics.

HASHing
Producing diagnostics related to the CCBLOK (Connection Control Block) services.

HTTP
Produces information from HTTPD processes.

IBBLOK
Produces IBBLOK management diagnostics.

ICMP
Tracks the production and handling of ICMP (ping) datagrams.

LIBR
Monitors the I/O of Librarian members.

LINK
Produces link driver diagnostics (other than CLAW).

LOCks
Produces diagnostic messages for the internal locks manager.

LPD
Produces Line Printer daemon diagnostics.

LPR
Produces diagnostics for LPR-related processes.

MISRoutedip
When in effect, misrouted IP datagrams are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

NONIp
When in effect, non-IP datagrams are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

OFF
Turns off the diagnose operation and resets all options.

PERForm
Displays performance data after the close of each connection.

POWer
Dumps the data areas returned from the POWER Application Programming Interface. You should use this parameter only if you are experiencing trouble with the POWER File/IO driver or as directed by CSI Technical Support.

PUBLish
Causes information to me stored (“published”) that can be later reviewed by running the PUBLISH batch utility.

PULSe
Produces information during outbound PULSE operations.

REJICmp
When in effect, rejected incoming ICMP datagrams are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

REJIGmp
When in effect, incoming IGMP datagrams are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

REJLngth
When in effect, incoming datagrams rejected because of an improper length are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

REJProto
When in effect, incoming datagrams rejected because of an "unknown protocol" are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

REJUdp
When in effect, rejected incoming UDP datagrams are summarized and dumped. This setting is disabled after 10 datagrams are intercepted and displayed.

RESet
Displays information whenever a TCP connection is reset by the foreign host or client.

RETRANsmit
Monitors TCP retransmit operations.

ROUTing
Displays information while the routing tables are being searched.

SECUrity
Produces security-processing diagnostics.

SOCkets
Produces socket interface messages.

SSL
Produces Secure Sockets Layer diagnostics.

TCP
Produces voluminous diagnostics on SYSLST that are useful to CSI Technical Support in diagnosing TCP issues. Use it briefly only when recommended by CSI Technical Support, and during periods when it will not affect your production environment.

TELetd
Produces information from TN3270 daemons

TELProxy
Produces Telnet Proxy diagnostics.

UDP
Enables diagnostic tracing of UDP datagrams through the stack.

WEB
Like HTTP, it issues messages that may be of interest while debugging a problem. Unlike DIAGNOSE HTTP, its focus is on special features rather than general HTTP operations.

Example

```
I diagnose perform
IPN524I Diagnose status for Perform set to on
```

Notes

The following notes apply to this command:

- Output from the DIAGNOSE command can be voluminous.
- Multiple diagnose options can be in effect concurrently. Reissuing the DIAGNOSE command with a different operand adds (or subtracts) the new value. To eliminate a value, you can clear it individually with a '-' prefix, or you can use the "OFF" operand to clear all entries.

Related Commands

MODIFY LOG
Changes the characteristics of a system log file.

QUERY DIAGNOSE
Displays the current DIAGNOSE settings in effect.

RAPTRAC
Records significant events that occur during TCP/IP FOR VSE processing. Used for troubleshooting at the direction of CSI Technical Support.

SET DIAGNOSE
Enables console display of messages resulting from the DIAGNOSE command.

DISCOVER

The DISCOVER command allows you to determine the largest MTU size permitted to a particular remote host.

Syntax

DISCover *host*

Arguments

host

Specifies the IP address of the target host. This may be an actual numeric address or a symbolic name that can be resolved to an IP address.

Example

```
 discover 192.168.2.10
TCP910I Test for MTU 576 succeeded
TCP910I Test for MTU 1501 failed
TCP910I Test for MTU 1038 succeeded
TCP910I Test for MTU 1269 succeeded
TCP910I Test for MTU 1385 succeeded
TCP910I Test for MTU 1443 succeeded
TCP910I Test for MTU 1472 succeeded
TCP910I Test for MTU 1486 succeeded
TCP910I Test for MTU 1493 succeeded
TCP910I Test for MTU 1497 succeeded
TCP910I Test for MTU 1499 succeeded
TCP910I Test for MTU 1500 succeeded
TCP910I Test for MTU 1501 failed
TCP910I The best MTU discovered: 1500.
```

Notes

The following notes apply to this command:

- Each DISCOVER command issues multiple PINGs of varying lengths to determine the maximum available MTU size.
- If the path to the remote host changes, the MTU value may change.
- If the path to the remote host is not reliable (for example, if there is packet loss), DISCOVER may return a lower than optimal value. For this reason, it may be useful to repeat the command.

Related Commands

DEFINE LINK

Creates a link between TCP/IP FOR VSE and a network or to a directly connected stack.

DEFINE ROUTE

Adds an entry to the TCP/IP FOR VSE routing table.

PING

Issues an ICMP Echo (PING) request.

TRACERT

Displays each “hop” in a route along with the time needed to reach it.

DOWNCHECK

The DOWNCHECK command controls whether a confirmation is required when the SHUTDOWN command is entered.

Syntax

DOWNcheck {ON|OFF}

Arguments

ON

(Default) When a SHUTDOWN command is entered, the operator is prompted for verification.

OFF

An operator SHUTDOWN command takes effect immediately, without a prompt.

Example

```
 downcheck on
IPN268I DOWNCHECK now set to ON
```

Related Commands

QUERY OPTIONS

Displays the current values of modifiable parameters.

SHUTDOWN

Terminates processing and shuts down the stack.

DUMP

The DUMP command is used for diagnostic purposes. When issued, it formats and prints the contents of the requested storage to SYSLST.

Syntax

```
DUMP {ALL|BUFfers|BUS|CONnects|DIRectory|Events|EXITS
      FRAgments|FREquests|FTPds|GIVesockets|GPSds|
      HTTpds|IVBlok|LINks|LPDs|MASks|MENus|NAMes|
      PARTITION|PROGrams|ROUtes|SOCkets|STAts|TASks|
      TELnetds|TIMERS|TLSds|TRACes|TRANslations|
      USErs|VERsions}
```

Arguments

ALL
All eligible areas are dumped. This option requires a lot of CPU and spool space.

BUFfers
The pooled Telnet buffers are dumped.

BUS
The data bus (IBBLOKs) is dumped.

CONnects
Connection control blocks are dumped.

DIRectory
The file system directory blocks are dumped.

Events
Currently queued automation Event Blocks are dumped.

EXITS
The contents of the control blocks (XIBLOK) created by issuing a DEFINE EXITS are dumped.

FRAgments
Datagrams fragments currently being processed are dumped.

FREquests
In-flight file I/O requests are dumped.

FTPds
Control blocks belonging to FTP daemons are dumped.

GIVesockets
The GiveSocket queue is dumped.

GPSds
Control blocks belonging to GPS daemons are dumped.

HTTpds
Control blocks belonging to HTTP daemons are dumped.

IVBlok
The Internal Vector Block is dumped.

LINks
Link driver control blocks are dumped.

LPDs
LPD control blocks are dumped.

MASks
The masks created by DEFINE MASKS are dumped.

MENus
TelnetD menus are dumped.

NAMes
Blocks created by DEFINE NAME are dumped.

PARTITION
Partition storage for the TCP/IP stack is dumped.

PROGrams
Loaded programs are dumped.

ROUtes
The routing table is dumped.

SOCkets
Currently allocated socket blocks are dumped.

STAts
The content of the statistics table is dumped.

TASks
The pseudo task control blocks are dumped.

TELnetds
Control blocks belonging to Telnet daemons are dumped.

TIMERS
Any timers that are internally defined and awaiting completion are dumped.

TLSds
Control blocks belonging to TLS daemons are dumped.

TRACes
Dumps the content of all currently running DEFINE TRACE and DEFINE SOTRACE commands.

TRANslations
The translate tables are dumped.

USErs
User blocks, as created by DEFINE USER, are dumped.

VERsions
The versions table is dumped.

Example

```
 dump users
IPN206I Dump has completed
```

Notes

The following notes apply to this command:

- Except for DUMP TRACES, output from the DUMP command is of little use to the installation. In general, you should use this command only when directed by CSI Technical Support.
- Regardless of log settings, the DUMP command never sends data to the console.
- Output from the DUMP command can be voluminous. TCP/IP FOR VSE processing is likely to be impacted during the dump.

Related Commands

DEFINE SOTRACE
Starts a Socket Trace.

DEFINE TRACE
Starts a Datagram Trace

QUERY TRACES
Displays a list of currently running traces.

DUMPOPTION

This command allows you to specify dump options for TCP/IP-related errors.

Syntax

DUMPOption FILeio={Csi|Ibm|None}

Arguments

FILeio=

This option controls dumps caused when the File I/O subtask abends. This subtask is considered to be "expendable" because certain common I/O-related conditions cannot be trapped by the stack nor can a retry routine be scheduled. Following an abend, the subtask is reattached and all queued I/O requests are recovered.

Csi

The dump produced is in CSI format, and various internal TCP/IP related areas are formatted. This is the default.

Ibm

The dump is created in standard IBM format. This may be more useful if the problem is recurring and/or if it involves a user-supplied File I/O routine, if IBM is being consulted, of if other third-party software is involved.

None

No dumps are produced. Use this option when you already have diagnostic information and further dumps would be a waste of resources.

Example

```
 dumpoption fileio=csi
IPN818I Dump Options for File I/O set to CSI format
```

Related Commands

QUERY FILEIO

Displays the status of the File I/O driver programs.

QUERY PROGRAMS

Displays the program phases being used by TCP/IP, their characteristics, their memory locations, and the library from which each was loaded.

EMAIL

The EMAIL command sets and displays various global values used by the EMAIL client to transmit email messages.

Syntax

```
EMAIL [,SMTPd=ip4addr] [,ATsign=7C] [,FRom=string64]
        [,DESTination=string64] [,REPlyto=string64]
        [,RPORT=25] [,SUBject=string64]
        [,USERid=$EMAIL] [,PASSword=$EMAIL]
        [,LUSERid=$EMAIL] [,LPASSword=$EMAIL]
        [,TRANslation=name16] [,TRAttachments=name16]
        [,TRMail=name16] [,CHECKname={Yes|No}]
        [,GMT=snum]
```

Arguments

The EMAIL command may be repeated, as needed, to establish and change the values. Default values are supplied the first time EMAIL is issued. Thereafter, if a parameter is omitted, the current value is left unchanged. Issuing EMAIL with no parameters is equivalent to issuing QUERY EMAIL.

ATsign=
The hexadecimal representation (EBCDIC) of the character used to separate the user ID portion of an e-mail address from the domain name. The default value is "7C" (@).

CHECKname=
Controls whether the EMAIL client checks for a valid e-mail name.

Yes
(Default) The e-mail name is checked to ensure that it is syntactically correct.

No
E-mail names are not checked and are used as is.

DESTination=
A 1- to 64-character string that defines a default domain for e-mail addresses. The EMAIL client forms a complete address by appending the at sign (@) and this string to the $$ LST DEST= parameter when DEFINE EVENT specifies HOSTname=USER. Setting this value is useful when a default domain (SET DEFAULT_DOMAIN) cannot be globally assigned in the TCP/IP FOR VSE initialization member. The domain you set with DESTINATION= takes precedence over the value defined by the SET DEFAULT_DOMAIN command.

FRom=
A 1- to 64-character string to be used as the default "FROM" field in e-mails.

GMT=
A signed numeric number in the range of +/- 2359 that is added algebraically to the local time to obtain GMT time. Times conveyed in e-mail messages are always expressed as GMT values. If not specified, the value set during IPL is used.

LPASSword=
This is the 1- to 16-character “local password” that will be passed to the Security Exit.

LUSERid=
This is the 1- to 16-character “local userid” that will be passed to the Security Exit.

PASSword=
This is the 1- to 16-character “SMTP password” that is used for access to the SMTP server if AUTH=ON is specified by the EMAIL client.

REPlyto=
A 1- to 64-character string to be used as the default “REPLY TO” field in e-mails.

RPORT=
This is the remote port for the SMTP server.

SMTPd=
The IP address that identifies the SMTP through which e-mail is to be routed.

SUBject=
A 1- to 64-character string to be used as the default “SUBJECT” field in e-mails.

TRANslation=
The name of the translation table to be used for translating between EBCDIC and ASCII. If not specified, the system default is used.

TRAttachments=
The name of the translation table to be used to convert attachments from EBCDIC to ASCII. If not specified, the value in effect for TRANSLATION= is used.

TRMail=
The name of the translation table to be used to convert the “body” text of the e-mail from EBCDIC to ASCII. If not specified, the value in effect for TRANSLATION= is used.

USERid=
This is the 1- to 16-character “SMTP userid” that is used for accessing the SMTP server if AUTH=ON is specified by the EMAIL client.

Example

```
 email gmt=-500
IPN253I << TCP/IP EMAIL >>
IPN835I UserID: $EMAIL RPort: 25
IPN838I Truncate: NO - AtSign: 7C GMT: -05:00
IPN840I Translation table (General): Default
IPN840I Translation table (Text): Default
IPN840I Translation table (Attachments): Default
```

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

QUERY EMAIL
Displays EMAIL client settings. **Note**: Issuing the EMAIL command without any parameters produces the same results.

SET DEFAULT_DOMAIN
Establishes a domain name to be automatically appended to unqualified names.

EXECUTE

The EXECUTE command directs TCP/IP FOR VSE to fetch a member from a VSE library and execute the contents as if the lines were entered from the console.

Syntax

EXECute *member*

Arguments

member

The name of the member containing the commands to be executed. A member type of '.L' is appended to the name. The member must be accessible in the LIBDEF search chain for the TCP/IP FOR VSE partition.

Example

```
 exec newroute
IPN397I Loading command deck NEWROUTE
IPN398I Command deck NEWROUTE has been completely loaded
IPN179I Direct Route 192.169.8.7= Net: 43272 Subnet: 0 Host: 7
IPN179I Direct Route 192.169.8.9= Net: 43272 Subnet: 0 Host: 9
```

Notes

The following notes apply to this command:

- It may be convenient to keep sets of like commands (such as DEFINE USER commands) in separate members and then execute them with the EXECUTE command.
- The EXECUTE command operates like the INCLUDE command.

Related Commands

INCLUDE

Includes a library member in the initialization parameter set.

EXTPGVS

The EXTPGVS command allows you to control whether external-partition socket requests are allocated in 31-bit private partition storage or in 31-bit system GETVIS storage.

You issue this command in the TCP/IP FOR VSE partition.

Syntax

```
EXTPGVS {ON|OFF}
```

Arguments

ON

The SVA-resident ASOCKET program CDLOADs the $BSOCKET.PHASE into private partition GETVIS. External socket requests are independently queued into a partition control block anchored in the COMREG-IJBTCPP2 field. The TCP/IP partition's CSOCKET program then uses access register mode to pull the TCP socket requests allocated in private 31-bit partition GETVIS from the external partition into the TCP/IP partition.

EXTPGVS ON, the default, assumes that you have adequate SVA storage to perform SOCKET allocation/deallocation. It is the preferred method for better performance.

OFF

The ASOCKET program uses the legacy method of allocating external-partition socket requests in 31-bit system GETVIS storage. At the time a socket send request occurs, the program copies the application buffers to system GETVIS and queues it to the TCP/IP partition.

EXTPGVS OFF is recommended for sites with inadequate SVA storage, as indicated by IPN960C messages from TCP/IP FOR VSE that warn of inadequate SVA storage.

Notes

The following notes apply to this command:

- For EXTPGVS ON, TCP/IP issues a checksum on the data, so when and if it changes the differences are detected.
- Non-TCP requests are not affected by the EXTPGVS ON setting.
- Use the QUERY EXTERNAL command to see overall SOCKET statistics for a specific batch partition as well as usage in a multi-stack environment. The statistics are reset at the start of a new job step.

Related Commands

DIAGNOSE GETVIS

Displays high-water usages of system GETVIS when issued in the TCP/IP partition.

QUERY EXTERNAL

Displays statistics from external socket-application partitions.

FIREWALL

The FIREWALL command controls and monitors the Firewall Shield optional feature.

For information on this feature, see the *TCP/IP FOR VSE Optional Features Guide*.

Syntax

The syntax is

```
FIREWALL {ON|OFF|LOAD [PHASE=phase-name]|WARN|FAIL|
          MSGON|MSGOFF|DEBUGON|DEBUGOFF|REPORT|ALLOWED|
          BLOCKED}
```

Arguments

ON
Is the default setting if the FIREWALL= parameter is used in the EXEC IPNET statement and a firewall configuration phase is successfully loaded during TCP/IP startup. If the firewall is not initially on, or it is turned off with the FIREWALL OFF command, FIREWALL ON can be issued to activate or reactivate the firewall.

OFF
Turns off the Firewall Shield feature.

LOAD PHASE=*phase-name*
Activates a new firewall configuration. This means a new firewall configuration phase will be loaded and activated. The default phase name is FIREWALL, but the PHASE= keyword can be used to load a different configuration phase.

WARN
Sets the firewall to warn mode. Firewall violations are displayed and logged, but an attempted access is allowed when this mode is active.

When in warn mode, the initial check for an IP address that does not match a range is allowed to pass to the next layer. The TCP ports, UDP ports, and ICMP are then also checked, and those checks may display additional blocked messages at the deeper layers for the corresponding protocol.

FAIL
Sets the firewall to fail mode. Firewall violations are displayed and logged, and each associated datagram is discarded immediately.

MSGON
Displays all blocked attempts on the console and SYSLST.

MSGOFF
Displays the first occurrence of blocked attempts on the console and SYSLST, but subsequent blocks are not displayed. A counter is maintained, and FIREWALL REPORT can be used to see the number of blocks for any specific IP address and port.

DEBUGON
Turns on debugging mode, and diagnostic dumps are created during firewall processing.

Caution:

FIREWALL DEBUGON can quickly send a lot of dumps to SYSLST in the TCP/IP partition. Use this setting only at the direction of Technical Support.

DEBUGOFF
Turns off debugging mode, and diagnostic dumps will not occur.

REPORT
Displays allowed IP addresses and IP addresses that were allowed but then blocked by TCPPORTS, UDPPORTS or ICMP.

ALLOWED
Displays a list of allowed IP addresses and the number of times each was allowed.

BLOCKED
Displays a list of IP addresses that were blocked because they are not in the firewall table.

Related Commands

IPSTAT
Controls whether an ISBLOK is allocated for every IP address that accesses the stack.

QUERY FIREWALL
Displays the current firewall settings being enforced.

QUERY IPSTAT
Displays IP address statistics stored in ISBLOKs.

FLUSH

The FLUSH command causes TCP/IP FOR VSE to flush all data and terminate all connections with the specified IP address.

Syntax

FLUSH *ip4addr* [,*port*]

Arguments

ip4addr
The network (IP) address to be flushed.

port
If specified, the remote port number restricts the operation of the FLUSH command to connections and data related to the port. If no port number is specified, all connections and data associated with the IP address are flushed.

Example

```
 flush 100.100.1.1
IPN500I All TCP/IP processing stopped
IPN357I Traffic has been flushed for IP: 100.100.1.1 Port: 0
IPN499I TCP/IP processing started
```

Notes

The FLUSH command is useful in terminating orphaned TN3270 sessions. It can also be used to clean up FTP sessions that are hung.

Related Commands

QUERY CONNECTIONS
Displays the status of one or more connections.

QUERY TELNETDS
Displays TN3270 and TN3270E Daemons.

FTPBATCH_FETCH

TCP/IP FOR VSE provides two batch FTP clients. One, invoked using EXEC FTPBATCH, provides its own internal FTP daemon and uses the TCP/IP partition only for data transmission. The other client, EXEC FTP, is only a client and relies on an FTP daemon running in the TCP/IP partition.

This command allows you to force an invocation of FTPBATCH, regardless of the program executed.

Syntax

FTPBatch_fetch {ON|OFF}

Arguments

ON
When the batch program "FTP" is executed, the FTPBATCH program will be fetched and executed instead.

OFF
Batch execution of "FTP" will run the batch FTP client.

Example

```
 ftpbatch_fetch on
IPN249I Value for FTPBATCH Fetch set to on
```

Notes

The following notes apply to this command:

- The batch FTP client "FTP" reads control cards and passes commands to the TCP/IP FOR VSE internal FTP client. This means that all disk/tape I/O and FTP processing occurs in the stack partition.
- FTPBATCH is self-contained and executes in a separate batch partition. It uses the stack partition only to process socket requests.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

GATEWAY

The GATEWAY command controls whether TCP/IP FOR VSE forwards transmissions between networks.

Syntax

GATEway {ON|OFF}

Arguments

ON

Datagrams not addressed to the stack are passed through the routing table and forwarded through the appropriate link.

OFF

Datagrams not addressed to the stack or directly connected stacks are discarded. This is the default and recommended value.

Example

```
 gateway off
IPN268I GATEWAY now set to OFF
```

Notes

The following notes apply to this command:

- Regardless of the GATEWAY setting, the stack always forwards datagrams to any stack identified by DEFINE ALTIP.
- Using TCP/IP FOR VSE as a router is not a cost-effective use of CPU if other routes are available. Use gateway support to redirect traffic to other partitions or LPARS that do not have their own direct access to the network.
- Most “misrouted” IP traffic is caused by a problem with an external router’s routing table. If unwanted traffic is reaching VSE, then this situation should be investigated and corrected.
- See the *TCP/IP FOR VSE Installation Guide* for information about using GATEWAY ON in conjunction with defining multiple TCP/IP FOR VSE stacks that share a network adapter.

Related Commands

DEFINE ADAPTER

Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ALTIP

Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE LINK

Creates a link between TCP/IP FOR VSE and a network or a directly connected stack.

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

QUERY HOME
Displays all IP addresses in the “Home Address” table.

QUERY STATISTICS
Displays a summary of stack-related information.

QUERY OPTIONS
Displays the current values of modifiable parameters.

IBBLOK

This command allows you to monitor and control the use of Internet Buffer Blocks (IBBLOKs).

Syntax

```
IBBLok [{RELEase|RESET|CLear}]
        [,{SIZE=num|,MTU=num}] [,FREE=num]
        [,STORage=num|PERcent=percent]
```

Arguments

(Null Arguments)
If no arguments are specified, the current values and statistics are displayed.

RELEase
Causes all IBBLOKs that are not currently in use to be freed.

RESET
Causes statistics on IBBLOK usage to be reset.

CLear
Causes previously made IBBLOK settings to be cleared.

SIZE=
Mutually exclusive with MTU=. This is used to select a specifically sized IBBLOK to be referenced by the FREE= parameter.

MTU=
Mutually exclusive with SIZE=. This is used to select a specifically sized IBBLOK by the MTU size of the data it can contain, to be referenced by the FREE= parameter.

FREE=
Used in conjunction with either the SIZE= or MTU= parameter, FREE specifies the maximum number of IBBLOKs of a particular size that will be retained when empty. Any empty IBBLOKs in excess of this value will be freed automatically. Values may range from 0 through 1000.

Note that IBBLOKs are sized in increments of 128 bytes.

STORage=
Mutually exclusive with PERCENT=, this parameter allows you to set a maximum size value for IBBLOKs. Be careful not to under-specify this value because unavailability of IBBLOKs will cause connection failure. Values may range from 20 through 1,000,000.

PERcent=
Mutually exclusive with STORAGE=, this parameter allows you to specify the percent of partition 31-bit GETVIS to be allowed for IBBLOK allocation. Values may range from 5 through 100.

Example

See the QUERY IBBLOKS command for an explanation of the message fields in this example.

```
 ibblok
IPN253I << TCP/IP IBBLOK Activity >>
IPN896I Counts - Current: 47 Peak: 86 Failed: 0
IPN897I Space - Current: 36k Peak: 109k Max: 51,092k (127%)
IPN889I Size: 384 MTU: 64 Free: 24/50 Hits: 4,465 Misses: 24
IPN889I Size: 512 MTU: 192 Free: 4/20 Hits: 569 Misses: 4
IPN889I Size: 640 MTU: 320 Free: 2/20 Hits: 54 Misses: 2
IPN889I Size: 768 MTU: 448 Free: 1/20 Hits: 24 Misses: 1
IPN889I Size: 896 MTU: 576 Free: 1/20 Hits: 12 Misses: 1
IPN889I Size: 1,024 MTU: 704 Free: 2/20 Hits: 20 Misses: 2
IPN889I Size: 1,152 MTU: 832 Free: 1/20 Hits: 32 Misses: 1
IPN889I Size: 1,280 MTU: 960 Free: 3/20 Hits: 32 Misses: 3
IPN889I Size: 1,408 MTU: 1,088 Free: 1/20 Hits: 31 Misses: 1
IPN889I Size: 1,536 MTU: 1,216 Free: 1/20 Hits: 32 Misses: 1
IPN889I Size: 1,664 MTU: 1,344 Free: 1/20 Hits: 7 Misses: 1
IPN889I Size: 1,792 MTU: 1,472 Free: 1/20 Hits: 1 Misses: 1
IPN889I Size: 1,920 MTU: 1,600 Free: 5/5 Hits: 2,892 Misses: 2,679
IPN889I Size: 64K+ MTU: -- Free: 0/0 Hits: 0 Misses: 0
```

Related Commands

QUERY IBBLOKS
Displays IBBLOK settings and statistics.

QUERY STOR
Displays detailed information on memory use.

INCLUDE

The INCLUDE command directs TCP/IP FOR VSE to fetch a member from a VSE library and execute the contents as if they were included in the initialization member.

Syntax

INCLude *member* [,DELAY]

Arguments

member

Specifies the member name containing the commands to be executed. The library and sublibrary are the same as the ones that contain the initialization member. A member type of '.L' is appended to the name. The member is normally processed before the command that follows the INCLUDE.

DELAY

Specifies that the member's contents are not to be executed until after TCP/IP FOR VSE initialization is complete and the dispatching engine is running.

Example

```
 include newroute
IPN397I Loading command deck NEWROUTE
IPN398I Command deck NEWROUTE has been completely loaded
IPN179I Direct Route 192.169.8.7= Net: 43272 Subnet: 0 Host: 7
IPN179I Direct Route 192.169.8.9= Net: 43272 Subnet: 0 Host: 9
```

Notes

The following notes apply to this command:

- It may be convenient to keep sets of like commands (such as DEFINE USER commands) in separate members and then execute them with the INCLUDE command.
- The INCLUDE command operates like the EXECUTE command.
- When an INCLUDE statement is encountered in a member, the processing of that member is suspended until the included member has been processed (unless DELAY is specified).
- During TCP/IP FOR VSE initialization, all configuration commands are read and processed before the dispatching engine is engaged. Using the DELAY parameter permits you to provide commands (such as PING) that are delayed until after initialization is complete and the engine has engaged.

Related Commands

EXECUTE

Executes an operator command script.

IPSTAT

The IPSTAT command controls whether an ISBLOK is allocated for every IP address that accesses the stack. Each ISBLOK keeps statistics for an IP address.

The Firewall Shield feature uses fields in the ISBLOK, so if the firewall is being used, IPSTAT cannot be turned off. See chapter 5, "Firewall Shield," in the *TCP/IP FOR VSE Optional Features Guide* for more information on this feature.

Syntax

IPSTAT {ON|OFF}

Arguments

ON
Enables allocating ISBLOKs. This is the default.

OFF
Disables allocating ISBLOKs. This setting does not take effect unless the firewall has first been disabled by the FIREWALL OFF command.

Example

```
 ipstat on
IPN268I IPSTAT now set to ON
```

Notes

This command is useful when testing the Firewall Shield feature.

The OFF setting affects the IP address statistics reported by the QUERY IPSTAT command.

Related Commands

QUERY IPSTAT
Displays IP address statistics.

FIREWALL
Controls and monitors the Firewall Shield optional feature.

ISOLATION

The ISOLATION command prevents remote hosts from establishing connections with TCP/IP FOR VSE daemons or applications.

Syntax

ISOLAtion {ON|OFF}

Arguments

ON
This setting prevents all inbound connection requests. Outbound requests continue to function.

OFF
This setting permits inbound connection requests to be made.

Example

```
 isolation on
IPN268I ISOLATION now set to ON
```

Notes

This feature is useful only when you absolutely do not want any remote access to your system.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

LISTIDCAMS

The LISTIDCAMS command is used with VSAMCAT. It controls whether the results of DELETE/DEFINE CLUSTER operations from a calling IDCAMS are output to SYSLST.

Syntax

LISTIDCAMS {ON|OFF}

Arguments

ON
The DELETE/DEFINE CLUSTER results are output to SYSLST. This is the default.

OFF
Results are not output to SYSLST.

Exposition

The VSAMCAT driver invokes IDCAMS to define a “cluster” (member definition) before it can write to it. It also invokes IDCAMS to delete a VSAMCAT member from a catalog. By default, when run as a batch job, IDCAMS sends its output to SYSLST so that you can verify it in case there were any problems. When another program calls IDCAMS, however, it can choose to NOT output to SYSLST. It is usually good to output to the TCP/IP’s SYSLST in case there are any problems that need to be corrected. But if this is not a concern, then the output to SYSLST can be disabled by setting LISTIDCAMS to OFF.

Example

```
 LISTIDCAMS OFF
IPN268I LISTIDCAMS now set to Off
```

LOCAL_DLBL

The LOCAL_DLBL command controls whether remote FTP users can specify VSE files using a local DLBL. Files that are specified using a percent character (%) and a DLBL are referred to as autonomous files. Such files are not defined to TCP/IP FOR VSE and do not have a public name.

This command controls access to autonomous files for all internal FTP servers (daemons) created using DEFINE FTPD. It does not affect the operation of external servers that were configured using FTPBATCH.

Note: This command is supported for compatibility with earlier releases. The DYNFILE= parameter on DEFINE FTPD and FTPBATCH provides the same control for individual daemons and should be used instead.

For details on the syntax used to specify autonomous files, see the *TCP/IP FOR VSE User Guide*, chapter 2, "FTP," subsection "VSE File Names."

Syntax

```
LOCAL_Dlbl {ON | OFF}
```

Arguments

ON
Remote FTP users are permitted to bypass the TCP/IP FOR VSE file system and transfer VSE files using autonomous file syntax. This is the default.

OFF
Remote FTP users are blocked from specifying autonomous files and can only specify files that have been defined to TCP/IP FOR VSE using DEFINE FILE. This setting overrides the default DYNFILE=YES setting in DEFINE FTPD.

Example

```
128 LOCAL_DLBL OFF
F8 0106 IPN268I LOCAL_DLBL now set to OFF
```

Related Commands

DEFINE FILE
Defines a file to the TCP/IP FOR VSE file system.

DEFINE FTPD
Configures an internal FTP server (daemon).

QUERY OPTIONS
Displays the current values of modifiable parameters.

MESSAGE (or MSG)

The MESSAGE command enables you to suppress console traffic by message number.

Note: MSG is a synonym of MESSAGE on TCP/IP FOR VSE. Do not confuse this command with the IBM "MSG *partition-id*" command.

Syntax

```
MESsage MSGID=msgid [Console={Yes|No}] [Log={Yes|No}]
        [SCROLlable={Yes|No}]
```

Arguments

MSGid=
Identifies the message whose display characteristics are to be altered. The message identifier is specified without the trailing importance indicator.

Console=
Controls message display on the console.

Yes
The message will be displayed on the console.

No
The message will not be displayed on the console.

Log=
Controls message recording in the printed log(s).

Yes
The message will be printed in the log.

No
The message will not be printed in the log.

SCROLlable=
Controls whether a message is displayed in scrollable mode on the console.

Yes
The message will be scrollable.

No
The message will not be scrollable. Be extremely careful because non-scrollable messages must be manually deleted. Once the console screen fills with non-scrollable messages, processing will cease until they are deleted.

Example

```
 message msgid=tel936,console=no
IPN801I Message suppression updated for TEL936

 message msgid=tel936,console=yes
IPN801I Message suppression updated for TEL936
```

Notes

The following notes apply to this command:

- Be careful when suppressing messages. Make sure you absolutely do not want to see the message before you use the facility.
- If you have a console automation product, you may want to specify message suppression criteria using that facility so that all message suppression originates in the same place.

Related Commands

DEFINE LOG
Creates a system log file.

MODIFY LOG
Changes the characteristics of a system log file.

MODIFY CONSOLE

This command allows you to control how messages are displayed on the operator console. It is a shortcut for MODIFY LOG,ID=CONSOLE.

Syntax

MODify CONsole [,LINELength=*num*]
[,TIMEstamp={Left|Right|None}] [,*routes*]

Arguments

LINELength=
This value indicates the maximum line length to be written. When a line exceeds this length, it is broken at a blank and a continuation character (>) is appended. Valid values are 40 through 132.

TIMEstamp=
Valid values: Left, Right, None. This value controls where the date and timestamp are placed on each line.

routes
Optional values indicate message levels that are added to or subtracted from the list of types to be logged. Note that the list is processed from left to right, so specifying "ALL NODIAG" ensures that everything but diagnostic messages are written to this log.

[NO]CRITical
Messages of a critical nature are logged/suppressed.

[NO]VITAL
Messages of vital nature are logged/suppressed.

[NO]WARNing
Warning messages are logged/suppressed.

[NO]IMPORTant
Important messages are logged/suppressed.

[NO]INFOrmational
General informational messages are logged/suppressed.

[NO]RESPonse
Responses to commands are logged/suppressed.

[NO]DIAGnose
Messages produced in response to the DIAGNOSE command are logged/suppressed. Note that these messages can be both voluminous and rapid.

[NO]SECurity
Messages related to security are logged/suppressed.

ALL
All message types are logged.

NONE
No messages will be logged.

Example

```
 modify console,none,resp,security

 query logs
IPN253I << TCP/IP Console Logs >>
IPT240I CONSOLE on CONSOLE
IPT241I Line Length: 65 Lines per page: 0 Timestamp: None
IPT242I Total lines: 0 Total pages: 0
IPT243I Logging: Response Security
```

Notes

Be careful that all messages are recorded somewhere. Otherwise, when a problem occurs, it may be difficult to obtain the necessary diagnostic information.

Related Commands

DEFINE LOG
Creates a system log file.

MESSAGE
Controls message suppression.

MODIFY LOG
Changes characteristics of a system log file. MODIFY CONSOLE is a shortcut for MODIFY LOG,ID=CONSOLE.

QUERY LOGS
Displays available consoles and logs along with their properties.

MODIFY FILE

The MODIFY FILE command changes the characteristics of a previously defined entry in the file system.

Syntax

```
MODify FILE PUBlic=pubname
        [,TYPe={ESDS|KSDS|SAM|LIBrary|ICCF|POWer|
        VSAMCAT|VTOC|HFS|DSPACE|BIM-EDIT|CONDOR|FALCON|
        VOLLIE|TAPE}] [,DLBL=name8] [,DRIVer=member]
        [,ALLOWsite={Yes|No} [,READonly={Yes|No}]
        [,CC={Yes|No}] [,TRcc={YES|NO}]
        [,CRlf={Yes|No}] [,RECFm=F|FB|V|VB|S|SV|SU|SB]
        [,LRECL=num] [,BLKsize=num] [,GID=snum]
        [,UID=snum] [,TRANslate=name16]
        [,SITE={Yes|No}] [,DBLOCKS=num] [,EXT=name8]
        [VOLid=volser] [,CIPHER={NULL-SHA1|
        SDESCBC-NULL|SDESCBC-SHA1|TDESCBC-NULL|
        TDESCBC-SHA1|AES128C-NULL|AES128C-SHA1|
        AES192C-NULL|AES192C-SHA1|AES256C-NULL|
        AES256C-SHA1| KEYMASTER}][,CIPHERKEY=CIALHFSK]
```

Arguments

The specifiable parameters closely follow those of DEFINE FILE. In all cases, if a parameter is omitted, the value is left unchanged.

TYPE=
Specifies the type of dataset.

ESDS
A VSAM ESDS dataset. FTP read requests result in the entire file being transmitted.

KSDS
A VSAM KSDS dataset. FTP read requests result in the entire file being transmitted. Write requests are processed as VSAM INSERT operations.

SAM
A Sequential Access Method dataset.

LIBrary
A VSE Library. Performing a directory listing on the public name returns a list of the sub libraries. Further qualifying the public name with a sublibrary name (such as *public.name.sublib*) returns a list of the members. FTP read requests retrieve the contents of a member. Write requests create or replace the contents of a member.

ICCF

An ICCF library structure. Directory listing is not supported. To retrieve data, use a fully qualified name consisting of the public name, the library number, and the member name (such as *pubname.num.member*). Writing to an ICCF library is not supported.

Note that to access an ICCF library using FTP, your FTP userid and password MUST be identical to your ICCF userid and password.

POWer

A POWER queue. To retrieve information from POWER, specify the public name qualified with LST, RDR, or PUN, followed by class and then by job name. Two additional qualifiers, job number and job suffix, may also be appended as needed. Writing to POWER creates a file on the specified queue.

VSAMCAT

A VSAM catalog. A directory list returns the names of all files in that catalog. Supported operations include reading, writing, creating, deleting, renaming, and appending. A full list of supported operations is provided in the *TCP/IP FOR VSE User Guide* in the "FTP" chapter.

VTOC

An entire VSE volume. You can perform limited operations on a file defined with TYPE=VTOC. A list of operations supported by TYPE=VTOC is contained in the *TCP/IP FOR VSE Installation Guide*.

HFS

CSI International's Hierarchical File System. This file type supports a PC or Linux-type file system with multiple subdirectory levels and long file names.

DSPACE

Causes a dataspace to be allocated and used as a file. See the *TCP/IP FOR VSE Installation Guide* for more information.

BIM-EDIT

CSI International's BIM-EDIT product. This type requires a BIM-EDIT license, and you must link the .OBJ file to make the I/O driver module usable. You must use DEFINE FILEIO to load the BIM-EDIT I/O driver module into storage.

CONDOR, FALCON, VOLLIE

File types associated with non-CSI products. For each type, you must obtain a vendor-provided I/O driver module and then install it. See "CA Vollie™ File Example," page 64, for more information.

TAPE
A tape drive file. You must use DEFINE FILEIO to load the associated TAPE I/O driver module into storage.

DLBL=
A DLBL name. This specifies the DLBL statement to be used to open the file. It must be accessible to the TCP/IP FOR VSE partition. Files of type "POWER" do not require or use a DLBL.

DRIVer=
If specified, this phase is loaded from the library search list when the dataset's driver is initialized. In general, you should let this value default to the phase provided for the file type. If the phase you need is not listed by QUERY FILEIO, you must use DEFINE FILEIO to load the phase before you can specify it in DEFINE FILE.

PUBlic=
The unique name that identifies this dataset to users. This is the name assigned to the file by the DEFINE FILE command.

READonly=
This setting can be used to selectively limit FTP access to this file.

No
FTP users may write to and update this file, subject to other security procedures.

Yes
Users do NOT have write or update access to this file, regardless of other security procedures.

ALLOWsite=
FTP SITE commands that are specific to a file type (for example, SITE PALTER) are passed to the file I/O driver of the file currently being accessed. Coding NO on this parameter overrides whatever may be specified in the FTPD definition and can be used to prevent unauthorized commands from being issued.

No
FTP users may NOT pass SITE commands to the file I/O driver associated with this file.

Yes
FTP users can pass SITE commands to the file I/O driver associated with this file.

EXT=
For TYPE=VSAMCAT, specifies a suffix to be appended to the file name. For example, EXT=.TXT would ensure that references to the files in the catalog would all be assumed to be "text."

CC=

Provides a default value for the FTP SITE CC command. If not specified, the value is supplied by client or user.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

Yes

The first byte of each record is assumed to be a carriage control byte, and this byte is retained as part of the data.

No

During downloads (from VSE), the first byte of each record is discarded. During uploads (to VSE), a blank character is prefixed to each record.

TRcc=

Provides a default value for the FTP SITE TRCC command. If not specified, the value is supplied by the client or user.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

No

No special processing is performed to simulate carriage control.

Yes

ANSI carriage control codes (+, 0, -, 1) cause simulation of the CC character. Forms control characters (CR, LF, FF) are added to the output records.

CRlf=

Provides a default value for CRLF record delimiter processing. If not specified, the value is supplied by client or user.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

Yes

Each record uploaded to VSE must be ended by an appropriate delimiter, generally CR/LF. On download, the appropriate delimiter is added to each record.

No

No delimiter(s) is added on download and no delimiter(s) is expected on upload. For incoming records, the data stream is divided based on the LRECL value.

RECFm=

Provides a default value for the FTP SITE RECFM command.

Specify a record format of F, FB, V, VB, or S. (String format is valid only with Librarian files.) This setting does NOT override the value specified on the DLBL definition. If the value is not specified on either DLBL or DEFINE FILE, then it can be supplied using a SITE command.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

LRECL=

Provides a default value for the FTP SITE LRECL command.

This value is the logical record length. It is a numeric and must be consistent with the value used when the file was created. The default is 80. See the tables that follow for selection information.

This value is not obtained from DLBL information and must be provided, either by DEFINE FILE or with an FTP SITE command.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

BLKsize=

Provides a default value for the FTP SITE BLKSIZE command.

This value is the block size used in the dataset. This information must be consistent with the value used when the file was created. There is no default. See the tables that follow for selection information.

This value is not obtained from DLBL information and must be provided, either by DEFINE FILE or with an FTP SITE command.

Note: This setting can be overridden. See note in the section "RECFM, LRECL, BLKSIZE" below for more information.

TRANslate=

Provides a default value for the FTP SITE TRANSLATE command when the file must be translated between ASCII and EBCDIC mode. If omitted, the default is the TCP/IP FOR VSE default translate table.

To be valid, the named translation table must already be loaded at the time the file is opened, rather than when DEFINE FILE is issued.

The DEFINE TRANSLATION command explains the definition and loading of translate tables.

SITE=

SITE commands are processed hierarchically. The FTP daemon handles most commands itself. Any command that is not understood by the daemon is passed to the file I/O driver of the currently selected file, as determined by the last-issued CD command.

No

FTP SITE commands that are NOT recognized by the FTP daemon are not passed to the file-specific driver routine.

Yes

FTP SITE commands that are not recognized by the FTP daemon are passed to the file I/O driver routine for possible processing.

GID=

Signed numeric, -9999999 through +9999999.

Defines this file as part of a group. TCP/IP FOR VSE does not use this field but passes it to the TCP/IP FOR VSE security exit. A GID value may also be assigned with DEFINE USER.

UID=

Signed numeric, -9999999 through +9999999.

Associates this file with a UNIX-style user ID. TCP/IP FOR VSE passes this field to the TCP/IP FOR VSE security exit. A UID value may also be assigned with DEFINE USER.

DBLOCKS=

For TYPE=DSPACE, specifies the space to be allocated to the virtual file.

VOLid=

For TYPE=VTOC, specifies the volume ID of the disk.

CIPHER=

For TYPE=HFS, indicates that files are to be stored in an encrypted form using the specified method. If this parameter is omitted, no encryption or decryption is performed.

CIPHERKEY=

For encrypted TYPE=HFS files, this keyword provides the name of the phase that contains information on the encryption keys. The default phase is shipped with TCP/IP and contains sample keys. You should create your own phase with your own keys before using this feature in a production environment. See the *TCP/IP FOR VSE Programmer's Guide* for more information.

Example

```
 modify file public=ijsysrs,readonly=yes
IPN163I Adjusted Dataset: IJSYSRS
```

RECFM, LRECL, BLKSIZE

Acceptable values for RECFM, LRECL, and BLKSIZE depend on how you originally defined the file, the access method you use, and the mode of access (reading or writing). 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: If EXTTYPES processing is NOT in effect for the file and SITELAST=YES in FTP, then SITE commands take precedence over the following DEFINE FILE parameters: BLKSIZE, CC, CRLF, LRECL, RECFM, TRCC. For details, see the *TCP/IP FOR VSE Installation Guide*, chapter 6, "Configuring FTP Clients and Daemons."

The sections that follow contain tables that show acceptable values for RECFM, LRECL, and BLKSIZE. "Input" headings indicate that you are reading from disk, and "Output" headings indicate that you are writing to disk. These terms do not indicate whether you are using the FTP client or the FTP daemon.

Sequential Disk File and VSAM-Managed SAM File Considerations

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

- 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.
- Although IBM's VSAM-managed SAM files will appear to be ESDS files when performing an IDCAMS LISTCAT of the VSAM catalog, it is recommended that you read them as SAM files rather than ESDS files. This is because the IBM routines that perform the SAM output of the file often will not correctly update the catalog after the file is closed. This may result in incomplete transfers when using certain graphic FTP clients.
- RECFM SU is interpreted as a spanned unblocked file. RECFM SU is interpreted as a spanned blocked file.

The following table shows appropriate values for this file type.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
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
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

VSAMCAT File Considerations

For VSAMCAT files, note the following information.

- For output files, if the file does not already exist, then the SITE command will establish the parameter values used in the DEFINE CLUSTER command that will be passed as a subtask to the IBM IDCAMS utility prior to writing to the file.

  If the output file already exists, then the SITE commands will be ignored and the IDCAMS utility will not be invoked. This means that the SITE commands you use must match the expected attributes.

- For input files, your SITE commands need not match the attributes of the existing file. This is true for all VSAM file types.

- Fixed-length records are padded if necessary when writing to the VSE/POWER spool. When padding occurs, text files are padded with blanks and binary files are padded with binary zeros.

- If you use the "blocked" type (VB or FB) for output, then the "(*nnnn*)" parameter of the DEFINE CLUSTER RECFM command

passed to the IDCAMS utility will be provided, where *nnnn* is the record length. Otherwise, use the "F" or "V" record format.

The following table shows appropriate values.

	Input		Output	
recfm	***lrecl***	***blksize***	***lrecl***	***blksize***
F	record size	N/A	N/A[1]	N/A[1]
V	maximum record size	N/A	N/A[1]	N/A[1]

[1] Depends on whether the output file exists. SITE command parameters are passed to the IDCAMS utility only upon file creation.

TAPE File Considerations

For TAPE files, note the following information:

- Fixed-length records are padded when necessary. When padding occurs, tape 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.

The following table shows appropriate values for TAPE files.

	Input		Output	
recfm	***lrecl***	***blksize***	***lrecl***	***blksize***
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
UN	maximum record size	N/A	maximum record size	N/A

VSE/POWER File Considerations

For POWER files, note the following:

- 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 files.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	Record size	N/A	N/A	N/A
V	Maximum record size	N/A	N/A	N/A

ESDS, KSDS File Considerations

For ESDS and KSDS VSAM files, the following table shows appropriate values:

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	Record size	N/A	N/A	N/A
V	Maximum record size	N/A	N/A	N/A

ICCF File Considerations

For ICCF files, note the following:

- The files are read only.
- The files always contain 80-byte records, regardless of specification.

The following table shows appropriate values for ICCF files.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	80	N/A	80	N/A

Library File Considerations

For Library files, note the following:

- FTP of phases is not supported.
- Records in a string file consist 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.

The following table shows appropriate values for library files.

RECFM	Input LRECL	Input BLKSIZE	Output LRECL	Output BLKSIZE
F	80	N/A	80	N/A
SV	Variable	N/A	Variable	N/A
S	N/A	N/A	N/A	N/A

Related Commands

AUTOLOAD
Determines automatic loading of file I/O drivers when files are defined.

DEFINE FILE
Defines a file in the TCP/IP FOR VSE file system and associates it with a file I/O driver.

DEFINE FILEIO
Loads a file I/O driver phase for a file type.

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

DELETE FILE
Removes a file from the TCP/IP FOR VSE file system

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY FILEIO
Displays the status of the file I/O driver programs.

SECURITY
Controls TCP/IP FOR VSE security functions.

MODIFY LOG

Messages are written to the console and one or more log files. This command allows you to change the characteristics of a log.

Syntax

MODify LOG ID=*id* [,LINELength=*num*]
[,TIMEstamp={Left|Right|None}] [,*routes*]

Arguments

ID=
The ID that identifies the log to be modified. The values CONSOLE and SYSLST are automatically defined. **Note:** MODIFY CONSOLE is a shortcut for MODIFY LOG,ID=CONSOLE.

LINELength=
This value indicates the maximum line length to be written. When a line exceeds this length, it is broken at a blank and a continuation character (>) is appended. Valid values are 40 through 132.

TIMEstamp=
Valid values: Left, Right, None. This value controls where the date and timestamp are placed on each line.

routes
Optional values indicate message levels that added to or subtracted from the list of types to be logged. Note that the list is processed from left to right, so specifying "ALL NODIAG" ensures that everything but diagnostic messages are written to this log.

[NO]CRITical
Messages of a critical nature are logged/suppressed.

[NO]VITAL
Messages of vital nature are logged/suppressed.

[NO]WARNing
Warning messages are logged/suppressed.

[NO]IMPORTant
Important messages are logged/suppressed.

[NO]INFOrmational
General informational message are logged/suppressed.

[NO]RESPonse
Responses to commands are logged/suppressed.

[NO]DIAGnose
Messages produced in response to the DIAGNOSE command logged/suppressed. Note that these messages can be both voluminous and rapid.

[NO]SECurity
Messages related to security are logged/suppressed.

ALL
All message types are logged.

NONE
No messages are logged.

Example

```
 define log,id=sys007,type=printer,lu=sys007,linel=132
 modify log,id=syslst,all,noresp,nosecurity
 modify log,id=console,all,nodiag
 modify log,id=sys007,none,resp,security
 query logs
IPN253I << TCP/IP Console Logs >>
IPT240I CONSOLE on CONSOLE
IPT241I Line Length: 65 Lines per page: 0 Timestamp: None
IPT242I Total lines: 0 Total pages: 0
IPT243I Logging: Critical Vital Warning Important Info Response Secur
IPT240I SYSLST on SYSLST
IPT241I Line Length: 132 Lines per page: 86 Timestamp: Left
IPT242I Total lines: 300 Total pages: 0
IPT243I Logging: Critical Vital Warning Important Info Diag
IPT240I SYS007 on SYS007
IPT241I Line Length: 132 Lines per page: 0 Timestamp: None
IPT242I Total lines: 44 Total pages: 0
IPT243I Logging: Response Security
```

Notes

The following notes apply to this command:

- Make sure that all messages are recorded somewhere. Otherwise, when a problem occurs, it may be difficult to obtain the necessary diagnostic information.
- The example above shows how to suppress “sensitive” messages from SYSLST while establishing a security audit trail in a separate, protected file.

Related Commands

DEFINE LOG
Creates a system log file.

MESSAGE
Controls message suppression.

MODIFY CONSOLE
This is a shortcut for the MODIFY LOG,ID=CONSOLE command.

QUERY LOGS
Displays available consoles and logs, along with their properties.

MODIFY ROUTE

The MODIFY ROUTE command changes values assigned to an already-defined route entry. You can also use this command to change the order of existing route entries.

Syntax

```
MODify ROUTe ID=id [,LINKid=name16] [,NUMber=0]
       [,IPaddr=ip4addr] [,GATEway=ip4addr1]
       [,AFTer=id] [,MTU=num] [,MSS=num]
       [,CRETran=msec] [,DRETran=msec]
       [,FIXRetran={Yes|No}] [,MINRetran=msec]
       [,MAXRetran=msec] [,PULse=sec] [,WINdow=num]
       [,RPAuse=msec] [,RETRY=num]
```

Arguments

LINKid=

This is the same value that is specified in the ID= parameter of the DEFINE LINK command that will be the destination for this route table entry.

NUMber=

For links with adapters, this directs the route to the specific numbered adapter. The default is “0”. This parameter is required if the NUMBER= parameter of the target DEFINE ADAPTER is not “0”.

IPaddr=

A TCP/IP network address or “zero host” address. All messages destined for this address are sent on the associated link.

GATEway=

The full network address of a gateway to other networks. A match on this table entry causes the data packet to be sent to the specified gateway.

AFTer=

The value of the name parameter identifying the DEFINE ROUTE statement after which this one is to be moved. If this parameter is omitted, the route entry’s position in the table is left unchanged. A special value of “TOP” can be coded to cause the route statement to be moved to the top of the list.

Placement in the table is very important because the look-up procedure is a top-to-bottom search for the first-match (except for “0.0.0.0,” which is always matched last).

MTU=
The MTU value to be used with this route. This value is meaningful only if it is less than the value specified by the target DEFINE LINE or DEFINE ADAPTER. Typically, this parameter only controls the size of outbound datagrams.

MSS=
The Maximum Segment Size to be used with this route. MSS is conveyed to the remote host during OPEN processing and control the size of datagrams being constructed by the remote host. The MSS specification is always reduced to MTU-40.

CRETran=
This specifies the number of milliseconds that TCP/IP FOR VSE will wait for an ACK in response to a connection request (SYN). Once this interval has elapsed, retransmission mode is entered.

DRETran=
This specifies the number of milliseconds that TCP/IP FOR VSE will wait for an ACK in response to a datagram transmission on an established connection. Once this interval has elapsed, retransmission mode is entered.

FIXRetran=
Yes
The values specified for DRETRAN= and RPAUSE= remain constant for the duration of the connection.

No
The values for DRETRAN= and RPAUSE= start out as specified but are dynamically adjusted as the network response is analyzed.

MINRetran=
If FIXRETRAN=NO is specified, this is the minimum time (in milliseconds) that can be dynamically assigned to DRETRAN.

MAXRetran=
If FIXRETRAN=NO is specified, this is the maximum time (in milliseconds) that can be dynamically assigned to DRETRAN.

RPAuse=
Once retransmit mode has been entered, this is the time (in milliseconds) that will elapse between retransmission attempts.

RETRY=
This parameter specifies the number of times an unacknowledged datagram will be retransmitted before the connection is considered to be dead.

PULse=

This specifies how long (in seconds) that a connection can be idle (no traffic of any kind) before a probe is made to determine whether the remote host is still active.

WINdow=

This value indicates the desired size of the Receive Window.

Example

```
 query routes
IPN253I << TCP/IP Routes >>
IPN448I ID: DEFAULT Link ID: LINK3172, 0
IPN449I IP Address: 0.0.0.0 Mask: 255.255.255.0
IPN450I Net: -- Subnet: -- Host: --
IPN448I ID: *Internal Link ID: *Internal
IPN449I IP Address: 127.0.0.0 Mask: 255.0.0.0
IPN450I Net: 127.0.0.0 Subnet: -- Host: --
IPN448I ID: *Internal001 Link ID: *Internal
IPN449I IP Address: 192.168.1.161 Mask: 255.255.255.0
IPN450I Net: 192.168.1.0 Subnet: -- Host: 0.0.0.161
IPN448I ID: LOCAL Link ID: LINK3172, 0
IPN449I IP Address: 192.168.1.0 Mask: 255.255.255.0
IPN450I Net: 192.168.1.0 Subnet: -- Host: --

 modify route,id=default,after=local

 query routes
IPN253I << TCP/IP Routes >>
IPN448I ID: *Internal Link ID: *Internal
IPN449I IP Address: 127.0.0.0 Mask: 255.0.0.0
IPN450I  Net: 127.0.0.0 Subnet: -- Host: --
IPN448I ID: *Internal001 Link ID: *Internal
IPN449I IP Address: 192.168.1.161 Mask: 255.255.255.0
IPN450I Net: 192.168.1.0 Subnet: -- Host: 0.0.0.161
IPN448I ID: LOCAL Link ID: LINK3172, 0
IPN449I IP Address: 192.168.1.0 Mask: 255.255.255.0
IPN450I  Net: 192.168.1.0 Subnet: -- Host: --
IPN448I ID: DEFAULT Link ID: LINK3172, 0
IPN449I IP Address: 0.0.0.0 Mask: 255.255.255.0
IPN450I Net: -- Subnet: -- Host: --
```

Notes

The following notes apply to this command:

- TCP/IP FOR VSE searches the route statements in the same order they are entered (except for an entry with an all-zero IP address). Using the AFTER= parameter ensures proper sequencing of the route table.
- To examine the search order, use the QUERY ROUTES command. This displays the route table in search order (except for entries with an all-zero IP address).

- A route statement with an all-zero IP address is matched only after all other entries have been tested (in order).
- Once a route statement is matched by IP address, the designated link is checked for availability. If the link cannot be used, the search continues with the next route entry.
- When a DELETE LINK command is issued because
  - — you want to disable a link,
  - — you want to change one or more attributes of a link, or
  - — you want to recover from a link that was deactivated by a hardware command,

  all DEFINE ROUTE statements referring to that link are automatically deleted.
- If you want to redefine a link you deleted, you first need to wait for the DELETE command to complete and then confirm that the LINK no longer exists by issuing a QUERY LINKS command. After that, you need to define the link and then reestablish the routes using the DEFINE ROUTE command.

  For example, after issuing a DELETE LINK, you can

  1. Use QUERY LINKS to confirm that the link is gone
  2. Use DEFINE LINK to reestablish the link
  3. Use DEFINE ROUTE to reestablish the routes.

  Alternatively, to reestablish a link you deleted, you can simply cycle the TCPIP FOR VSE stack.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ALTIP
Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE MASK
Creates a subnet mask for a particular network.

DEFINE ROUTE
Adds an entry to the TCP/IP FOR VSE routing table.

DELETE ALTIP
Removes an alternate home address.

DELETE LINK
Removes a link between TCP/IP FOR VSE and a network or a directly connected stack.

DELETE ROUTE
Removes an entry from the network routing table.

DISCOVER
Determines the “best” MTU size to a remote host.

GATEWAY
Controls forwarding of datagrams not intended for the VSE stack.

QUERY ARPS
Displays the current content of the ARP table.

QUERY LINKS
Displays the status of network links.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

QUERY OPTIONS
Displays the current values of modifiable parameters.

SET IPADDR
Establishes the default home address for the stack.

SET MASK
Establishes a default subnet mask.

SET MAX_SEGMENT
Controls the default setting for the inbound Maximum Segment Size.

SET PULSE_TIME
Controls the default setting for the interval between probes of inactive connections (PULSE= parameter on DEFINE ROUTE).

SET RETRANSMIT
Controls the length of the default interval before TCP/IP FOR VSE enters retransmit mode for unacknowledged data packets.

SET WINDOW
Controls the default setting of the TCP inbound window.

TRACERT
Displays each “hop” in a route along with the time needed to reach it.

PING

The PING command allows testing of a communication path without involving clients and servers on each end. The basic PING function is supported at a low level by most TCP/IP stacks.

Syntax

PING *host*

Arguments

host

Specifies the IP address of the host to be pinged. This may be an actual numeric address or a symbolic name that can be resolved to an IP address.

Example

```
 ping e-vse.com
TCP915I PINGING 012.182.034.227 (e-vse.com)
TCP910I PING 1 was successful, milliseconds: 00058.
TCP910I PING 2 was successful, milliseconds: 00072.
TCP910I PING 3 was successful, milliseconds: 00065.
TCP910I PING 4 was successful, milliseconds: 00072.
TCP910I PING 5 was successful, milliseconds: 00074.
TCP910I PING Complete
```

Notes

The following notes apply to this command:

- Each PING command causes five Ping operations to be attempted. The result of each attempt is displayed.
- The PING process fails based on a time-out method. Response messages may not be instantaneous.
- The PING command is extremely useful in determining the ability to reach a remote client.
- If the PING command fails and the remote host is functional, ensure that the proper DEFINE ROUTE commands have been issued.
- PING may also be issued with the CICS PING transaction and the batch PING client.
- Some gateways will not forward traffic to VSE until they have received traffic from VSE. PING is a useful method for providing this initial traffic.
- If the PING command fails and you are using a symbolic name that is being resolved by an external domain name server, make sure that you can successfully PING the domain name server.

Related Commands

PING_MESSAGE

Controls the “ping request received” console message.

TRACERT

Displays each “hop” in a route along with the time needed to reach it.

PING_MESSAGE

The PING_MESSAGE command controls the production of console messages produced by TCP/IP FOR VSE each time a PING request is received from the network.

Syntax

```
PING_Message {ON|OFF}
```

Arguments

ON
Enables console notification each time an inbound ping request is received. This is the default.

OFF
No message is displayed on the console when an inbound ping request is detected.

Example

```
 ping_message on
IPN538I Ping Message has been set ON

IPC108I ICMP Echo request has been received from: 192.168.5.100
```

Notes

The following notes apply to this command:

- Receipt of a PING request does not mean that TCP/IP connections can be made. Making a connection always requires two-way communications. Before a connection is possible, PING must be successful in both directions.
- The Automatic Security Exit can be configured to prevent the stack from responding to PING requests.
- The SECURITY command can be used to pass incoming PING requests to the User Security Exit for evaluation before the stack responds.

Related Commands

ASECURITY
Configures the Automatic Security Exit

PING
Issues an ICMP Echo (PING) request.

SECURITY
Controls TCP/IP FOR VSE security functions.

PORTQUEUE

The PORTQUEUE command establishes, modifies, and disables port queuing.

This command can be executed only after TCP/IP FOR VSE has fully started. To run it from the TCP/IP FOR VSE initialization member (the initialization deck), use the command "INCLUDE *lib_member*,DELAY" and place the PORTQUEUE command in *lib_member*.

Note:

To use port queuing with BSD socket applications, QUEDMAX must be set to 0 (the default) in the $SOCKOPT options phase. For more information, see the listen( ) function description in the *TCP/IP FOR VSE Programmer's Guide*, chapter 2, "BSD Socket Interface."

Syntax

PORTQueue PORT=*port* [,TIMEOUT=*sec*] [,DEPTH=*num*]

Arguments

PORT=

The port number for which queuing is to be established or modified. Valid values range from 1 to 65535.

TIMEOUT=

This is how long a queued connection can wait for service before it is discarded. Valid values range from 1 to 60 seconds. If not specified, the stack assigns a value of its choice for a new specification and leaves existing values as found.

DEPTH=

This is the maximum number of connections that may be queued at any given time. Once this number is reached, additional incoming requests are refused. Valid values range from 0 (no queuing) to 100. If not specified, any existing value remains unchanged.

Example

```
 portqueue port=4099, timeout=5, depth=10
IPN405I Port queue values successfully set
```

Exposition

Standard TCP/IP processing is simple in concept. Each connection consists of series of interactions, as follows, between two applications:

1. Application A issues a passive OPEN (a listen).
2. Application B issues an active OPEN.
3. The applications issue SENDs and RECEIVEs as appropriate.
4. Each application issues a CLOSE.

In general, the application issuing the passive OPEN (the listen) is considered the server, and the application issuing the active OPEN is considered the client.

For example, a Web server issues its listen on the well-known port 80. When a browser (the client) wants to obtain a Web page, it issues an active OPEN to port 80 and sends its request.

Once the server fills the request, the connection is closed, and the server re-issues its listen to await the next client's request.

In practice, the difficulty with this process is coping with connection requests that occur when the server is already processing another request, that is, when there is no listen in effect. The rules governing the stack are explicit and require it to forcefully reject (RESET) any connection request that cannot be paired immediately with an existing listen connection.

The most common way of ensuring that all (most) connection requests are successful is to maintain multiple listen connections on the server for the same local port. As each of these passive OPENs completes, the server immediately replaces it with another connection. Processing of the established connections can overlap to any degree desired.

This programming is complex, however, and it still permits some requests to be lost due to processing delays at the application level. And it is even more difficult to provide the ability to change the queuing depth without requiring modifications to the program's code.

The PORTQUEUE facility avoids these problems and eliminates the need for most multi-threading in applications. To use it, you must first determine the port number on which connection requests are to be queued. For example, a web-server application would choose the well-known port 80.

Once the port is designated as eligible for queuing, incoming connection requests that cannot be paired with existing listen connections are assigned to a "blank" listen connection, which the stack provides automatically. This connection is negotiated by proxy with a closed window. When the server eventually issues a listen, the next opened-by-proxy connection is assigned, the window is opened to its normal value, and processing proceeds in the expected manner.

Related Commands

QUERY PORTQUEUE

Displays statistics associated with queued connection requests.

INCLUDE

Directs TCP/IP FOR VSE to fetch a member from a VSE library and execute the contents as if they were included in the initialization member.

PORTRANGE

The PORTRANGE command controls the range of local ports that are used for dynamic assignment.

Syntax

PORTRange LOw=*port_num*,HIgh=*port_num*

Arguments

LOw=

The lowest port number that will be dynamically allocated. The value must be at least 1024 lower than the one specified by HIGH=, and it must be in the range of 1024 through 65535.

HIgh=

The highest port number that will be dynamically allocated. The value must be at least 1024 higher than the one specified by LOW=, and it must be in the range of 4096 through 65535.

Example

```
 portrange low=32768,high=65535
IPN127I Port range changed to 32,768 65,535
```

Exposition

Server applications generally issue a passive OPEN (a listen) on a well-known port and then wait for an external client to connect. Because they need to be found easily, they explicitly specify the local port to be used. In the case of FTP, the local port assignment for a data connection is different.

Once an FTP client has established a control connection with each of two daemons, it must issue instructions to each daemon to enable them to open a data connection. The client instructs one daemon to issue a passive OPEN and then to return the port number to the client. The client next instructs the other daemon to issue an active OPEN to that port.

Typically, the passive FTP daemon asks the local stack to provide the next available port number. This allows the stack to ensure that proper time intervals elapse between port re-use. It also ensures that the address and port combination is unique.

Using PORTRANGE can ensure that the port number assigned is acceptable to firewall requirements. A site may be limited to a port range based on internal network specifications, and allocating ports outside of that range would cause intermittent failures of user-written programs.

Related Commands

DEFINE FTPD

Creates a File Transfer Protocol daemon.

QUERY CONNECTIONS

Displays the status of one or more connections.

QUERY ACTIVE

The QUERY ACTIVE command displays active Telnet sessions, active FTP sessions, active links, or all three.

Syntax

`Query ACTive [,SYSLST] TYPe={TELnetd|FTPd|LINK|ALL}`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`TYPe=`
Specifies the data to be displayed.

`TELnetd`
Displays all active Telnet daemons.

`FTPd`
Displays all active FTP daemons.

`LINK`
Displays all active links.

`ALL`
Displays all active daemons and links

Example

See the *TCP/IP FOR VSE Messages* for an explanation of each message.

```
 query active
IPN435I ID: FTP01 Port: 21 Driver: FTPDAEMN SSL: No
IPN874I Buffers: 2 Buffer size: 65536 bytes
IPN350I Current Status: Listen
IPN362I Maximum sessions: 5, Current: 1
IPN364I Userid: EWF connected from 192.168.1.66,1158
IPN365I Started at: 13:57:49 2015/03/08
IPN368I Last Command: LIST
IPN363I Last Command time: 13:57:51 2015/03/08
IPN368I Last reply: 226 Closing data connection
IPN363I Last reply time: 13:57:51 2015/03/08
IPN253I << TCP/IP ACTIVE LINKS >>
IPN440I ID: LINK3172 Type: 802.3 Adapter: 0
IPN438I MTU: 1500 IP Address: 192.168.1.161
IPN441I MAC address: 00:00:E2:90:C0:45
IPN350I Current Status: Active
IPN437I ID: LINK3172 Type: OSA2 Dev: (0032,0033)
IPN350I  Current Status: Active
IPN437I ID: *Internal Type: *Int Dev: (0000,0000)
IPN438I MTU: 32767 IP Address: 127.0.0.1
IPN350I Current Status: Active
```

Notes

The output of this command can be voluminous.

Related Commands

QUERY FTPDS
Displays the status of the File Transfer Protocol daemons.

QUERY LINKS
Displays the status of network links.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

QUERY ALL

The QUERY ALL command displays all available information.

Syntax

```
Query ALL [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
query all,syslst
```

Notes

The following notes apply to this command:

- The output of this command can be voluminous.
- All information displayed by QUERY ALL can be obtained through more specific query commands.
- The individual response lines comprising the output to QUERY ALL are documented in the TCP/IP FOR VSE Messages manual.

Related Commands

QUERY ALTIPS
Displays all alternate IP addresses.

QUERY ARPS
Displays the current content of the ARP table.

QUERY CGIS
Displays all currently available CGI programs.

QUERY CONNECTIONS
Displays the status of one or more connections.

QUERY DUMPOPTION
Displays options for TCP/IP-produced dumps.

QUERY EMAIL
Displays Email client settings. **Note**: Issuing the EMAIL command without any parameters produces the same results.

QUERY EVENTS
Displays the status of automation processing.

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY FILEIO
Displays the status of the file I/O driver programs.

QUERY FTPDS
Displays the status of the File Transfer Protocol daemons.

QUERY GPSDS
Displays the status of the General Print Server daemons.

QUERY HOME
Displays all IP addresses in the "home address" table.

QUERY HTTPDS
Displays the status of the Hypertext Transfer Protocol (Web server) daemons.

QUERY ISTATS
Displays statistics detailing internal stack functions.

QUERY LINKS
Displays the status of network links.

QUERY LPDS
Displays the status of Line Printer daemons.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY MENUS
Displays menus available for TN3270 use.

QUERY NAMES
Displays TCP/IP FOR VSE names and the values associated with them.

QUERY NTPDS
Displays status of NTP daemons.

QUERY OPENFILES
Displays a list a files that are currently open.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY PRODKEYS
Displays the TCP/IP FOR VSE product keys being used, the names of the licensed products, and the expiration dates of the keys.

QUERY PROGRAMS
Displays the program phases being used by TCP/IP FOR VSE, their characteristics, their memory locations, and the library from which each was loaded.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

QUERY SECURITY
Displays current security settings.

QUERY STATISTICS
Displays a summary of stack-related information.

QUERY SUSPENDED
Displays a list of suspended tasks.

QUERY TASKS
Displays a list of pseudo tasks.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

QUERY TRACES
Displays a list of currently running traces.

QUERY TRANSLATES
Displays a list of available translate tables.

QUERY USERS
Displays a list of defined user IDs.

QUERY VERSIONS
Displays the versions and maintenance levels of stack components.

QUERY ALTIPS

The QUERY ALTIPS command displays the current contents of the alternate IP address table.

Syntax

```
Query ALTips [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query altips
IPN253I << TCP/IP Alternate IP Addresses >>
IPN380I Alternative IP address, ID: TEST IPAddr: 64.10.5.5
IPN380I Alternative IP address, ID: SYS2 IPAddr: 64.10.5.2
```

Related Commands

DEFINE ALTIP
Causes the stack to monitor and respond to ARP requests for additional home addresses.

DELETE ALTIP
Removes an alternate home address.

GATEWAY
Controls forwarding of datagrams not intended for the VSE stack.

QUERY HOME
Displays all IP addresses in the “Home Address” table.

QUERY LINKS
Displays the status of network links.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

QUERY ARPS

The QUERY ARPS command displays the contents of the Address Resolution Protocol (ARP) table. This table maps a TCP/IP network address to a physical hardware address.

Syntax

`Query ARPs [,SYSLST] [,IPaddr=`*`ip4addr`*`]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`IPaddr=`
Specifies the TCP/IP network address for which the ARP entry is to be displayed. If omitted, all ARP table entries are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query arps
IPN253I << TCP/IP ARP Table >>
IPN409I Information for Link LINK3172, Adapter 0
IPN410I 192.168.1.4 > 00:00:00:00:00:00 C: 0
IPN410I 192.168.1.1 > 00:0C:41:80:DD:30 C: 1
IPN410I 192.168.1.66 > 00:19:B9:75:5E:96 C: 3
```

Notes

The following notes apply to this command:

- The stack updates the table every time it receives an ARP. If a MAC address changes, an alert message is issued.
- The "C:" field in the command response refers to the number of times that VSE has been ARPed from the IP address. A large number in this field could mean that a particular host is creating unnecessary TCP/IP overhead by ARPing frequently.
- If an IP address is reassigned to a different MAC address, its first attempt to contact VSE (for example, PING) will update the ARP table. This also occurs if the device issues a broadcast ARP.

Related Commands

QUERY ALTIPS
Displays all alternate IP addresses.

QUERY LINKS
Displays the status of network links.

QUERY ROUTES

Displays the content of the network routing table or the route taken to reach a specific address.

DEFINE ROUTE

Adds an entry to the TCP/IP FOR VSE routing table.

QUERY CGIS

The QUERY CGIS command provides information about common gateway interface (CGI) routines used by the HTTP daemons.

Syntax

Query CGIs [,SYSLST] [,PUBlic=*pubname*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

PUBlic=
Specifies the CGI to be displayed. If this option is omitted, all CGIs are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query cgis
IPN253I << TCP/IP CGIs >>
IPN432I Public Name:  CGIIPLDT
IPN433I Type:  CGI DLBL:  Driver: CGIIPLDT
IPN432I Public Name:  CGIIPLD2
IPN433I Type:  CGI-BAL DLBL:  Driver: IPNFCGIB
IPN432I Public Name:  CGISTATS
IPN433I Type:  CGI DLBL:  Driver: CGISTATS
IPN432I Public Name:  CGISUBM1
IPN433I Type:  CGI DLBL:  Driver: CGISUBM1
IPN432I Public Name:  VSECOM
```

Related Commands

DEFINE CGI
Loads a CGI program and makes it available for use.

DELETE CGI
Removes a CGI program from storage.

QUERY PROGRAMS
Displays the program phases being used by TCP/IP FOR VSE, their characteristics, their memory locations, and the library from which each was loaded.

QUERY CONNECTIONS

The QUERY CONNECTIONS command returns detailed information about the status of TCP/IP FOR VSE connections.

Syntax

```
Query CONnections [,SYSLST] [,IPaddr=ip4addr]
        [,FPort=port] [,LPort=port]
        [,STate={Listen|Established}] [,EXTended]
        [,Trace]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

IPaddr=
If specified, only connections for the specific address are displayed.

FPort=
If specified, the display is limited to connections using this foreign port.

LPort=
If specified, the display is limited to connections using this local port.

STate=
This parameter limits the display to the connection's state.

Listen
Only connections in the listen state are displayed.

Established
Only established connections are displayed. This includes connections that are in the process of terminating.

EXTended
This parameter causes extended information on the connection to be displayed. If omitted, only summary information is shown.

Trace
This parameter causes a mini trace to be displayed that shows the previous few inbound and/or outbound datagrams.

Example 1

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in these examples.

```
 query connections
IPN253I << TCP/IP Connections >>
IPT353I >21 0.0.0.0, 0 TCP 0 Listen
IPT345I Open by FTPDAEMN in F4 Ident: C59A9E0A96075000
IPT356I Total sockets: RECV: 0 (0); SEND: 0 (0); STAT: 0; CLOSE: 0
  ABORT: 0 CNTL: 0
IPT353I >21 192.168.1.66, 3911 TCP 0 Established
IPT345I Open by FTPDAEMN in F4 Ident: C59A952773C1E000
IPT341I Start: 13:57:54 Idle: 0.020 Duration: 2.579
IPT356I Total sockets: RECV: 12 (43); SEND: 12 (535); STAT: 12; CLOSE: 0
  ABORT: 0 CNTL: 12
IPT360I Socket RECV 0 bytes, Duration: 0.656 Timeout: 1/00:00:03.663
IPT353I >515 0.0.0.0, 0 TCP 0 Listen
IPT345I Open by LPD in F4 Ident: C599C9ACA0687000
IPT356I Total sockets: RECV: 0 (0); SEND: 0 (0); STAT: 0; CLOSE: 0
  ABORT: 0 CNTL: 0
```

Example 2

```
 query connections,extended
IPN253I << TCP/IP Connections >>
IPT353I >21 0.0.0.0, 0 TCP 0 Listen
IPT345I Open by FTPDAEMN in F4 Ident: C59A9E0A96075000
IPT356I Total sockets: RECV: 0 (0); SEND: 0 (0); STAT: 0; CLOSE: 0
  ABORT: 0 CNTL: 0
IPT353I >21 192.168.1.66, 3911 TCP 0 Established
IPT345I Open by FTPDAEMN in F4 Ident: C59A952773C1E000
IPT341I Start: 13:57:54 Idle: 10.595 Duration: 13.154
IPT342I Send IS: 70F02774 Data: 535
IPT342I Recv IS: F06E1465 Data: 43
IPT343I Route: LOCAL; MTU: 1,500; Send MSS: 1,460 (1,460); Recv MSS: 1,460;
  Buffer: 65,534/1,825
IPT344I Send: 25 blocks, 536 bytes. Retrans: 0 blocks, 0 bytes. Eff: 34% IPT344I
Recv: 16 blocks, 44 bytes. Duplicate: 0 blocks, 0 bytes. Eff: 6%
IPT350I Send Window: 70F0298C thru 70F12774; 65,000 bytes. Max:
65,535.   Closed: 0. Time closed: 0.000
IPT350I Recv Window: F06E1491 thru F06F148F; 65,534 bytes. Max:
65,534.   Closed: 0. Time closed: 0.000
IPT347I Fixed Retrans; Init: 1,000/1,000; Limit: 500/2,000; Delay:
500;   Retries: 50
IPT348I Retran Start: 1,000; Times: 0; Current: Off; Blocks: 0; Cost: 0 sec
IPT349I Pulse: Enabled; Interval: 60 sec; Count: 0
IPT352I Roundtrip: Min: 2 ms; Max: 217 ms; Last: 217 ms
IPT356I Queued sockets: RECV: 1 (0); SEND: 0 (0); STAT: 0; CLOSE: 0
  ABORT: 0 CNTL: 0
IPT356I Total sockets: RECV: 12 (43); SEND: 12 (535); STAT: 12; CLOSE:
0   ABORT: 0 CNTL: 12
IPT360I Socket RECV 0 bytes, Duration: 11.231 Timeout: 23:59:53.088
IPT353I >515 0.0.0.0, 0 TCP 0 Listen
IPT345I Open by LPD in F4 Ident: C599C9ACA0687000
IPT356I Total sockets: RECV: 0 (0); SEND: 0 (0); STAT: 0; CLOSE: 0 ABORT:
0   CNTL: 0
```

Example 3

```
 query connections,lport=21,trace
IPN253I << TCP/IP Connections >>
IPT353I >21 0.0.0.0, 0 TCP 0 Listen
IPT345I Open by FTPDAEMN in F4 Ident: C59A9E0A96075000
IPT356I Total sockets: RECV: 0 (0); SEND: 0 (0); STAT: 0; CLOSE: 0
  ABORT: 0 CNTL: 0
IPT353I >21 192.168.1.66, 3911 TCP 0 Established
IPT345I Open by FTPDAEMN in F4 Ident: C59A952773C1E000
IPT341I Start: 13:57:54 Idle: 0.000 Duration: 1:02.563
IPT356I Total sockets: RECV: 12 (43); SEND: 12 (535); STAT: 12; CLOSE: 0
  ABORT: 0 CNTL: 12
IPT360I Socket RECV 0 bytes, Duration: 1:00.640 Timeout: 23:59:03.679
IPT357I RECV: F06E1485 A: 70F02909 W: FE6B D: 0000 C: 10 ACK
IPT357I RECV: F06E1485 A: 70F02909 W: FE6B D: 0006 C: 18 ACK,PSH
IPT357I SEND: 70F02909 A: F06E148B W: FFFE D: 0031 C: 18 ACK,PSH
IPT357I RECV: F06E148B A: 70F0293A W: FE3A D: 0000 C: 10 ACK
IPT357I RECV: F06E148B A: 70F0293A W: FE3A D: 0006 C: 18 ACK,PSH
IPT357I SEND: 70F0293A A: F06E1491 W: FFF8 D: 0000 C: 10 ACK
IPT357I SEND: 70F0293A A: F06E1491 W: FFFE D: 0000 C: 10 ACK
IPT357I SEND: 70F0293A A: F06E1491 W: FFFE D: 0035 C: 18 ACK,PSH
IPT357I RECV: F06E1491 A: 70F0296F W: FE05 D: 0000 C: 10 ACK
IPT357I SEND: 70F0296F A: F06E1491 W: FFFE D: 001D C: 18 ACK,PSH
IPT357I RECV: F06E1491 A: 70F0298C W: FDE8 D: 0000 C: 10 (002) ACK
IPT357I SEND: 70F0298B A: F06E1491 W: FFFE D: 0000 C: 10 * ACK
```

Related Commands

DEFINE SOTRACE
Starts a Socket Trace.

DEFINE TRACE
Starts a Datagram Trace

FLUSH
Terminates all processing with a specific remote host.

PORTQUEUE
Controls how inbound connection requests are queued for an application.

QUERY DIAGNOSE

The QUERY DIAGNOSE command displays any currently enabled diagnostic modes.

Syntax

Query DIAGnose [,SYSLST]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query diagnose
IPN805I Diagnose on for: PERFORM
```

Related Commands

DIAGNOSE
Controls diagnostic display options.

QUERY LOGS
Displays available consoles and logs and their properties.

QUERY DUMPOPTIONS

The QUERY DUMPOPTIONS command displays the current format settings for dumps.

Syntax

```
Query DUMPoptions [,SYSLST]
```

Arguments

SYSLST

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query dumpoptions
IPN818I Dump Options for File I/O set to CSI format
```

Related Commands

DUMPOPTION

Controls the format of TCP/IP-initiated storage dumps.

QUERY EMAIL

The QUERY EMAIL command lists many of the current values that are set by the EMAIL command.

Syntax

`Query EMAil [,SYSLST]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query email
IPN253I << TCP/IP EMAIL >>
IPN835I  UserID: $EMAIL RPort: 25
IPN838I  Truncate: NO AtSign: 7C GMT: (System)
IPN840I  Translation table (General): Default
IPN840I  Translation table (Text): Default
IPN840I  Translation table (Attachments): Default
```

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

EMAIL
Sets global default options for the email client. **Note**: Issuing this command without any parameters produces the same results as QUERY EMAIL.

QUERY EVENTS

The QUERY EVENTS command enables you to inspect the current settings of TCP/IP FOR VSE system events. These events are processed by an automation daemon and result in FTP, LPR, or Email transfers.

Syntax

`Query EVents [,SYSLST] [,ID=`*`id`*`] [,DETail]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`ID=`
The ID of the event to be listed. If omitted, all events are displayed.

`DETail`
Displays not only the event, but all of the processes that are waiting to be delivered, are being delivered, or have finished being delivered but the final disposition of which has yet to be performed (deleting the report, or changing it to "HOLD").

Example 1

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example and the next.

```
 query events
IPN253I << TCP/IP Events >>
IPN587I  Event Pause Interval: 1500 (5 Seconds)
IPN273I  Maximum LPR Events: 5 Current Events: 0
IPN273I  Maximum FTP Events: 3 Current Events: 0
IPN273I  Maximum Email Events: 1 Current Events: 0
IPN426I  Event ID: AUTOLPR1
IPN428I  Class: Y, Queue: LST, Action: LPR, POWER SYSID:
IPN427I  Priority: Yes, Order: Yes, Script: L
IPN429I  Host field: DEST, User ID: $EVENT, Single: N
IPN430I  Action: LPR, Retries: 1, Time: 45 sec
IPN426I  Event ID: AUTOFTP2
IPN428I  Class: X, Queue: PUN, Action: FTP, POWER SYSID:
IPN427I  Priority: Yes, Order: Yes, Script: L
IPN429I  Host field: USER, User ID: $EVENT, Single: N
IPN430I  Action: FTP, Retries: 1, Time: 45 sec
IPN426I  Event ID: AUTOFTP1
IPN428I  Class: X, Queue: LST, Action: FTP, POWER SYSID:
IPN427I  Priority: Yes, Order: Yes, Script: L
IPN429I  Host field: USER, User ID: $EVENT, Single: N
IPN430I  Action: FTP, Retries: 1, Time: 45 sec
```

Example 2

```
 QUERY EVENTS,DETAILS
IPN253I << TCP/IP Queued Event Settings >>
IPN237I  (Generated on 10/05/15 at 11.02)
IPN273I  Maximum LPR Events: 5 Current Events: 0
IPN273I  Maximum FTP Events: 1 Current Events: 0
IPN273I  Maximum Email Events: 1 Current Events: 0
IPN273I  Maximum overall Events: 7 Current Events: 0
IPN237I  SINGLEDEST is on
IPN237I  AUTO_TIME is 30 seconds
IPN253I << TCP/IP Queued Event Details >>
IPN426I  Event ID: CLASSM            LST(M)  EMAIL
IPN443I     SVA-21429-000 <SYSENHED        >In Process
IPN443I     SVA-33192-000 <SYSENHED        >Failed to send, retrying
IPN443I     SVA-44958-000 <SYSENHED        >Completed and awaiting purge
```

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

DELETE EVENT
Terminates the monitoring of a POWER class.

SEPARATOR_PAGES
Controls the generation of POWER separator pages.

SET AUTO_TIME
Determines the interval for automation to check the POWER queues.

QUERY EXTERNAL

The QUERY EXTERNAL command displays certain information about partitions external to the TCP/IP FOR VSE stack that are communicating with the stack. The details include those connections within the TCP/IP FOR VSE stack as well as the attributes of the non-TCP/IP FOR VSE partition(s).

The output shows each TCP/IP FOR VSE identification number, which means that if you have two stacks operating within the same VSE, then you will see two different stack identifiers. This indicates whether a connection exists between two different TCP/IP FOR VSE stacks.

Syntax

```
Query EXTERNAL [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to both SYSLST and the SYSLOG.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
Q EXTERNAL
IPN684I BG PXBLOK:217E8F00 PIK:0010 TIK:0021 STACK:00 SQBLOK:0033D000
IPN685I BG last socket job step 20160724 5:40:27.116
IPN686I T1 csoc=0 + ntcp=0 = 0
IPN687I T1 spas=0 + async=0 + ntlk=0 = 0
IPN689I T1 Total passive opens(15) successful connects(2)
```

Notes

Use this command when you want to see overall SOCKET statistics for a specific batch partition as well as usage in a multi-stack environment.

Related Commands

EXTPGVS
Enables allocating external-partition socket requests in 31-bit private partition storage.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY EXTTYPES

The QUERY EXTTYPES command displays the entries in the External Types table that were last loaded into storage.

Syntax

```
Query EXTtypes [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query exttypes
IPN253I << TCP/IP External Names Table >>
IPN844I  AIF Binary S audio/x-aiff
IPN844I  AIFC Binary S audio/x-aiff
IPN844I  AIFF Binary S audio/x-aiff
IPN844I  ASC Text FB text/plain
IPN844I  AU Binary S audio/basic
IPN844I  AVI Binary S video/x-msvideo
IPN844I  BIN Binary S application/octet-stream
IPN844I  BINJOB Bin80 S
IPN844I  BJB Bin80 S
IPN844I  CAB Binary S application/octet-stream
...
IPN844I  TEXT Text FB text/plain
IPN844I  TIF Binary S image/tiff
IPN844I  TIFF Binary S image/tiff
IPN844I  TTF Binary S application/octet-stream
IPN844I  TXT Text FB text/plain
IPN844I  VCF Binary S application/octet-stream
IPN844I  WAV Binary S audio/x-wav
IPN844I  WRL Text FB x-world/x-vrml
IPN844I  WRZ Text FB x-world/x-vrml
IPN844I  XLS Binary S application/excel
IPN844I  XLT Binary S application/excel
IPN844I  XML TextC FB application/xml
IPN844I  ZIP Binary S application/zip
```

Notes

The following notes apply to this command:

- The External Types table is used by several applications (FTP, HTTP, EMAIL) to associate properties with a file based on the file's extension.
- Each line in the output shows an included file type, the assigned transfer type (Binary, Bin80, TEXT, or TEXTC), the format (String or Fixed Block), and the MIME content type, which identifies the format of files on the Internet. For FTP, default values for LRECL, BLKSIZE, and RECFM are associated with each transfer type. For details, see the *TCP/IP FOR VSE Installation Guide*, chapter 6, "Configuring FTP Clients and Daemons," subsection "Controlling Defaults Using EXTTYPES.L".

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE HTTPD
Creates a Hypertext Transfer Protocol (Web server) daemon.

EMAIL
Sets global default options for the email client.

RELOAD
Reloads a control table.

QUERY FILEIO

The QUERY FILEIO command displays information on installed file I/O drivers.

Syntax

```
Query FILEIo [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query fileio
IPN253I << TCP/IP File I/O Drivers >>
IPN640I  REXX  Driver: IPNFREXX Entry: 805D3050 Use:  0
IPN640I  LIBRARY  Driver: IPNFLIBR Entry: 805BC050 Use:  0
IPN640I  POWER  Driver: IPNFPOWR Entry: 005B2050 Use:  0
```

Related Commands

DEFINE FILE
Defines a file in the TCP/IP FOR VSE file system and associates it with a file I/O driver.

DEFINE FILEIO
Defines a file I/O driver for a file type.

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY FILES

The QUERY FILES command enables you to inspect the current specification of one or more files in the TCP/IP FOR VSE file system.

Syntax

`Query FIles [,SYSLST] [,PUBlic=`*`pubname`*`]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`PUBlic=`
The public name of a file to be listed. If omitted, all files are listed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query files
IPN253I << TCP/IP Files >> IPN432I  Public Name: DEVLIB
IPN433I  Type: LIBRARY DLBL: DEVLIB Driver: IPNFLIBR
IPN545I  GID: +0 UID: +0 Permissions: ------
IPN432I  Public Name: IJSYSRS
IPN433I  Type: LIBRARY DLBL: IJSYSRS Driver: IPNFLIBR
IPN545I  GID: +0 UID: +0 Permissions: ------
IPN432I  Public Name: POWER
IPN433I   Type: POWER DLBL: Driver: IPNFPOWR
IPN545I   GID: +0 UID: +0 Permissions: ------
IPN432I  Public Name: PRD1
IPN433I   Type: LIBRARY DLBL: PRD1 Driver: IPNFLIBR
IPN545I   GID: +0 UID: +0 Permissions: ------
IPN432I  Public Name: PRD2
IPN433I  Type: LIBRARY DLBL: PRD2 Driver: IPNFLIBR
IPN545I   GID: +0 UID: +0 Permissions: ------
IPN432I  Public Name: PRODLIB
IPN433I  Type: LIBRARY DLBL: PRODLIB Driver: IPNFLIBR
IPN545I  GID: +0 UID: +0 Permissions: ------
```

Related Commands

DEFINE FILE
Defines a file in the TCP/IP FOR VSE file system and associates it with a file I/O driver.

DELETE FILE
Removes a file from the TCP/IP FOR VSE file system.

QUERY FIREWALL

The QUERY FIREWALL command displays the current firewall settings being enforced for the Firewall Shield optional feature.

Syntax

```
Query FIREWALL [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
query firewall
IPI208I FIREWALL loaded at 82240980 with 10 ranges defined
IPI210I FIREWALL range allowed 127.0.0.1-127.0.0.1 Hits:248 ICMP=YES FREEPBLK=NO
IPI215I TCPPORTS:ALL.
IPI216I UDPPORTS:ALL.
IPI210I FIREWALL range allowed 38.101.62.131-38.101.62.131 Hits:582 ICMP=YES
FREEPBLK=NO
IPI215I TCPPORTS:ALL.
IPI216I UDPPORTS:ALL.
IPI210I FIREWALL range allowed 38.101.62.132-38.101.62.255 Hits:0 ICMP=YES
FREEPBLK=YES
IPI215I TCPPORTS:ALL.
IPI216I UDPPORTS:ALL.
IPI210I FIREWALL range allowed 69.55.70.38-69.55.70.38 Hits:0 ICMP=YES FREEPBLK=NO
IPI215I TCPPORTS:ALL.
IPI216I UDPPORTS:ALL.
IPI210I FIREWALL range allowed 216.29.249.42-216.29.249.42 Hits:0 ICMP=YES
FREEPBLK=NO
IPI215I TCPPORTS:NONE.
IPI216I UDPPORTS:ALL.
IPI210I FIREWALL range allowed 39.101.62.0-39.101.62.255 Hits:0 ICMP=YES FREEPBLK=YES
IPI215I TCPPORTS:00021,00020.
IPI216I UDPPORTS:NONE.
```

Related Commands

FIREWALL
Controls and monitors the Firewall Shield optional feature.

IPSTAT
Controls whether an ISBLOK is allocated for every IP address that accesses the stack.

QUERY IPSTAT
Displays IP address statistics stored in ISBLOKs.

QUERY FRAGMENTS

The QUERY FRAGMENTS command allows you to inspect the current state of the fragmented-datagram reassembly process.

Syntax

`Query FRAGments [,SYSLST]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query fragments
IPN253I << TCP/IP IP Fragmentation >>
IPN812I Pending Fragments:    3
IPN812I Pending Datagrams:    2
IPN812  Total Fragments:    987
IPN812I Total Reassembled:  857
IPN812I Total Used:         752
IPN812I Total Expired:       15
IPN812I Total Duplicates:    27
IPN812I Total Invalid:        0
```

QUERY FTPDS

The QUERY FTPD command displays the characteristics of the currently defined FTP daemons.

Syntax

`Query FTPds [,SYSLST] [,ID=`*`id`*`] [,EXTended]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`ID=`
The ID of the FTP daemon assigned by a DEFINE FTPD command.

`EXTended`
Additional information is displayed for each daemon. **Note:** The "Idle timeout" value is not displayed if the default (0) was used.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query ftpds
IPN253I << TCP/IP FTP Daemons >>
IPN435I  ID: FTP01  Port: 21 Driver: FTPDAEMN SSL: No
IPN874I  Buffers: 2 Buffer size: 65536 bytes
IPN350I  Current Status: Listen
IPN362I  Maximum sessions: 5, Current: 1
IPN364I  Userid: EWF connected from 192.168.1.66,3928
IPN881I 1 FTP Daemons defined, 1 sessions active

 query ftpds,extended
IPN253I << TCP/IP FTP Daemons >>
IPN435I  ID: FTPD9921  Port: 9921 Driver: FTPDAEMN SSL: No
IPN874I  Buffers: 2 Buffer size: 65536 bytes
IPN350I  Current Status: Listen
IPN362I  Maximum sessions: 3, Current: 0
IPN435I  ID: FTPD0021  Port: 21 Driver: FTPDAEMN SSL: No
IPN874I  Buffers: 2 Buffer size: 65536 bytes
IPN872I  Xmit hesitation: 0, Idle timeout: 120 seconds
IPN350I  Current Status: Listen
IPN362I  Maximum sessions: 13, Current: 1
IPN364I  Userid: DSTOEVER connected from 192.168.0.153,1870
IPN365I  Started at: 17:00:15 2010/02/27
IPN366I  2:files sent,  3:files received
IPN367I  79,426 bytes sent (79,426), 90,646 bytes received
IPN368I  Last Command: RETR SERVP15G.ZIP
IPN363I  Last Command time: 17:02:55 2010/02/27
IPN368I  Last reply: 226 Closing data connection
IPN363I  Last reply time: 17:02:57 2010/02/27
IPN881I 2 FTP Daemons defined, 1 sessions active
```

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DELETE FTPD
Terminates a File Transfer Protocol daemon.

QUERY ACTIVE
Displays the status of active daemons.

QUERY GPSDS

The QUERY GPSD command provides information about GPS daemons.

Syntax

Query GPSD [,SYSLST] [,ID=*id*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
The ID of the specific GPS daemon for which you want information. If this argument is omitted, all daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query gpsd,id=gpsd3
IPN253I << TCP/IP GPS Daemons >>
GPS930I GPSD3 Host:  192.168.020.007 Printer:  KEDIT
GPS931I GPSD3 Termname:  GPS3 Logmode:  DSC2K
GPS932I GPSD3 User:  Translate:  Script:
GPS933I GPSD3 Log=NO Debug=NO
GPS934I GPSD3 Insess=DBDCCICS Target:
GPS935I GPSD3 Storage=PRD2.SAVE
GPS936I GPSD3 Retries:  3 Delay:  18000
GPS937I GPSD3 Maxpages:  100 Now queued:  0
GPS938I GPSD3 Maxlines:  10000 Now queued:  0
GPS939I GPSD3 Maxchars:  1000000 Now queued:  0
GPS940I GPSD3 Maxidle:  3000 (10 seconds)
```

Related Commands

DEFINE GPSD
Creates a General Print Server daemon.

DELETE GPSD
Terminates a General Print Server daemon.

QUERY HOME

The QUERY HOME command causes a display of all IP addresses by which the stack is known and reachable.

Syntax

```
Query HOME [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query home
IPN253I << TCP/IP Home Addresses >>
IPN582I   127.0.0.1           Link    *Internal
IPN582I   192.168.1.161       Main
IPN582I   192.168.1.161       Link    LINK3172
IPN582I   192.168.1.162       ARP
```

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ALTIP
Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE LINK
Creates a link between TCP/IP FOR VSE and a network or to a directly connected stack.

SET IPADDR
Establishes the default home address for the stack.

QUERY HTTPDS

The QUERY HTTPDS command displays the characteristics of the currently defined HTTP (Web) daemons.

Syntax

`Query HTtpds [,SYSLST] [,ID=`*`id`*`]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`ID=`
Specifies the name of the HTTP daemon as assigned by the DEFINE HTTPD command. If this option is omitted, all HTTP daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query httpds
IPN253I << TCP/IP HTTP Daemons >>
IPN509I  ID: H1  Port:  80  Driver: HTTPD
IPN510I   Confine: NO Root: TBXBITL.TBXBITL.
IPN867I  Ignorename: Yes Max: 50 Actual: 0
HTT946I  Security: No Scanblocker: NO
HTT948I  Translate: OS_02  SOSI: None/None
HTT949I  Count: 3 Timeout: 90000
HTT950I ------------------------------------
```

Related Commands

ASECURITY WEBL
Sets access security based on the network address.

DEFINE HTTPD
Creates a Hypertext Transfer Protocol (Web server) daemon.

DELETE HTTPD
Terminates a Hypertext Transfer Protocol (Web server) daemon.

QUERY IBBLOKS

The QUERY IBBLOKS command displays current settings and statistics related to Internet Buffer Block usage.

Syntax

```
Query IBbloks [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

The key fields in this example are described in the table below. See the *TCP/IP FOR VSE Messages* manual for more information.

```
 query ibbloks
IPN253I << TCP/IP IBBLOK Activity >>
IPN896I Counts - Current: 47 Peak: 86 Failed: 0
IPN897I Space - Current: 36k Peak: 109k Max: 51,092k (127%)
IPN889I Size: 384 MTU: 64 Free: 24/50 Hits: 4,465 Misses: 24
IPN889I Size: 512 MTU: 192 Free: 4/20 Hits: 569 Misses: 4
IPN889I Size: 640 MTU: 320 Free: 2/20 Hits: 54 Misses: 2
IPN889I Size: 768 MTU: 448 Free: 1/20 Hits: 24 Misses: 1
IPN889I Size: 896 MTU: 576 Free: 1/20 Hits: 12 Misses: 1
IPN889I Size: 1,024 MTU: 704 Free: 2/20 Hits: 20 Misses: 2
IPN889I Size: 1,152 MTU: 832 Free: 1/20 Hits: 32 Misses: 1
IPN889I Size: 1,280 MTU: 960 Free: 3/20 Hits: 32 Misses: 3
IPN889I Size: 1,408 MTU: 1,088 Free: 1/20 Hits: 31 Misses: 1
IPN889I Size: 1,536 MTU: 1,216 Free: 1/20 Hits: 32 Misses: 1
IPN889I Size: 1,664 MTU: 1,344 Free: 1/20 Hits: 7 Misses: 1
IPN889I Size: 1,792 MTU: 1,472 Free: 1/20 Hits: 1 Misses: 1
IPN889I Size: 1,920 MTU: 1,600 Free: 5/5 Hits: 2,892 Misses: 2,679
IPN889I Size: 64K+ MTU: -- Free: 0/0 Hits: 0 Misses: 0
```

Field	Description
Size	Buffers (IBBLOKs) are allocated only in discrete, 128-byte increments. Buffers up to 8K bytes long are retained in an IBBLOK pool for reuse, and same-sized buffers are tracked and displayed as a group. This value indicates the IBBLOK size to which the following field values apply.
MTU	The maximum size datagram that will fit in the buffer. Essentially, this is the data size plus 40 bytes for the headers.
Free	The number of currently free IBBLOKs out of the total for this size in the pool.
Hits	The number of times a free IBBLOK of this size was found in the pool and reused.
Misses	The number of times a free IBBLOK of this size was not found when requested and memory was allocated. IBBLOKs greater than 8K are always allocated and released.

Notes

Use the IBBLOK command to monitor and control the use of IBBLOKs.

QUERY IPSTAT

The QUERY IPSTAT command displays the contents of ISBLOKs for IP addresses that have accessed the stack when the IPSTAT command is set to ON (the default). The Firewall Shield optional feature uses the contents of the ISBLOKS.

Caution:

This command generates 70+ lines of output for every IP address (ISBLOK) and could run for a very long time if there is a huge number of ISBLOKs.

Syntax

```
Query IPSTAT [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query ipstat
IPN253I << TCP/IP IP Statistics >>
IPN516I   ......................
IPN516I   IP Address.........................127.0.0.1
IPN516I    Performance Information.....................
IPN516I    - Overall..................................
IPN516I      - TCP Connections.....................2
IPN516I      - Total Connect Time.....................0
IPN516I      - Maximum Turn Around....................0
IPN516I      - Maximum Depth..........................0
IPN516I      - Maximum Window.........................0
IPN516I    - SWS Mode.................................
IPN516I      - Time...................................0
IPN516I      - Count..................................0
IPN516I    - Depth Mode...............................
IPN516I      - Time...................................0
IPN516I      - Count..................................0
IPN516I    - Retransmit Mode..........................
IPN516I      - Time...................................0
IPN516I      - Times in Mode..........................0
IPN516I      - Count..................................0
IPN516I    Application Information.....................
IPN516I    - FTP......................................
IPN516I      - Connections............................0
IPN516I      - Inbound Files..........................0
IPN516I      - Outbound Files.........................0
IPN516I      - Inbound File Bytes.....................0
IPN516I      - Outbound File Bytes....................0
IPN516I    - HTTP.....................................
IPN516I      - Connections............................0
```

```
IPN516I      - Inbound Files.........................0
IPN516I      - Outbound Files........................0
IPN516I      - Inbound File Bytes.....................0
IPN516I      - Outbound File Bytes....................0
IPN516I    - Telnet..................................
IPN516I      - Connections...........................0
IPN516I      - Inbound Files.........................0
IPN516I      - Outbound Files........................0
IPN516I      - Inbound File Bytes.....................0
IPN516I      - Outbound File Bytes....................0
IPN516I    Transfer Information.....................
IPN516I    - Non IP..................................
IPN516I      - Datagrams.............................0
IPN516I      - Bytes.................................0
IPN516I    - Misdirected IP..........................
IPN516I      - Datagrams.............................0
IPN516I      - Bytes.................................0
IPN516I    - ARP Inbound.............................
IPN516I      - Inbound Datagrams.....................0
IPN516I      - Outbound Datagrams....................0
IPN516I      - Inbound Bytes.........................0
IPN516I      - Outbound Bytes........................0
IPN516I    - ICMP....................................
IPN516I      - Inbound Datagrams.....................0
IPN516I      - Outbound Datagrams....................0
IPN516I      - Inbound Bytes.........................0
IPN516I      - Outbound Bytes........................0
IPN516I    - IP......................................
IPN516I      - Inbound Datagrams.....................19
IPN516I      - Outbound Datagrams....................18
IPN516I      - Inbound Bytes.....................7,160
IPN516I      - Outbound Bytes....................6,796
IPN516I    - TCP.....................................
IPN516I      - Inbound Datagrams.....................19
IPN516I      - Outbound Datagrams....................18
IPN516I      - Inbound Bytes.....................7,160
IPN516I      - Outbound Bytes....................6,796
IPN516I    - UDP.....................................
IPN516I      - Inbound Datagrams.....................0
IPN516I      - Outbound Datagrams....................0
IPN516I      - Inbound Bytes.........................0
IPN516I      - Outbound Bytes........................0
IPN516I    - Refused Data............................
IPN516I      - Inbound Datagrams.....................0
IPN516I      - Inbound Bytes.........................0
```

Related Commands

IPSTAT
Controls whether an ISBLOK is created for every IP address that accesses the TCP/IP FOR VSE stack.

FIREWALL
Controls and monitors the Firewall Shield optional feature.

QUERY ISTATISTICS

The QUERY ISTATISTICS causes the display of internal TCP/IP FOR VSE statistics. TCP/IP FOR VSE maintains counters to furnish information on how various internal processes are functioning.

Syntax

```
Query ISTATistics [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query istatistics
IPN253I << TCP/IP Internal Statistics >>
IPN516I  Total Dispatches.....................4,579
IPN516I  - All Bound..........................335
```

Related Commands

QUERY STATISTICS
Displays statistics detailing internal stack functions.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY LINKS

The QUERY LINKS command displays the status of TCP/IP links to the various physical networks.

Syntax

Query LINks [,SYSLST] [,ID=*id*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
The ID (as specified in the DEFINE LINK command) identifying the link to be displayed. If this option is omitted, all links are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query links
IPN253I << TCP/IP LINKS >>
IPN437I  ID: LINK3172  Type: OSA2  Dev: (0032,0033)
IPN350I  Current Status: Active
IPN440I  ID: LINK3172  Type: 802.3  Adapter: 0
IPN438I  MTU:  1500  IP Address: 192.168.1.161
IPN441I  MAC address: 00:00:E2:90:C0:45
IPN350I  Current Status: Active
IPN437I  ID: *Internal   Type: *Int  Dev: (0000,0000)
IPN438I  MTU: 32767  IP Address: 127.0.0.1
IPN350I  Current Status: Active
```

Notes

For LCS-type devices, all adapters associated with a link are also displayed.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE LINK
Creates a link between TCP/IP FOR VSE and a network or to a directly connected stack.

DELETE LINK
Removes a link between TCP/IP FOR VSE and a network or to a directly connected stack.

QUERY LOCKS

The QUERY LOCKS command displays information on the stack's internal lock mechanism. It shows the number of times a subtask may be paused because of conflicts or as a result of normal task balancing.

Syntax

```
Query LOCKs [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query locks
IPN253I << TCP/IP Lock Information >>
IPN936I Lock: SOBLOK  Requests: 0 Conflicts: 0
IPN936I Lock: TKBLOK  Requests: 115 Conflicts: 0
IPN936I Lock: CCBLOK  Requests: 9260 Conflicts: 20
IPN936I Lock: LDBLOK  Requests: 109 Conflicts: 0
IPN936I Lock: IBBLOK  Requests: 3135 Conflicts: 0
IPN936I Lock: FILEIO  Requests: 1521 Conflicts: 0
IPN936I Lock: ANBLOK  Requests: 0 Conflicts: 0
```

QUERY LOGS

This command displays the status of TCP/IP FOR VSE's various displays and logs.

Syntax

```
Query LOGs [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query logs
IPN253I << TCP/IP Console Logs >>
IPT240I CONSOLE on CONSOLE
IPT241I  Line Length: 65 Lines per page: 0 Timestamp: None
IPT242I  Total lines: 0 Total pages: 0
IPT243I  Logging:  Critical Vital Warning Important Info Response Security
IPT240I SYSLST on SYSLST
IPT241I  Line Length: 132 Lines per page: 86 Timestamp: Left
IPT242I  Total lines: 300 Total pages: 0
IPT243I  Logging:  Critical Vital Warning Important Info Diag
IPT240I SYS007 on SYS007
IPT241I  Line Length: 132 Lines per page: 0 Timestamp: None
IPT242I  Total lines: 44 Total pages: 0
IPT243I  Logging: Response Security
```

Related Commands

DEFINE LOG
Creates a system log file.

MODIFY LOG
Changes the characteristics of a system log file.

QUERY LPDS

The QUERY LPDS command displays the status of currently defined Line Printer daemons.

Syntax

```
Query LPds [,SYSLST] [,ID=id]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
The ID of a Line Printer daemon, as specified by the DEFINE LPD command. If this keyword is omitted, all Line Printer daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query lpds
IPN253I << TCP/IP Line Printer Daemons >>
IPN444I  Print Name: HEX UserID: $LPD Hexdump Mode
IPN445I  Queue: POWER.LST.A.
IPN446I  Library: MEMORY.
IPN444I  Print Name: LOCAL UserID: $LPD
IPN445I  Queue: POWER.LST.A.
IPN446I  Library: MEMORY.
```

Related Commands

DEFINE LPD
Creates a Line Printer daemon.

DELETE LPD
Terminates a Line Printer daemon.

QUERY MASKS

The QUERY MASKS command displays the current contents of the sub network mask table.

Syntax

```
Query MASKs [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query masks
IPN253I << TCP/IP Network Masks >>
IPN575I  Network: 192.168.30.0 (12,625,950) Mask: 255.255.255.0
IPN575I  Network: 127.0.0.0 (127) Mask: 255.0.0.0
IPN575I  Network: Default (--) Mask: 255.255.255.0
```

Notes

The following notes apply to this command:

- Only one subnet mask may be specified for any given network.
- See the discussion of IP addresses, masks, and subnetworking in the *TCP/IP FOR VSE Installation Guide*.
- If a mask is not defined for a particular network, the default mask (SET MASK=) is used.

Related Commands

DEFINE MASK
Creates a subnet mask for a particular network.

DELETE MASK
Deletes a subnet mask for a particular network.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

QUERY MENUS

The QUERY MENUS command displays the menus currently available for use by Telnet daemons.

Syntax

`Query MENus [,SYSLST]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query menus
IPN253I << TCP/IP Menu Definitions >>
IPN391I  ID: MENUUSR  Member: MENU2
IPN391I  ID: MENUADM  Member: MENU1
```

Related Commands

DEFINE MENU
Loads a menu file and makes it available for use by Telnet daemons.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

DELETE MENU
Removes a TN3270 menu file from memory.

QUERY TELNETDS
Displays the TN3270 and TN3270E daemons.

QUERY NAMES

The QUERY NAMES command displays the contents of the symbolic names table created by DEFINE NAME (IPv4-only addresses).

Syntax

Query NAMes [,SYSLST] [,TYPe={ALL|STATIC|DYNAMIC}]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

TYPe=
Specifies a subset of names to display:

STATIC
Only names that were explicitly defined with a DEFINE NAME command are displayed. This is the default.

DYNAMIC
Only names that were resolved by a DNS call and are currently in cache are displayed. (Names added by DEFINE NAME are not displayed.)

ALL
Both static and dynamic entries are displayed.

Example

```
 query names
IPN253I << TCP/IP Names >>
IPN503I  Symbolic Name: AUTOFTP
IPN504I   Script Name: AUTOFTP
IPN503I  Symbolic Name: AUTOLPR
IPN504I   Script Name: AUTOLPR
IPN503I  Symbolic Name: DEVPUNE
IPN504I   Script Name: DEVPUNE
IPN503I  Symbolic Name: DEVPUN
IPN504I   Script Name: DEVPUN
IPN503I  Symbolic Name: DEVLST
IPN504I   Script Name: DEVLST
IPN503I  Symbolic Name: DELL
IPN504I  IP Address: 192.168.1.66
IPN503I  Symbolic Name: VSE01
IPN504I  IP Address: 192.168.1.162
IPN503I  Symbolic Name: VSE26
IPN504I  IP Address: 192.168.1.161
IPN503I  Symbolic Name: VSE
IPN504I  IP Address: 192.168.1.161
```

Related Commands

DEFINE NAME

Associates a TCP/IP name with an IPv4 IP address or a script file.

DELETE NAME

Removes a TCP/IP symbolic name from the names table created by DEFINE NAME.

QUERY NTPD

The QUERY NTPD command provides information on a currently running NTP daemon.

Syntax

`Query NTPd [,SYSLST] [,ID=`*`id`*`]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`ID=`
Specifies the ID from the DEFINE NTPD command that created the daemon you want to display. If this keyword is omitted, all NTP daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query ntpd
IPN253I << TCP/IP NTP Daemons >>
NTP105I NTP Daemon: ID NTP1
NTP106I  Protocol: UDP Port 37
NTP107I  Hour correction: +0 Second correction: +0
NTP108I  Requests: 0 Last requestor: 0.0.0.0
```

Related Commands

DEFINE NTPD
Creates a Network Time Server daemon.

DELETE NTPD
Terminates a Network Time Server daemon.

QUERY OPENFILES

This command displays a list of all files that are currently open in the TCP/IP FOR VSE partition's file system.

Syntax

```
Query OPENfiles [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query openfiles
IPN576I << TCP/IP Open Files >>
IPN432I Public Name:   EWF.SAVE.T999.LOG
IPN433I Type:   LIBRARY DLBL:   EWF Driver:  IPNFLIBR
IPN577I Mode:   Output Record Counter:   0
IPN432I Public Name: EWF.SAVE.T999.PRINT
IPN433I Type: LIBRARY DLBL:   EWF Driver:   IPNFLIBR
IPN577I Mode:   Output Record Counter:   2
```

Notes

If multiple processes are using a file, the file is displayed multiple times.

Related Commands

CLOSE FILE
Closes an open file.

QUERY FILES
Displays the contents of the TCP/IP FOR VSE file system.

QUERY OPTIONS (or QUERY SET)

The QUERY OPTIONS command lists the current setting of all variables accessible by the SET command.

QUERY SET is a synonym for QUERY OPTIONS.

Syntax

```
Query OPTions [,SYSLST]
```

Arguments

SYSLST

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query options
IPN253I << TCP/IP Current Options >>
IPN452I System ID: 00
IPN453I IP Address: 192.168.1.161 Submask: 255.255.255.0
IPN454I Link Retry Time: 1800
IPN458I Console Hold: On Record: Off
IPN459I Gateway: Off Isolation: Off
IPN804I Telnet translation will use System Default
IPN806I Checksum: Software Connect_Sequence: Off
IPN806I Downcheck: Off Dynamic_Route: On
IPN806I Fixed_Retransmit: Off Full_CETI: On
IPN806I Full_Critical: Off Local_DLBL: On
IPN806I Ping_Message: On
IPN806I Email Separator_Pages: Off
IPN806I HTTP Separator_Pages: Off
IPN806I LPR Separator_Pages: Off
IPN806I FTP Separator_Pages: Off
IPN806I SDOpen_Extra: Off Spincheck: On
IPN806I Traffic: On Singledest: On
IPN806I ARPdelete: Off ARP_Time: 90000
IPN806I Auto_time: 1500 Pulse_Time: 18000
IPN806I Console_Port: 0
IPN806I DNS1: 65.24.7.3 Timeout: 1200
IPN806I DNS3: 65.24.7.6 Timeout: 1200
IPN806I Reuse_Size: 10
IPN806I Window_Depth: 5
IPN806I Default_Domain:
IPN806I Separator Page Count: 0
IPN806I FTPBATCH_Fetch: Off
IPN806I Subtask_OPEN: Off Stealth mode: Off
IPN806I Memory Verification: Off
IPN806I Buffer Validation: Off POWERUSERID: SYSTCPIP
IPN806I ListIDCAMS: On AutoLoad: On
```

QUERY PORTQUEUE

The QUERY PORTQUEUE command displays the current status of all ports where connections are being queued.

Syntax

```
Query PORTqueue [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query portqueue
IPN253I << TCP/IP Port Queuing >>
IPN400I Port: 88 Timeout: 60 sec Limit: 100
IPN401I Queued: 0/0/0 Reject: 0 Good: 0 Lost: 0
IPN402I Avg wait: 0 msec; Avg lost: 0 msec; Int: 5:16.754
IPN403I Partition: FB Phase: IPNET
IPN400I Port: 80 Timeout: 60 sec Limit: 100
IPN401I Queued: 0/59/177 Reject: 0 Good: 105 Lost: 72
IPN402I Avg wait: 1,226 msec; Avg lost: 3,209 msec; Int: 5:16.764
IPN403I Partition: FB Phase: IPNET
```

Notes

Port queuing can be established by applications (BSD). Values established by applications can be overridden by the operator. See the following references for more information.

- *TCP/IP FOR VSE Installation Guide*, chapter 12, "Performance," subsection "Port Queuing"
- *TCP/IP FOR VSE Programmer's Guide*, chapter 2, "BSD Socket Interface"

Related Commands

PORTQUEUE
Controls how inbound connection requests are queued for an application.

QUERY CONNECTIONS
Displays the status of one or more connections.

QUERY PRODKEYS

The QUERY PRODKEYS command displays the product keys currently in use.

Syntax

```
Query PRODkeys [,SYSLST] [,ALL]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ALL
Besides displaying the product keys that are in use, this option also displays those products for which you do not have a valid key for the specific CPU ID indicated in the output.

Example 1

```
 query prodkeys
IPN253I << TCP/IP Product Keys >>
IPN886I Stack....... (CSI) Expires on 2017/07/15
IPN886I SSL......... (CSI) Expires on 2017/07/15
IPN886I GPS......... (CSI) Expires on 2017/07/15
IPN886I SecureFTP... (CSI) Expires on 2017/07/15
```

Example 2

```
 query prodkeys,ALL
IPN253I << TCP/IP Product Keys >>
IPN885I CPU ID: 01D377 (014377)
IPN886I Stack....... (CSI) Expires on 2020/12/31
IPN886I Base........ included in Stack
IPN886I Firewall.... not licensed
```

Notes

When multiple keys are present for a particular feature, TCP/IP FOR VSE selects the best available key.

Related Commands

QUERY PROGRAMS
Displays the program phases being used by TCP/IP FOR VSE, their characteristics, their memory locations, and the library from which each was loaded.

QUERY PROGRAMS

The QUERY PROGRAMS command displays information about the various phases being used to provide TCP/IP and related services.

Syntax

```
Query PROGrams [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query programs
IPN253I << TCP/IP Phases and Routines >>
IPN636I  FTPDAEMN 00604050 02E250 80604050 SYST   RENT   PRODLIB.PHASE
IPN636I  IPDSLIBR 005F6050 000958 805F6050 SYST   RENT   PRODLIB.PHASE
IPN636I  IPTRLOAD 005103A0 0026F1 00000000 SYST   RENT   PRODLIB.PHASE
IPN636I  ASOCKET   00319000 0011A8 80319020 SYS    SVA    PRODLIB.PHASE
IPN636I  CLIENTD   00633050 011F78 80633050 0020          PRODLIB.PHASE
IPN636I  CSOCKET   0361F050 002D98 8361F050 SYS           PRODLIB.PHASE
IPN636I  IOAPPEND 002E60F0 000460 802E60F0 SYS    SVA    PRODLIB.PHASE
IPN636I  IPCCSERV 0361A050 002AE8 8361A050 SYS           PRODLIB.PHASE
IPN636I  IPNCICMP 0354A050 001600 8354A050 0013          PRODLIB.PHASE
IPN636I  IPNCOCMP 03552850 000540 83552850 0014          PRODLIB.PHASE
IPN636I  IPNETAS   005AD050 000B00 805AD050 SYS          PRODLIB.PHASE
IPN636I  IPNFLIBR 005BC050 016D20 805BC050               PRODLIB.PHASE
IPN636I  IPNFPOWR 005B2050 009788 005B2050               PRODLIB.PHASE
IPN636I  IPNFREXX 005D3050 0011E8 805D3050               PRODLIB.PHASE
IPN636I  IPNIRAW1 03558050 000F50 83558050 0007          PRODLIB.PHASE
IPN636I  IPNIRAW2 03556050 001848 83556050 0008          PRODLIB.PHASE
IPN636I  IPNIVFGH 03627050 004A60 83627050 SYS           PRODLIB.PHASE
IPN636I  IPNLIEEE 03549850 0004B0 83549850 0017          PRODLIB.PHASE
IPN636I  IPNLOEEE 03549050 000450 83549050 0018          PRODLIB.PHASE
IPN636I  IPNL127   03647850 000680 83647850 0002          PRODLIB.PHASE
IPN636I  IPNRBSDC 044852D0 014D30 844892B0 SYS    SVA    PRODLIB.PHASE
IPN636I  IPNROUTE 03622050 001288 83622050 SYS           PRODLIB.PHASE
IPN636I  IPNTITCP 0354E050 001AD8 8354E050 000F          PRODLIB.PHASE
IPN636I  IPSECURE 00595050 003260 00595050 SYS           PRODLIB.PHASE
IPN636I  LOCKMGRX 0052C050 004AB8 8052C050 SYS           PRODLIB.PHASE
IPN636I  LPD       03559050 00B1A0 83559050 0005         PRODLIB.PHASE
IPN636I  MSKELIP   03649050 0193E8 83649050 SYS          PRODLIB.PHASE
IPN636I  PRODKEYS 00525850 000310 00525850 SYS           PRODLIB.CONFIG
IPN636I  SOCKPASS 0361D050 001CD8 8361D050 SYS           PRODLIB.PHASE
IPN636I  SOCTRACE 03624050 002E08 83624050 SYS           PRODLIB.PHASE
IPN636I  VERCHECK 00522050 0036A0 00522050 SYS           PRODLIB.PHASE
```

Related Commands

QUERY VERSIONS
Displays the versions and maintenance levels of stack components.

QUERY PUBLISHERS

The QUERY PUBLISHERS command displays the status of the publisher daemons.

Syntax

Query PUBLishers [,SYSLST] [,ID=*id*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
The ID of the publisher daemon, as assigned by the DEFINE PUBLISHER command. If this keyword is omitted, all publisher daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query publishers
IPN694I PUBLISH01 publishing daemon active
IPN694I Event-to-action List:
IPN694I Event-Item 1 will be passed to ENTRFRSH CSI-FAQS IMOD handler
```

Related Commands

DEFINE PUBLISHER
Creates a publisher daemon.

DELETE PUBLISHER
Terminates a publisher daemon.

QUERY ROUTES

The QUERY ROUTES command displays the current contents of the routing table.

Syntax

Query ROUTes [,SYSLST] [,ID=*id*|IPaddr=*ip4addr*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
The ID of the route statement to be displayed. If you do not specify the ID= parameter or the IPADDR= parameter, all route table entries are displayed.

IPaddr=
Specifies the full TCP/IP network address to be displayed. The routing table is searched and the first matched entry is displayed. If you do not specify the ID= parameter or the IPADDR= parameter, all route table entries are displayed.

Example 1

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query routes
IPN253I << TCP/IP Routes >>
IPN448I  ID: *Internal  Link ID: *Internal
IPN449I  IP Address: 127.0.0.0 Mask: 255.0.0.0
IPN450I  Net: 127.0.0.0 Subnet: -- Host: --
IPN875I  MTU: 32767 Max Seg: 32727 Pulse: 60s
IPN876I  SYN Retran: 100ms Data Retran: 100ms Fixed: Yes
IPN877I  Retran Min: 100ms Max: 100ms
IPN882I  Retry Delay: 100ms  Retries 10
IPN884I  RWin: 65534
IPN448I  ID: *Internal001  Link ID: *Internal
IPN449I  IP Address: 192.168.1.161 Mask: 255.255.255.0
IPN450I  Net: 192.168.1.0 Subnet: -- Host: 0.0.0.161
IPN875I  MTU: 32767 Max Seg: 32727 Pulse: 60s
IPN876I  SYN Retran: 100ms Data Retran: 100ms Fixed: Yes
IPN877I  Retran Min: 100ms Max: 100ms
IPN882I  Retry Delay: 100ms  Retries 10
IPN884I  RWin: 65534
IPN448I  ID: LOCAL  Link ID: LINK3172, 0
IPN449I  IP Address: 192.168.1.0 Mask: 255.255.255.0
IPN450I  Net: 192.168.1.0 Subnet: -- Host: --
IPN875I  MTU: 5040 Max Seg: 5000 Pulse: 60s
IPN876I  SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
```

(continued)

```
IPN877I  Retran Min: 500ms Max: 2000ms
IPN882I  Retry Delay: 500ms  Retries 50
IPN884I  RWin: 65535
IPN448I  ID: DEFAULT  Link ID: LINK3172, 0
IPN449I  IP Address: 0.0.0.0 Mask: 255.255.255.0
IPN450I  Net: -- Subnet: -- Host: --
IPN537I  Gateway IP Address: 192.168.1.1
IPN875I  MTU: 0 Max Seg: 1400 Pulse: 60s
IPN876I  SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
IPN877I  Retran Min: 500ms Max: 2000ms
IPN882I  Retry Delay: 500ms  Retries 50
IPN884I  RWin: 65535
```

Example 2

```
 query route,ipaddr=192.168.1.66
IPN253I << TCP/IP Routes >>
IPN449I  IP Address: 192.168.1.66 Mask: 255.255.255.0
IPN450I  Net: 192.168.1.0 Subnet: -- Host: 0.0.0.66
IPN448I  ID: LOCAL  Link ID: LINK3172, 0
IPN449I  IP Address: 192.168.1.0 Mask: 255.255.255.0
IPN450I  Net: 192.168.1.0 Subnet: -- Host: --
IPN875I  MTU: 5040 Max Seg: 5000 Pulse: 60s
IPN876I  SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
IPN877I  Retran Min: 500ms Max: 2000ms
IPN882I  Retry Delay: 500ms  Retries 50
IPN884I  RWin: 65535
```

Notes

The following notes apply to this command:

- You can use the QUERY ROUTES command with the IPADDR parameter to help determine which link TCP/IP for VSE uses for that IP address. The routing table is searched in order, and the first matching entry is displayed. This is the same algorithm that TCP/IP for VSE uses when it needs to send a packet into the network.
- Routes whose IDs begin with "*" were automatically generated to control internal routing of datagrams within the stack.

Related Commands

DEFINE MASK
Creates a subnet mask for a particular network.

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

QUERY SECURITY

The QUERY SECURITY command displays the current security status of the TCP/IP FOR VSE partition.

Syntax

```
Query SECURity [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query security
IPN253I << TCP/IP Security Settings >>
IPN750I   Security Processing: Enabled
IPN750I   ARP Checking:   Disabled
IPN750I   IP Address Checking: Disabled
IPN751I   Exit data:   Undefined
IPN750I   Automatic Security:   Enabled ARP=   BLOCKIP=Y FTPC=
 FTPD= ICMP=Y IPAV=   WEBL=
IPN750I   Security Exit:   Undefined
IPN750I   Batch Security:   Disabled
IPN752I   Security Mode: Fail Log: Fail Dump: Fail
```

Related Commands

ASECURITY
Configures the Automatic Security Exit.

QUERY USERS
Displays a list of defined user IDs.

SECURITY
Controls the TCP/IP FOR VSE security functions.

QUERY STATISTICS (or QUERY STATS)

The QUERY STATISTICS command displays the TCP/IP FOR VSE statistics. TCP/IP FOR VSE maintains counters for data volumes and transfer events. This command is issued automatically during system shutdown.

QUERY STATISTICS can also be used to display link driver statistics for an OSA Express adapter. QUERY STATS is a synonym for QUERY STATISTICS.

Syntax

`Query STATistics [,SYSLST] [,LINKid=`*`nnnn`*`[,RESET]]`

Arguments

`SYSLST`

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

`LINKid=`

Specifies the link ID of an OSA Express adapter. This option causes the command to display statistics for this link only. The RESET option resets the counters for this link after the statistics are displayed.

Example 2 shows sample statistics for an OSA Express adapter.

Example 1

```
 query statistics
IPN253I << TCP/IP Operational Statistics >>
IPN516I FTP Daemons.................................1
IPN516I - Cur. Active...............................0
IPN516I - Max. Active...............................1
IPN516I - Cur. Buffers..............................0
IPN516I - Max. Act Buff.............................0
IPN516I Telnet Daemons..............................0
IPN516I - Current Active............................0
IPN516I - Maximum Active............................0
IPN516I - Current Buffers...........................0
IPN516I - Max Act Buffers...........................0
IPN516I LP Daemons..................................2
IPN516I HTTP Daemons................................0
IPN516I FTP Sessions................................2
IPN516I Telnet Sessions.............................0
IPN516I LPR Requests................................0
IPN516I HTTP Requests...............................0
IPN516I TCP Inbound Reject..........................0
IPN516I FTP Files Sent..............................2
IPN516I FTP Files Received..........................0
```

(continued)

```
IPN516I FTP File Bytes Sent.....................65,001
IPN516I FTP File Bytes Recv.....................0
IPN516I Telnet Bytes Sent.......................0
IPN516I Telnet Bytes Recv.......................0
IPN516I TCP Bytes Sent.........................67,659
IPN516I TCP Bytes Received.....................3,054
IPN516I UDP Bytes Sent..........................0
IPN516I UDP Bytes Received......................0
IPN516I IP Bytes Sent..........................78,395
IPN516I IP Bytes Received......................12,238
IPN516I Storage Releases........................0
IPN516I Received Blocks.........................43
IPN516I - Inbound Datagrams.....................59
IPN516I - Non-IP................................4
IPN516I - Miss Routed IP........................0
IPN516I - Discarded UDP......... ................0
IPN516I - Unsupported ICMP........ ..............0
IPN516I - Unsupported IGMP......... .............0
IPN516I - Unsup. Protocol.......... .............0
IPN516I - Connect Reject............ ............0
IPN516I - TCP Checksum Err............ ...........0
IPN516I - IP Checksum Err............. ........... 0
IPN516I - UDP Checksum Err.............. ..........0
IPN516I - Datagram len Err.............. .........0
IPN516I - ARPs.......................... .........2
IPN516I - ARP Requests..................... .....0
IPN516I Transmitted Blocks.....................115
IPN516I - Outbound Datagrams....................94
IPN516I - ARP Requests..................... .....0
IPN516I - ARP Replies...................... .....1
IPN516I VERCHECK................................1
IPN516I IPMSSG..................................1
IPN516I MSKELIP.................................1
IPN516I CLIENTD.................................1
IPN516I IPNAFTP.................................2
```

Example 2 The key fields in this example are explained in the table below.

```
MSG F3,DATA=QUERY STATS,LINKID=OSA854
AR 0015 1I40I  READY
F3 0111 0003: IPL613I OSA EXPRESS STATISTICAL SUMMARY FOR LINK OSA854
F3 0111 0003: IPL614I  MTU: 1500 INTERVAL: 7:07:43.356
F3 0111 0003: IPL615I  RECEIVED BLOCKS.............122,143
F3 0111 0003: IPL615I  RECEIVED BYTES............129,246,830
F3 0111 0003: IPL615I  RECEIVED BYTES, LARGE......127,762,068
F3 0111 0003: IPL615I  SEND BYTES................109,642,920
F3 0111 0003: IPL615I  SEND BLOCKS...................120,287
F3 0111 0003: IPL615I  SEND, FULL BLOCK..............2,930
F3 0111 0003: IPL615I  SEND, BUFFERED...............116,654
F3 0111 0003: IPL615I  SEND, CLEAR..................51,783
F3 0111 0003: IPL615I  SEND, BUSY.....................703
F3 0111 0003: IPL615I  SEND, MAX BUFFER.............24,000
F3 0111 0003: IPL615I  BUSY MODE.....................0
F3 0111 0003: IPL615I  BUSY MODE, LONGEST.............0
```

Field	Explanation (from *TCP/IP FOR VSE Messages* manual)
RECEIVED BLOCKS	Total receive count.
RECEIVED BYTES	Total bytes received.
RECEIVED BYTES, LARGE	Total bytes received in datagrams over 576 bytes.
RECEIVED BLOCKS, LARGE	Total received blocks over 576 bytes long.
SEND BYTES	Total bytes sent.
SEND BLOCKS	Total blocks sent.
SEND, FULL BLOCK	Total times the adapter accepted a block and reported that it had transferred its buffer.
SEND, BUFFERED	Total times the adapter accepted a block and indicated that it had been buffered.
SEND, BUSY	Total times the adapter rejected a block because all of its buffers were full. This count includes all retry attempts.
SEND, MAX BUFFER	Maximum bytes buffered before the buffer was shipped.
SEND, CLEAR	Total times the buffer was manually cleared.
BUSY MODE	Total times the adapter entered busy mode. This value is incremented each time a SEND initially fails because the device was busy. These failed attempts were added to the chain of datagrams awaiting a successful transmission.
BUSY MODE, LONGEST	Largest number of consecutive times the adapter reported busy.

Related Commands

DEFINE LINK
Defines link parameters.

QUERY ISTATS
Displays statistics detailing internal stack functions.

QUERY STOR

The QUERY STOR command allows you to monitor storage use in the TCP/IP partition. The detail provided far exceeds that produced by the VSE GETVIS command.

Syntax

Query STOR [,SYSLST] [,SAVe|TRend|MAXimum]

Query STOR [,SYSLST] ,SPid=*name* [,DUmp={Yes|No}]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

SAVe
When specified, the current totals from the display are saved for future comparison.

TRend
When specified after a SAVE, the displayed data shows the changes since the SAVE.

The values displayed may be "signed," indicating either an increase or a decrease in storage.

MAXimum
This option causes a display of the high-water marks for each subpool and value.

SPid=
Specifies a particular subpool ID. When specified, the display summarizes each storage element in the named subpool. The number of lines written can be voluminous.

DUmp=
Controls whether the contents of the specified storage area(s) are dumped.

Yes
A formatted storage dump is produced. This data is not displayed on a console but is routed to eligible printer files.

No
Storage is not dumped. This is the default.

Example 1

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in Examples 1 through 4.

```
 query stor
IPN253I << TCP/IP Current Storage Information >>
IPN921I Map Pages: 10 Subpools: 57
IPN925I -----24 bit---- -----31 bit----
IPN930I SPID Elem Storage Elem Storage Requests
IPN922I ARBLOK 0 0K 2 1K 0k
IPN922I BEGINS 0 0K 1 1K 0k
IPN922I CCBLOK 0 0K 13 173K 0k
IPN922I CHBLOK 0 0K 0 0K 0k
IPN922I CLBLOK 0 0K 1 1K 0k
IPN922I COBLOK 1 1K 1 1K 0k
IPN922I DIBLOK 1 1K 3 265K 0k
IPN922I DRBLOK 0 0K 3 1K 0k
IPN922I DWRKAR 1 1K 1 8K 0k
IPN922I DYNAM 0 0K 0 0K 0k
```

Example 2

```
 query stor,trend
IPN253I << TCP/IP Storage Trends >>
IPN921I Map Pages: 10 Subpools: 57
IPN925I -----24 bit---- -----31 bit----
IPN930I SPID Elem Storage Elem Storage Requests
IPN922I ARBLOK +0 +0K +2 +0K 0k
IPN922I BEGINS +0 +0K +1 +0K 0k
IPN922I CCBLOK +0 +0K +13 +172K 0k
IPN922I CLBLOK +0 +0K +1 +0K 0k
IPN922I COBLOK +1 +0K +1 +0K 0k
IPN922I DIBLOK +1 +0K +3 +264K 0k
IPN922I DRBLOK +0 +0K +3 +0K 0k
IPN922I DWRKAR +1 +0K +1 +8K 0k
IPN922I DYSTOR +6 +1K +10 +11K 570k
IPN922I EMAIL +0 +0K +1 +0K 0k
IPN922I EXTTYP +0 +0K +1 +8K 0k
IPN922I FIBLOK +13 +3K +0 +0K 0k
IPN922I FTBLOK +1 +0K +0 +0K 0k
IPN922I FTBXWK +0 +0K +1 +2K 0k
```

Example 3

```
 query stor,spid=rtblok
IPN253I << TCP/IP Storage Information >>
IPN926I Address: 5D7000 Length: D8
IPN927I Macid: IPNET15 Locate: 805593BE
IPN928I Usage: ID:
IPN929I Allocated at: 09/24/2016 14:28:41
IPN926I Address: 5D7100 Length: D8
IPN927I Macid: IPNET15 Locate: 805593BE
IPN928I Usage: ID:
IPN929I Allocated at: 09/24/2016 14:28:41
```

Example 4

```
 query stor,spid=rtblok,dump=yes
IPN253I << TCP/IP Storage Information >>
IPN926I Address: 5D7000 Length: D8
IPN927I Macid: IPNET15 Locate: 805593BE
IPN928I Usage: ID:
IPN929I Allocated at: 09/24/2016 14:28:41
IPN861I Storage Dump Storage Dump, SPID=RTBLOK
005D7000 000000 D9E3C2D3 D6D20000 5CC995A3 85999581 *RTBLOK..*Interna|........\.......*
005D7010 000010 93404040 40404040 005D7100 40404040 *l .).. |.@@@@@@@.|q.@@@@*
005D7020 000020 40404040 40404040 40404040 40404040 * |@@@@@@@@@@@@@@@@*
    Line(s) 005D7030 (000030) through 005D704F (00004F) same as above
005D7050 000050 40404040 40404040 40404040 7F000000 * "...|@@@@@@@@@@@@....*
005D7060 000060 00000000 5CC995A3 85999581 93404040 *....*Internal |....\........@@@*
005D7070 000070 40404040 005D5000 00000000 00000000 * .)&.........|@@@@.|P.........*
005D7080 000080 00000000 00000000 7F000000 00007FFF *........".....".|................*
005D7090 000090 00007FD7 00000064 00000064 80000000 *..."P.............|.......d...d....*
005D70A0 0000A0 00000064 00000064 0000000A 0000003C *.................|...d...d.......<*
005D70B0 0000B0 0000FFFE 00000064 D5000000 7F000000 *........N..."...|.......d........*
005D70C0 0000C0 B22C41E0 00000000 00000000 00000000 *...\.............|.,A.............*
005D70D0 0000D0 00000000 00000000          *........ |........@@@@@@@@*
```

QUERY TASKS

The QUERY TASKS command displays the currently active pseudo tasks. All work performed by TCP/IP FOR VSE is assigned to internal pseudo tasks. Pseudo tasks are similar to true VSE tasks, but they are controlled and dispatched by the TCP/IP FOR VSE engine.

Syntax

```
Query TASKs [,SYSLST] [,ID=hexnum|NAMe=name]
          [,EXTended]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
Only the pseudo task with the specified task number is displayed.

NAMe=
All pseudo tasks executing the named phase are displayed.

EXTended
Use when requested by CSI Technical Support. Displays extended information about tasks such as the total number of IBBLOKS; the storage queued on busses; whether a task is dispatchable; waiting ECBs and "real ECBs"; and currently running timers. This information is only of use to CSI Technical Support.

Example 1

See the *TCP/IP FOR VSE Messages* manual for details on each message number in this example.

```
 query tasks
IPN253I << TCP/IP Pseudo Tasks >>
IPN468I  0000: DRIVER   D:        0  000000
IPN468I  0002: IPNL127  D:      349  000132 IPNL127
IPN468I  0004: FTPDAEMN D:        5  0064B6 FTPDAEMN
IPN468I  0005: LPD      D:        1  0005A6 LPD
IPN468I  0006: IPNLRAWX D:      133  000132 IPNLRAWX
IPN468I  0007: IPNIRAW1 D:      415  0004BA IPNIRAW1
IPN468I  0008: IPNIRAW2 D:        1  00092A IPNIRAW2
IPN468I  0009: IPNIIIP  D:      415  000498 IPNIIIP
IPN468I  000A: IPNIOIP  D:      433  00088C IPNIOIP
IPN468I  000B: IPNIPRE1 D:      433  000270 IPNIPRE1
IPN468I  000C: IPNIPRE2 D:      433  000A76 IPNIPRE2
IPN468I  000D: IPNIGARB D:    1,277  00021E IPNIGARB
IPN468I  0002: IPNL127  D:      349  000132 IPNL127
IPN468I  0004: FTPDAEMN D:        5  0064B6 FTPDAEMN
IPN468I  0005: LPD      D:        1  0005A6 LPD
IPN468I  000E: IPNIARP  D:        6  00020C IPNIARP
```

(continued)

```
IPN468I  000F: IPNTITCP D:      410  000E9A IPNTITCP
IPN468I  0010: IPNTOTCP D:      428  00041C IPNTOTCP
IPN468I  0011: IPNUIUDP D:        1  000716 IPNUIUDP
IPN468I  0012: IPNUOUDP D:        1  0002F4 IPNUOUDP
IPN468I  0013: IPNCICMP D:        6  000838 IPNCICMP
IPN468I  0014: IPNCOCMP D:        6  00037C IPNCOCMP
IPN468I  0015: IPNGIGMP D:        1  00034E IPNGIGMP
IPN468I  0016: IPNGOGMP D:        1  000222 IPNGOGMP
IPN468I  0017: IPNLIEEE D:        1  0002E6 IPNLIEEE
IPN468I  0018: IPNLOEEE D:        1  00028C IPNLOEEE
IPN468I  001F: IPNTYTCP D:        1  006AF6 IPNTYTCP
IPN468I  0020: IPNLOSA2 D:      223  000402 IPNLOSA2
IPN468I  0025: CLIENTD  D:   15,252  00060A CLIENTD
IPN468I  0045: FTPX1000 D:        1  00213A FTPDAEMN
IPN468I  0046: IPNTYTCP D:        1  006AF6 IPNTYTCP
```

Example 2

```
 query tasks,name=ipntytcp
IPN253I << TCP/IP Pseudo Tasks >>
IPN468I  001F: IPNTYTCP  D:       1  006AF6 IPNTYTCP
IPN468I  0046: IPNTYTCP  D:       1  006AF6 IPNTYTCP
```

Example 3

```
 QUERY TASKS,EXT
IPN253I << TCP/IP Pseudo Tasks >>
IPN468I  0000: DRIVER   D:        0  000000
IPN468I  0002: IPNL127  D:        1  000132 IPNL127
IPN471I        ECB: 8066500C Content: 813DDC00
```

Related Commands

RESUME
Resumes processing a suspended pseudo task.

SUSPEND
Halts processing by a pseudo task.

QUERY TELNETDS

The QUERY TELNETDS command displays the currently defined Telnet daemons.

Syntax

Query TELnetds [,SYSLST] [,ID=*name*]

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=
Specifies the ID from the DEFINE TELNETD command that created the daemon you want to display. If omitted, all Telnet daemons are displayed.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query telnetds,id=cf42
IPN469I << TCP/IP TELNET Daemons >>
IPN470I ID: CF42
IPN471I Terminal: TELNLU42 Target: PRODCICS
IPN472I Port: 1042 Driver: TELNETD
IPN473I Menu:
IPN549I Logmode2: S3270 Logmode3: D4B32783
IPN356I Logmode4: D4B32784 Logmode5: D4B32785
IPN350I Current Status: Inactive

 query telnetds,id=lu05
IPN469I << TCP/IP TELNET Daemons >>
IPN470I ID: LU05
IPN471I Terminal: TELNLG05 Target: DBDCCICS
IPN472I Port: 23 Driver: TELNETD
IPN473I Menu:
IPN549I Logmode2: S3270 Logmode3: D4B32783
IPN356I Logmode4: D4B32784 Logmode5: D4B32785
IPN350I Current Status: Active
IPN351I Current IPaddr: 192.168.0.2
IPN352I Current Applid: DBDCCICS
IPN354I Current Logmod: S3270
```

Related Commands

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

DELETE TELNETD
Terminates a TN3270 or TN3270E daemon.

QUERY TLSDS

The QUERY TLSDS command displays the status of SSL/TLS daemons.

Syntax

Query TLSDs [,SYSLST] [,ID=*id*]

Arguments

SYSLST

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

ID=

Specifies the ID of the TLS daemon assigned by the DEFINE TLSD command. If this option is omitted, all TLS daemons are displayed.

Example

```
 query tlsd
IPN253I << TCP/IP TLS Daemons >>
IPN617I ID: TLS01 Cipher: 08090A2F35
IPN618I Port: 992 Passport: 992 Type: Server
IPN619I Driver: SSLD Minimum version: 0300
```

Related Commands

DEFINE TLSD

Creates an SSL/TLS daemon.

QUERY TRACES

The QUERY TRACES command lists all currently active traces.

Syntax

```
Query TRACes [,SYSLST]
```

Arguments

SYSLST
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 query traces
IPN253I << TCP/IP Traces >>
IPN534I ID: TRACE1 Type: IBBLOK Kind: TCP
IPN374I Count: 51 Max: 500 IP: 192.168.1.66 Port: 0 Num: 0
IPN534I ID: TRACE2 Type: Socket Kind: ALL Scope: All
IPN374I Count: 107 Max: 500 IP: 0.0.0.0 Port: 0 Num: 0
```

Notes

All traces listed by the QUERY TRACES command are in progress and are accumulating storage.

Related Commands

DEFINE SOTRACE
Starts a Socket Trace.

DEFINE TRACE
Starts a Datagram Trace.

DELETE TRACE
Terminates a trace and free its storage.

DUMP
Performs a formatted dump of various TCP/IP control blocks.

QUERY TRANSLATIONS

The QUERY TRANSLATIONS command displays the currently available translation tables and related options.

Syntax

`Query TRANslations [,SYSLST]`

Arguments

`SYSLST`
The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query trans
IPN253I << TCP/IP Translate Tables >>
IPN542I SYSTEM
IPN542I US_ENG_03
IPN542I OS_02 (Default)
```

Related Commands

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

SET TELNET_TRANSLATE
Sets the name of the translate table that will be used with Telnet (not TN3270) connections.

QUERY TRUSTED

The QUERY TRUSTED command displays a list of IP addresses that have been marked as "trusted."

Syntax

Query TRUSted [,SYSLST]

Arguments

SYSLST

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query trusted
IPN253I << TCP/IP Trusted IP Addresses >>
IPN127I 192.168.1.66
IPN127I 192.168.1.1
12.182.34.168
```

Related Commands

TRUST

Establishes that an IP address is trusted and suspicious activity is to be ignored.

QUERY USERS

The QUERY USERS command lists the contents of the user ID table. This table is constructed by using the DEFINE USER command.

Syntax

Query USERs [,SYSLST] [,NAMe=*name*]

Arguments

SYSLST

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console. This argument is positional and must be placed immediately after the command.

NAMe=

Displays a specific entry based on the user ID. User IDs are not case sensitive. If more than one entry matches the user ID (that is, if there is a duplicate entry), all matching entries are displayed. The security processor recognizes only the first entry.

Example

```
 query users
IPN253I << TCP/IP User IDs >>
IPN475I User ID: TBDBITL
IPN883I Valid for: Web
IPN475I User ID: UPLOAD
IPN883I Valid for: *All*
IPN475I User ID: GUEST
IPN883I Valid for: *All*
IPN475I User ID: AUTOFTP
IPN883I Valid for: FTP, LPR
IPN475I User ID: EWF
IPN883I Valid for: FTP, LPR, Web, Telnet
```

Related Commands

DEFINE USER

Creates a user ID and password.

DELETE USER

Removes a user ID and password entry.

QUERY VERSIONS

The QUERY VERSIONS command displays the current version and maintenance level of all TCP/IP FOR VSE phases.

Syntax

`Query VERsions [,SYSLST]`

Arguments

`SYSLST`

The query results are sent to SYSLST only. If this option is omitted, the results are sent to SYSLST and the console.

Example

```
 query versions
IPN253I << TCP/IP Version Information >>
IPN209I Service Pack F (APAR PK33472) has been applied. Pack status is GA
IPN115I Fixes applied: 002, 003, 006 (Test), 008 (Test), 009 (Test), 103, 105
IPN115I Fixes applied: 106, 109, 114, 115, 116, 118, 122, 123, 128, 133, 134
IPN115I Fixes applied: 135, 143, 144, 145, 146, 147, 148, 149, 150, 152, 154
IPN115I Fixes applied: 155, 159, 160, 163, 164, 165, 172, 174, 177, 178, 179
IPN115I Fixes applied: 180, 185, 186, 188, 189, 191, 192, 193, 195, 196, 197
IPN115I Fixes applied: 198, 211, 213, 218, 220, 235, 238, 240, 254, 257, 261
IPN115I Fixes applied: 263, 265, 266, 273, 274, 276, 278, 279, 281, 282, 283
IPN115I Fixes applied: 285, 286, 289, 290, 291, 292, 295, 296, 297, 298, 303
IPN115I Fixes applied: 332, 341, 342, 349, 354, 362, 366, 384, 391, 396, 397
IPN115I Fixes applied: 399, 502, 504, 506, 507, 509, 510, 511, 512, 513, 517
IPN115I Fixes applied: 519, 520, 521, ASOCKET at ZP15F172, CLIENTD at ZP15F512
IPN115I Fixes applied: CMDEXEC at ZP15F188, CSOCKET at ZP15F148, FTPDAEMN at ZP15F366
IPN115I Fixes applied: IPDRIVER at ZP15F114, IPMSSG at ZP15F177, IPNADNSC at ZP15F509
IPN115I Fixes applied: IPNAFTPC at ZP15F292, IPNET at ZP15F114, IPNFPOWR at ZP15F504
IPN115I Fixes applied: IPNIGARB at ZP15F114, IPNLRAWX at ZP15F009
IPN115I Fixes applied: IPNRBSDC at ZP15F286, IPNTYTCP at ZP15F192
IPN115I Fixes applied: IPNUIUDP at ZP15F145, IPSERVIB at ZP15F257
IPN115I Fixes applied: MSKELIP at ZP15F174
IPN111I ASOCKET 01.05 F 02/19/08 09.49 ZP15F172
IPN111I CLIENTD 01.05 F 02/19/08 09.50 ZP15F512
IPN111I CMDEXEC 01.05 F 03/03/08 08.07 ZP15F188
IPN111I CMDPARS 01.05 F 03/01/08 10.10
IPN111I CMDSTASK 01.05 F 02/29/08 10.38
IPN111I CSOCKET 01.05 F 03/02/08 23.58 ZP15F148
IPN111I FTPDAEMN 01.05 F 02/29/08 09.58 ZP15F366
. . .
IPN111I IPSPINCP 01.05 F 02/25/08 21.24
IPN111I IPSTORX 01.05 F 02/19/08 10.10
IPN111I LOCKMGRX 01.05 F 04/19/07 21.17
IPN111I LPD 01.05 F 02/19/08 10.10
IPN111I MSKELIP 01.05 F 03/02/08 23.58 ZP15F174
IPN111I SOCKPASS 01.05 F 02/21/08 00.26
IPN111I SOCTRACE 01.05 F 02/11/08 11.26
IPN111I VERCHECK 01.05 F 03/03/08 22.56
```

Related Commands

QUERY PROGRAMS

Displays program phases being used by TCP/IP, their characteristics, their memory locations, and the library from which each was loaded.

QUIESCE

The QUIECE command disables application processing by preventing new connections. Existing connections are not disturbed.

Syntax

QUIESce {ON|Off}

Arguments

ON

When QUIESCE ON is in effect, the stack refuses requests to establish new connections; however, existing connections continue to operate normally.

Off

When QUIESCE OFF is issued, normal processing resumes.

Example

```
 quiesce on
IPN225I TCP/IP quiescing; connection requests are rejected
...
IPI515I TCP/IP Stack QUIESCE progress
IPI516I 1 connections active on port 21
IPI516I 4 connections active on port 23
...
IPI515I TCP/IP Stack QUIESCE progress
IPI516I 2 connections active on port 23
...
IPI517I TCP/IP processing has been quiesced

 quiesce off
IPN226I TCP/IP QUIESCE is canceled. Normal processing resumes.
```

Notes

The following notes apply to this command:

- When QUIESCE ON is in effect, applications issuing socket OPEN requests are notified of “system shutdown,” and incoming connection requests are rejected with a RESET. Depending on the host and application, it may not be possible to simply resume processing.
- Whenever a QUIESCE ON condition exists, a periodic console display automatically shows the progress made toward the quiesced state. This display includes a count of active connections by local port number. Once the system is quiesced, a “nag” message is issued until either QUIESCE OFF is specified or TCP/IP FOR VSE is shut down.

Related Commands

QUERY CONNECTIONS
Displays the status of one or more connections.

QUERY TASKS
Displays a list of pseudo tasks.

SHUTDOWN
Terminates processing and shuts down the stack.

RAPTRAC

The RAPTRAC command is used to record significant events that occur during TCP/IP processing. Its output can assist in diagnosing problems.

Syntax

RAPTRAC {START|REPORT|STOP}

Arguments

START
Enables tracking of significant events based on settings contained in the IPDSEVTB phase.

REPORT
Prints a single line to SYSLST for each captured event from the current event buffers. The messages are issued in chronological order.

STOP
Disables the gathering of event data.

Example

```
 raptrac start
IPN694I RapTrac started
```

Notes

The following notes apply to this command:

- RAPTRAC's output consists of CSI internal diagnostics. Use this command only when requested by CSI Technical Support to troubleshoot a problem.
- This command requires about 512K of additional 31-bit partition GETVIS storage in the TCP/IP partition.
- The default IPDSEVTB phase is configured to log all events. CSI Technical Support may supply a custom IPDSEVTB that is specific to the problem being investigated. The custom settings can also cause the RAPTRAC report to be produced automatically when a specific event occurs.
- The external batch FTP client (FTPBATCH) can also use events but requires different commands (SET DIAGNOSE EVENTS and STATUS EVENTS). See the *TCP/IP FOR VSE User Guide* for details.

- DIAGNOSE commands are normally used to continuously capture information related to a specific function, but the voluminous messages issued to SYSLOG and SYSLST may be unrelated to the problem experienced at the customer site. These additional messages may degrade overall system performance and obfuscate the actual problem being analyzed. It can be more efficient to capture problem data using RAPTRAC because it continuously gathers events and then generates data lines (a REPORT) in chronological sequence that reveal the interactions among multiple processes when the problem occurred. The thousands of interactions that can occur in less than a second can be identified and analyzed.

Related Commands

DIAGNOSE
Enables diagnostic messages.

QUERY DIAGNOSE
Displays the diagnostic settings in effect.

RECORD

The RECORD command controls the recording of information about TCP/IP FOR VSE internal pseudo tasks.

Syntax

RECord {ON|OFF}

Arguments

ON
A completion record is written to SYSLST following the completion of each pseudo task.

OFF
No information is written when a pseudo task completes. This is the default.

Example

```
 record off
IPN268I RECORD now set to OFF
```

Notes

The following notes apply to this command:

- The format of the data being recorded is subject to change with each TCP/IP for VSE release.
- The recording is performed by way of message IPN878I. See the *TCP/IP FOR VSE Messages* manual for the format of this message.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

REDEFINE

The REDEFINE command reissues a DEFINE command without requiring its parameters to be re-entered.

Syntax

REDefine START

REDefine *type* ,*id* [,*parameters*]

Arguments

START

The REDEFINE facility is dormant until started. This means that no commands are stored and no 31-bit storage is consumed. Only DEFINE commands entered after REDEFINE START is issued can be referenced by the REDEFINE command.

If you want to use REDEFINE, place a REDEFINE START command in your initialization deck ahead of the first DEFINE that you want to store.

type

The type of daemon to be redefined.

GPSd

A GPS daemon will be redefined.

TELnetd

A Telnet daemon will be redefined. See the note,below, about using the COUNT= parameter.

FTPd

An FTP daemon will be redefined.

HTTpd

An HTTP daemon will be redefined.

id

This is the value coded for ID= in the original DEFINE command.

parameters

When you issue a REDEFINE command, all parameter values default to those used in the last-issued DEFINE for the particular daemon identified by *id*. You may override any or all parameters, including ID=. If you do override ID=, a “new” daemon is started.

Example

```
 define gpsd,id=gps1,storage=gps.save,ipaddr=rmt1,printer=prt1
GPS900I GPS1 GPS Daemon Starting
GPS917I GPS1 Waiting for BIND
GPS922I GPS1 GPS Shutting down
GPS924I GPS1 GPS Shutdown complete

 redefine gps,gps1
GPS900I GPS1 GPS Daemon Starting
GPS917I GPS1 Waiting for BIND
```

Notes

The following notes apply to this command:

- If you used the COUNT= parameter to define multiple daemons with a single DEFINE command, only this command was stored. Although you cannot delete and then redefine a daemon so created, you can reference the original DEFINE command by using ID= to reference the pattern.
- After a daemon is REDEFINED, any parameter alterations are stored and used for the next REDEFINE of the same daemon.
- If you alter the ID= parameter to define a different daemon, the parameters in effect are stored for a future REDEFINE of the new daemon. The original “pattern” is not affected.

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE GPSD
Creates a General Print Server daemon.

DEFINE HTTPD
Creates a Hypertext Transfer Protocol (web server) daemon.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

RELOAD

The RELOAD command refreshes internal tables without the need to cycle the stack partition. It may be used to reload library members that are maintained in storage.

Syntax

RELOad {EXTtypes|HACKlist|TERMtypes}

Arguments

EXTtypes
The External Types table will be reloaded from EXTTYPES.L

HACKlist
The "Hack List" table will be reloaded from HTTPHACK.L

TERMtypes
The Terminal Types table will be reloaded from TERMTYPE.L

Example

```
 reload exttypes
IPA617I Processing EXTTYPES.L load request
IPA616I External types table has been loaded.

 reload hacklist
IPA617I Processing HTTPHACK.L load request
IPA616I Hacker-Attack table has been loaded.

 reload termtypes
IPA617I Processing TERMTYPE.L load request
IPA616I Terminal Types Table has been loaded.
```

Notes

The following notes apply to this command:

- The Hack List table is used by the HTTP daemon to identify attempts to hack the system. This is done by identifying certain key "requests" that would never be sent to a VSE-based server. Such requests are a tip-off that the requestor is performing random scans to locate a victim. Examine the contents of the distributed HTTPHACK.L member for details on adding or removing key strings.

- The External Types table is used by the HTTP and FTP daemons and the Email client to determine file characteristics. This is done by matching the file's name extension with a file type entry in the table. Each file type is mapped to an FTP transfer type with default transfer values (such as for BLKSIZE). Each entry is also mapped to a MIME content type. Information on modifying this table is in the distributed EXTTYPES.L member and the *TCP/IP FOR VSE Installation Guide* (chapter 6, "Configuring FTP Clients and Daemons").

- The Terminal Types table is used to map terminal type names (as presented by a TN3270E client) to an appropriate VTAM logmode. For more information, consult the TERMTYPE.L member.

Related Commands

DEFINE FTPD
Creates a File Transfer Protocol daemon.

DEFINE HTTPD
Creates a Hypertext Transfer Protocol (web server) daemon.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon

RESUME

The RESUME command is used to resume execution of a pseudo task that has been suspended by means of the SUSPEND command.

Syntax

RESume *tasknum*

Arguments

tasknum
Specifies the hexadecimal task number of the task to be resumed.

Example

```
 resume 000d
IPN477I Task 000D has resumed in phase IPNIGARB. Disp: 3,855
```

Related Commands

QUERY TASKS
Displays a list of pseudo tasks.

SUSPEND
Halts processing by a pseudo task.

SDOPEN_EXTRA

The SDOPEN_EXTRA command causes TCP/IP FOR VSE to check whether a sequential disk (SD) file has been properly opened by a site's OEM SD file manager before permitting it to be transferred. If the SD file manager fails to properly prepare ("pre-open") the file, TCP/IP FOR VSE will not attempt to transfer it.

This command should only be used by sites that have encountered a problem sending every SD file controlled by their disk manager software. Otherwise, it should be set to its default (OFF).

Syntax

SDOPEN_EXTRA {ON|OFF}

Arguments

ON
An SD file's open status is checked for before it is transferred.

OFF
An SD file's status is not checked before it is transferred. This is the default.

Example

```
 SDOPEN_EXTRA ON
IPN268I SDOPEN now set to ON
```

SECURITY

The SECURITY command controls what security, if any, is provided for TCP/IP FOR VSE access.

Syntax

```
SECURity [ON|OFF] [,BATCH={ON|OFF}] [,PHASE=member]
         [,XDATA=string] [,ADATA=string]
         [,ASMDATE=string] [,ASMTIME=string]
         [,VERSION=string] [,AUTO={ON|OFF}]
         [,EXIT={ON|OFF}] [,ARP={ON|OFF}]
         [,MODE={WARN|FAIL}] [,LOGGING={ALL|FAIL|NONE}]
         [,DUMP=ALL|FAIL|NONE}] [,LOCK]
```

Arguments

ON
Security processing is enabled globally.

OFF
Security processing is disabled globally. Caution: Be very careful with this option.

BATCH=
This parameter controls whether FTPBATCH processing will be under control of the stack's security processing, including calls to the security exit(s). This prevents using FTPBATCH as a convenient way of bypassing security.

ON
Security processing is enabled for FTPBATCH.

OFF
Security processing is disabled for FTPBATCH.

PHASE=
Specifies the name of an optional installation-supplied security exit. This value provides the phase name only. The phase is not loaded nor is its existence verified until EXIT=ON is specified.

XDATA=
Specifies a 40-byte character string to be passed to the installation-supplied security exit each time it is called.

ADATA=
Specifies a 40-byte character string to be passed to the Automatic security exit each time it is called.

ASMDATE=
Specifies a 1- to 8-byte character string to be compared with the assembly date of the installation-supplied security exit. This can be used to ensure that the exit has not been tampered with.

ASMTIME=
Specifies a 1- to 8-byte character string to be compared with the assembly time of the installation-supplied security exit. This can be used to ensure that the exit has not been tampered with.

VERSION=
Specifies a 1- to 8-byte character string to be compared with the version number of the installation-supplied security exit. This can be used to ensure that the exit has not been tampered with.

AUTO=
This parameter controls whether the Automatic Security Exit is enabled.

ON
Automatic security is enabled.

OFF
Automatic security is disabled.

EXIT=
This parameter controls the loading and enabling of the installation-supplied security exit.

ON
The installation-supplied security exit is loaded and its initialization routine is called.

OFF
The installation-supplied security exit's termination routine is called and the exit is then removed from storage.

ARP=
This parameter controls whether ARP requests are examined before being processed. This might be useful if there is concern about unauthorized access or misrouting of data on the local network segment.

ON
ARP requests are passed to the Automatic security exit (if enabled) and the installation-supplied security exit (if enabled) for validation.

OFF
ARP requests are not checked.

MODE=
This parameter determines the response level to security violations.

WARN
Security failures are treated as warnings, and all requests are allowed to complete. This is useful for testing security rules prior to enforcing them.

FAIL
Security failures result in the requested action being denied.

LOGGING=
Security requests may be logged minimally, aggressively, or not at all.

ALL
All security requests are logged.

FAIL
Failing security requests are logged.

None
Security requests are not logged unless the security exit explicitly indicates that logging should occur.

DUMP=
The Security eXit BLOcK (SXBLOK) contains all information on a particular security validation request. It is created by the process requiring authorization and is passed to all validation processing and exits. Once complete, the SXBLOK contains information that either permits or prevents the operation in question. This parameter permits you to dump a copy of a failed security request, either for logging or for debugging purposes.

ALL
The SXBLOK of failed security requests is dumped regardless of WARN or FAIL mode.

FAIL
The SXBLOK of failed security requests is dumped.

None
SXBLOKs are be dumped.

LOCK
Once issued, all security settings are locked to their current values. Security settings cannot be altered until the stack is cycled.

Example

```
 security on,auto=on,batch=on
IPN759I Security status change: Security Processing Enabled
IPN759I Security status change: Automatic Security Enabled
IPN759I Security status change: Batch Security Enabled
```

Related Commands

ACCESS
Controls access to VSE by IP address.

ASECURITY
Configures the Automatic Security Exit.

DEFINE USER
Creates a user ID and password.

QUERY SECURITY
Displays current security settings.

QUERY USERS
Displays a list of defined user IDs.

SEGMENT

The SEGMENT command segments the SYSLST file. This enables you to obtain a partial SYSLST file without shutting down TCP/IP FOR VSE.

Syntax

SEGment [NEW]

Arguments

NEW

Directs TCP/IP FOR VSE to issue the IPWSEGM macro instead of the SEGMENT macro.

Some customers have reported problems with IPWSEGM, so the default is to use the older SEGMENT macro. SEGMENT NEW can be used on systems running VSE/ESA 2.2 or later.

Example

```
 segment
IPN480I TCP/IP FOR VSE log has been segmented.
```

Notes

When SEGMENT is issued, the characteristics of the SYSLST file are reset to VSE defaults.

Related Commands

DEFINE LOG
Creates a system log file.

MESSAGE
Controls message suppression.

MODIFY LOG
Changes characteristics of a system log file.

QUERY LOGS
Displays available consoles and logs, along with their properties.

SEPARATOR_PAGES

The SEPARATOR_PAGES command controls whether POWER-generated page separators are output for various operations.

Syntax

SEParator_pages {OFF|ON|[-]Email|[-]FTp|[-]HTtp|[-]LPr}

Arguments

OFF
POWER-generated separator pages will not be included for any process. This is the default.

ON
Files read from POWER queues will contain POWER-generated separator pages for all processes

[-]EMail
Controls Email use of POWER-generated separator pages.

[-]FTp
Controls FTP use of POWER-generated separator pages.

[-]HTtp
Controls HTTP use of POWER-generated separator pages.

[-]LPr
Controls LPR use of POWER-generated separator pages.

Example

```
 separator_pages on
IPN249I Value for Separator Page set to on
```

Notes

The following notes apply to this command:

- Under the LPR/LPD protocol, the responsibility for producing separator pages belongs to the LPD that ultimately sends the data to a physical printer. Because many commonly available daemons do not support this feature, however, the standard separators produced by POWER can be passed through.
- Separator pages are valid only for POWER output and only if you request that POWER create them. See IBM manual *VSE/Enterprise Systems Architecture VSE Central Functions: VSE/POWER Administration and Operation* (SC33-6633-01) for more information about having POWER create separator pages.
- This command affects READ operations when you are extracting data from POWER. To generate separator pages when you are writing data to POWER, see the SET PAGE_COUNT command.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

SET PAGE_COUNT
Sets the number of POWER-generated separator pages.

SET AUTO_TIME

The SET AUTO_TIME command controls how often TCP/IP FOR VSE's event processing (as enabled by the DEFINE EVENT command) scans the POWER queues looking for new work.

Syntax

SET AUTO_TIME=*sec300*

Arguments

sec300

Numeric, 1500 through 18000, the number of 300th-second units.

The amount of time between scans of the POWER queues by the TCP/IP FOR VSE event timer.

Example

```
 set auto_time=1500
IPN268I AUTO_TIME now set to 1500 300th sec
```

Notes

The following notes apply to this command:

- See the description of the DEFINE EVENT command for more information about TCP/IP for VSE event processing.
- A large value saves CPU but makes TCP/IP FOR VSE take longer to detect new work on the POWER queues. A small value uses additional CPU but makes event processing more responsive to new work.

Related Commands

DEFINE EVENT

Monitors a POWER class for automatic report distribution.

QUERY EVENTS

Displays the status of automation processing.

QUERY OPTIONS

Displays the current values of modifiable parameters.

SET DEFAULT_DOMAIN

The SET DEFAULT_DOMAIN command specifies the common portion of the symbolic host names used in your installation. This value is added to the local names table (see Notes below).

Syntax

SET DEFAULT_DOMAIN=*string*

Arguments

string

A string to be used as the default domain. If an asterisk (*) is specified, the value is nullified.

Note: For email addresses, the EMAIL command's DESTINATION= setting takes precedence over this value.

Example

```
 set default_domain=csi-international.com
IPN268I DEFAULT_DOMAIN now set to CSI-INTERNATIONAL.COM
```

Notes

When an application issues a call to convert a symbolic name to an IP address, TCP/IP FOR VSE searches the symbolic names table. If no entry is found, the name is passed to the specified domain name server (DNS). If the DNS cannot resolve the name, then a period and the default domain are appended to the name, and the entire search process (cache checking through DNS lookup) is repeated using this composite name.

For instance, using the above example, if "helpdesk" cannot be resolved, an attempt will be made to resolve "helpdesk.csi-international.com."

Related Commands

EMAIL

Sets global default options for the EMAIL client.

DEFINE NAME

Associates a TCP/IP name with an address or a script file.

QUERY NAMES

Displays the contents of the symbolic names table created by DEFINE NAME.

QUERY OPTIONS

Displays the current values of modifiable parameters, including the addresses of the servers specified by SET DNS*n*.

SET DNS*n*

Specifies a DNS to be used for name resolution.

SET DIAGNOSE

The SET DIAGNOSE command controls where DIAGNOSE-issued messages are written. You can use this command to direct these messages to the TCP/IP FOR VSE console as well as to SYSLST.

Syntax

SET DIAGnose {NOCONSOLE|CONSOLE}

Arguments

NOCONSOLE
Messages produced in response to the DIAGNOSE command are written to SYSLST only. This is the default.

CONSOLE
Messages produced in response to the DIAGNOSE command are sent to both SYSLST and the console.

Example

```
 set diagnose console
IPN868I DIAGNOSE display on console is on
```

Notes

The following notes apply:

- If you do not have access to SYSLST and you are diagnosing a problem, then this command is an option that permits viewing diagnostics on the console. While it does have its uses, be aware that it has the potential to flood your console with messages, which might interfere with normal operations in a production environment.
- Output from the DIAGNOSE command can be voluminous. Use the SET DIAGNOSE CONSOLE command with caution.

Related Commands

DIAGNOSE
Enables diagnostic messages.

QUERY DIAGNOSE
Displays the current DIAGNOSE settings in effect.

SET DNS*n*

Each SET DNS*n* command specifies the IP address of a domain name server (DNS) that is used to resolve a host name to an IP address or an IP address to a host name.

Syntax

SET DNS*n*=*ip4addr*

Arguments

n

Numeric value, 1 through 4.

It identifies the DNS. Up to four servers may be specified.

ip4addr

The IPv4 address of the domain name server to be used. If omitted, no DNS lookup is performed.

Example

```
 SET DNS1=65.24.7.3
IPN254I DNS 1 address is 65.24.7.3, Timeout is 1200
```

Notes

The following notes apply to this command:

- When you need to convert a symbolic name to an IP address, TCP/IP for VSE first consults the local names table. If this search fails, a call is made to the specified DNS. If the name is still not resolved and a domain string is set (SET DEFAULT_DOMAIN), then a period and the domain string are appended to the symbolic name, and the entire search process (cache checking through DNS lookup) is repeated using this composite name.
- TCP/IP for VSE supports both GetHostByName and GetHostByAddr calls to the DNS.
- If more than one DNS is specified, TCP/IP for VSE tries each DNS in turn, beginning with DNS1, until a response is received or all servers have been polled. The first server to respond determines whether the request is successful, meaning that if DNS1 responds negatively ("No such domain name exists," for example), DNS2 is not queried.
- See the SET DNST*n* command for information about determining whether domain name servers have responded.

Related Commands

DEFINE NAME

Associates a TCP/IP name with an IP address or a script file. The resulting names table is searched when an application issues a call to convert a symbolic name to an IP address.

QUERY NAMES

Displays the contents of the symbolic names table created by DEFINE NAME.

QUERY OPTIONS

Displays the current values of modifiable parameters, including the IP addresses of the servers specified by SET DNS*n*.

SET DEFAULT_DOMAIN

Establishes a domain name to be automatically appended to unqualified names.

SET DNST*n*

Controls the time-out value to be used for a corresponding DNS (specified by SET DNS*n*).

SET DNST*n*

Each SET DNST*n* command specifies a timeout value for a corresponding SET DNS*n* command. For example, the SET DNST1 command specifies the timeout value for the domain name server (DNS) specified by SET DNS1.

Syntax

SET DNST*n*=*sec300*

Arguments

n

Numeric value, 1 through 4, that identifies the DNS entry to modify. Up to four servers can be specified.

sec300

Numeric value, 300 through 65535, the number of 300th-second units.

Specifies the timeout value to be used for the corresponding DNS. The default is 1200 (4 seconds).

Example

```
 set dnst1=1500
IPN254I DNS 1 address is 65.24.7.3, Timeout is 1500
```

Related Commands

QUERY OPTIONS

Displays the current values of modifiable parameters, including the address of each DNS defined using SET DNS*n*.

SET DNS*n*

Specifies the IPv4 address of a domain name server.

SET FIXED_RETRANSMIT

The FIXED_RETRANSMIT command sets the default value for the FIXRETRAN= parameter on the DEFINE ROUTE command.

Syntax

SET FIXED_RETRansmit={ON|OFF}

Argument

ON

The default for the DEFINE ROUTE FIXRETRAN= parameter is set to YES. This means that if FIXRETRAN= is not specified on the route statement, the values specified for DRETRAN= and RPAUSE= on DEFINE ROUTE will remain constant for the duration of the connection.

OFF

The default for the DEFINE ROUTE FIXRETRAN= parameter is set to NO. This means that if FIXRETRAN= is not specified on the route statement, the values for DRETRAN= and RPAUSE= start out as specified but will be dynamically adjusted as the network response is analyzed. OFF is the system default.

Example

```
 set fixed_retransmit=on
IPN251I Fixed retransmission rate set to on
```

Notes

The following notes apply to this command:

- The DEFINE ROUTE FIXRETRAN= parameter takes precedence over SET FIXED_RETRANSMIT. Using FIXRETRAN= in each route statement is the recommended way of updating this setting.
- If the FIXRETRAN= parameter is not set in a DEFINE ROUTE statement, the default setting is displayed in the DEFINE ROUTE output as follows (the Fixed: field).

```
IPN876I     SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
```

- Using SET FIXED_RETRANSMIT does not affect existing route definitions.
- See DEFINE ROUTE for more information on the DRETRAN= and RPAUSE= parameters.

Related Commands

DEFINE ROUTE

Adds an entry to the TCP/IP routing table.

MODIFY ROUTE
Changes values on an existing entry in the Route Table.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

SET IPADDR

The SET IPADDR command sets the default network address of TCP/IP FOR VSE.

Syntax

SET IPaddr=*ip4addr*

Arguments

IPaddr=
The default network address of the TCP/IP FOR VSE partition.

Example

```
 SET IPADDR=192.168.1.161
IPN268I IPADDRESS now set to 192.168.1.161
IPN188I IP Address 192.168.1.161 = Net: 192.168.1.0 Subnet: -- Host: 0.0.0.161
```

Notes

The following notes apply to this command:

- The SET IPADDR= command must be specified in the initialization library member and, once specified, should not be changed without restarting TCP/IP for VSE.
- The IPN188I message in response to the SET IPADDR command shows the network and subnetwork numbers after the subnet mask is applied. The mask is set using the SET MASK command.
- This IP address should be "overridden" with the IPADDR= parameter on the DEFINE LINK and DEFINE adapter commands.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE ALTIP
Causes the stack to monitor and respond to ARP requests for additional home addresses.

DEFINE LINK
Creates a link between TCP/IP and a network or to a directly connected stack.

DEFINE MASK
Creates a subnet mask for a particular network.

QUERY LINKS
Displays the status of network links.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY OPTIONS

Displays the current values of modifiable parameters.

SET MASK

Establishes a default subnet mask.

SET LINK_RETRY

The SET LINK_RETRY command specifies the default length of time that must elapse before TCP/IP FOR VSE attempts to reinitialize a link driver that fails to initialize.

Syntax

SETLINK_retry=*sec300*

Arguments

sec300

Numeric, 0 through 9999999, the number of 300th-second units.

The interval to wait between attempts to initialize a link driver. The default value is 18000 (60 seconds).

Example

```
 set link_retry=1800
IPN268I LINK_RETRY now set to 1800
```

Notes

You can also specify a value of LINK_RETRY= on the DEFINE LINK command. That value overrides the global default defined by SET LINK_RETRY.

Related Commands

DEFINE LINK

Creates a link between TCP/IP and a network or to a directly connected stack.

QUERY OPTIONS

Displays the current values of modifiable parameters.

SET MASK

The SET MASK command identifies what portion of the host number in a network address is used to identify a subnetwork.

Syntax

SET MASK=*ip4addr*

Arguments

ip4addr

Specifies the value of the mask to be applied to the network address to obtain the subnetwork number. This value is coded in dotted decimal notation in the same manner as a TCP/IP network address, that is, *n.n.n.n*, where each instance of *n* is the decimal representation of one byte.

Example

```
 set mask=255.255.255.0
IPN268I MASK now set to 255.255.255.0
IPN188I IP Address 192.168.1.161 = Net: 192.168.1.0 Subnet: -- Host: 0.0.0.161
```

Notes

The following notes apply to this command:

- IP addresses consist of a network number and a host number. For added flexibility, a mask may be applied to the host number to yield a subnetwork number. This subnet number can then be used when coding DEFINE ROUTE statements and when generic addresses are desired.
- SET MASK establishes the mask that is to be applied to any network that is not explicitly defined with a DEFINE MASK command.
- The IPN188I message in response to the SET IPADDR command shows the network and subnetwork numbers after the subnet mask is applied. The IP address is set using the SET IPADDR command.
- See the *TCP/IP FOR VSE Installation Guide* for more information on defining networks and subnetworks.

Related Commands

CONNECT_SEQUENCE

Controls whether connection requests are allocated by IP address pattern checking.

DEFINE FTPD

Creates a File Transfer Protocol daemon.

DEFINE MASK

Creates a subnet mask for a particular network.

DEFINE ROUTE
Adds an entry to the TCP/IP FOR VSE routing table.

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

QUERY MASKS
Shows all defined subnetwork masks by network number.

QUERY OPTIONS
Displays the current values of modifiable parameters.

SET IPADDR
Establishes the default home address for the stack.

SET TELNET_TRANSLATE
Sets the name of the translate table that will be used with Telnet (not TN3270) connections.

SET MAXIMUM_MESSAGES

The SET MAXIMUM_MESSAGES command sets the maximum number of queued (not printed) system messages before the stack ignores the rest and rejects additional incoming messages to SYSLOG. If the messages are only to be routed to SYSLST, then those messages will still be sent to the printer. If the messages are to go to SYSLST as well as SYSLOG, but there is a queuing, then neither will be output.

Normally the limit is reached only if the messages are going to SYSLOG and the operator scrolls backward on the console and leaves it in that mode. This would effectively block message output.

Syntax

SET MAXIMUM_MESSAGES=*num*

Arguments

num

Numeric; 0 through 9999999. The default is 1000.

Example

```
 SET MAXIMUM_MESSAGES=500
IPN268I MAXIMUM_MESSAGES now set to 500
```

Related Commands

MESSAGE

Controls message suppression.

MODIFY CONSOLE

Controls how messages are displayed on the console.

QUERY LOGS

Displays available consoles and logs, along with their properties

SET DIAGNOSE

Enables console display of messages resulting from the DIAGNOSE command.

SET MAX_EMAIL_EVENTS

This command controls how many simultaneous EMAIL events will be handled by Automation processing.

Syntax

SET MAX_Email_events=*num*

Arguments

num

Numeric, 1 through 9999999. The default is 1.

Example

```
 set max_email_events=1
IPN268I MAX_EMAIL now set to 1
```

Notes

The following notes apply to this command:

- Email transmission generally occurs quickly.
- An SMTP server may limit the number of simultaneous connections with a host.
- If an email transmission requires auxiliary services, such as PDF conversion, you must consider how much CPU loading is appropriate for the TCP/IP FOR VSE partition.

Related Commands

DEFINE EVENT

Monitors a POWER class for automatic report distribution.

QUERY EMAIL

Displays EMAIL client settings.

QUERY EVENTS

Displays the status of automation processing.

SET MAX_FTP_EVENTS

Establishes the maximum number of simultaneous FTP events for Automation processing.

SET MAX_LPR_EVENTS

Establishes the maximum number of simultaneous LPR events for Automation processing.

SINGLEDEST

Determines how automation processing handles multiple reports queued for the same destination (host).

SET MAX_FTP_EVENTS

This command controls how many simultaneous FTP events will be handled by Automation processing.

Syntax

SET MAX_Ftp_events=*num*

Arguments

num
Numeric, 1 through 9999999. The default is 1.

Example

```
 set max_ftp_events=1
IPN268I MAX_FTP now set to 1
```

Notes

The following notes apply to this command:

- FTP transmissions generally occur quickly.
- Each FTP thread consumes an available FTP session. Be sure that sufficient free sessions are available.
- If FTP transmission requires auxiliary services, such as PDF conversion or encryption, consider how much CPU loading is appropriate for the TCP/IP FOR VSE partition.

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

QUERY EVENTS
Displays the status of automation processing.

QUERY FTPDS
Displays the status of the File Transfer Protocol daemons.

SET MAX_EMAIL_EVENTS
Establishes the maximum number of simultaneous EMAIL events for Automation processing.

SET MAX_LPR_EVENTS
Establishes the maximum number of simultaneous LPR events for Automation processing.

SINGLEDEST
Determines how automation processing handles multiple reports queued for the same destination (host).

SET MAX_LPR_EVENTS

This command controls how many simultaneous LPR events will be handled by Automation processing.

Syntax

SET MAX_Lpr_events=*num*

Arguments

num
Numeric, 1 through 9999999. The default is 5.

Example

```
 set max_lpr_events=3
IPN268I MAX_LPR now set to 3
```

Notes

The following notes apply to this command:

- LPR transmission to a true LPD generally occurs quickly.
- An LPD server limits the number of simultaneous connections with a host.
- If the destination LPD is actually a printer, a single LPR transmission may require minutes or hours. During this time, resource consumption (CPU, I/O) is generally low.

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

QUERY EVENTS
Displays the status of automation processing.

SET MAX_EMAIL_EVENTS
Establishes the maximum number of simultaneous EMAIL events for Automation processing.

SET MAX_FTP_EVENTS
Establishes the maximum number of simultaneous FTP events for Automation processing.

SINGLEDEST
Determines how automation processing handles multiple reports queued for the same destination (host).

SET MAX_SEGMENT

The SET MAX_SEGMENT command limits the size of incoming TCP data segments. This command sets the default value for the MSS= parameter in DEFINE ROUTE statements.

SET MAX_SEGMENT and SET MAXSEGMENT are synonyms.

Syntax

SET MAX_Segment={32684|*num*}

Arguments

num

Numeric; 576 through 65495.

The maximum size of TCP data segments to be negotiated by TCP/IP FOR VSE. The default is 32684. The recommended value is 65495.

Example

```
 set max_segment=65495
IPN268I MAXIMUM_SEGMENT now set to 65495
```

Notes

The following notes apply to this command:

- When a TCP connection is negotiated, each side tells the other the size of the maximum data segment that can be sent. Normally, TCP/IP FOR VSE requests the maximum size segment that can be achieved without datagram fragmentation. This can be computed as the MTU size minus 40.
- If you are experiencing problems with the inbound data stream, reducing the Maximum Segment Size may help in diagnosing the problem. Note that this has no effect on the size of outbound datagrams.
- The DEFINE ROUTE MSS= parameter takes precedence over SET MAX_SEGMENT. Setting MSS= on each DEFINE ROUTE statement is the recommended way of updating the value.
- The MSS= value that is used for a DEFINE ROUTE statement is displayed in the DEFINE ROUTE output as follows (the Max Seg field).

  ```
  IPN875I     MTU: 0 Max Seg: 32684 Pulse: 60s
  ```

- Using SET MAX_SEGMENT does not affect existing route definitions.

Related Commands

DEFINE ROUTE

Adds an entry to the TCP/IP routing table.

QUERY OPTIONS

Displays the current values of modifiable parameters.

SET PAGE_COUNT

The SET PAGE_COUNT command determines how many POWER-generated separator pages are used with each file read from POWER if JSEP is enabled.

Syntax

SET PAGE_COUnt=*num*

Arguments

num

Numeric, 0 through 255.

The number of POWER-generated separator pages that will be requested. The default is 0.

Example

```
 set page_count=3
IPN268I PAGE_COUNT now set to 65535
```

Notes

You must also use the SEPARATOR_PAGES command to specify what processes are eligible to receive them.

Related Commands

QUERY OPTIONS

Displays the current values of modifiable parameters.

SEPARATOR_PAGES

Controls the generation of POWER separator pages.

SET PASSWORD

The SET PASSWORD command establishes a password that is required when sending messages to the TCP/IP FOR VSE partition.

Syntax

SET PASSword=*string*

Arguments

string

1- to 8-character string, case insensitive.

This password is required when you issue the MSG command to the TCP/IP FOR VSE partition.

Example

```
 set password=wavv
msg f4
IPN223A Please enter password
wavv
IPN300A Enter TCP/IP Command
```

Notes

The following notes apply to this command:

- There is no response from this command, and you cannot query the current password value.
- If you place this command in your initialization deck, you can prepend the command with a plus sign (+) to suppress echo printing of the password.
- To remove the password, issue the SET PASSWORD= command with no parameter.
- The ability to enter commands using batch interface IPNETCMD is not affected by passwords. If security is an issue, ensure that IPNETCMD is not available to users.
- See the “Security” chapter in the *TCP/IP FOR VSE Installation Guide* for more information about using this facility.

Related Commands

CONSOLE_HOLD

Maintains a console command prompt.

SET POWERPASSWORD

The SET POWERPASSWORD command establishes a password that will be used for TCP/IP FOR VSE access to the VSE/POWER queues.

Syntax

SET POWERPassword=*string8*

Arguments

string8

1- to 8-character string, case insensitive.

This password is used for POWER access. Although there is no default value for this parameter, POWER access generally does not require a password.

Example

```
 set powerpassword=fishbulb
IPN268I POWERPASSWORD now set to ++Suppressed++
```

Related Commands

SET POWERUSERID

Establishes the user ID for POWER access.

SET POWERUSERID

The SET POWERUSERID command establishes the userid that TCP/IP will use when accessing the VSE/POWER queues.

Syntax

SET POWERUserid=*string8*

Arguments

string8

1- to 8-character string, case insensitive.

This is the user ID to use for POWER access. The default user ID is "SYSTCPIP".

Example

```
 set poweruserid=SYSTCPIP
IPN268I POWERUSERID now set to SYSTCPIP
```

Related Commands

SET POWERPASSWORD

Establishes the password for POWER access.

SET PULSE_TIME

The SET PULSE_TIME command sets the interval after which TCP/IP FOR VSE tests inactive connections to see if they should be terminated.

This command sets the default for the PULSE= parameter on the DEFINE ROUTE command.

Syntax

SET PULse_time={18000|*sec300*}

Arguments

sec300

Numeric, 0 through 9999999, the number of 300th-second units.

Specifies the time interval that TCP/IP FOR VSE allows a TCP connection to remain idle before it tests for the continued presence of a remote host. The default is 18000 (one minute). Setting a value of 0 disables the pulse mechanism.

Example

```
 set pulse_time=30s
IPN268I PULSE_TIME now set to 9000 300th sec
```

Notes

The following notes apply to this command:

- TCP/IP FOR VSE tests an idle TCP connection by retransmitting an empty packet with the same sequence number as the last acknowledged byte. TCP protocols require the remote host to re-acknowledge receipt of the byte and to discard it. If the byte is not acknowledged, standard retransmission routines gain control and, if an acknowledgement is still not forthcoming, the connection is dropped and the application owning the connection is notified.
- If a PC running a TN3270 session is booted without terminating the session, the associated TN3270 daemon remains in session because it has no way of knowing that the remote host is gone. When the pulse process activates, the session closes.
- A low PULSE_TIME value results in unnecessary processing. A high PULSE_TIME value allows dead connections to persist.
- The pulse mechanism is also useful in restoring connections that have gone to sleep.
- The DEFINE ROUTE PULSE= parameter takes precedence over SET PULSE_TIME. Setting PULSE= on each DEFINE ROUTE statement is the recommended way of updating the value.

- The PULSE parameter value that is used in a route definition is displayed in the DEFINE ROUTE output as follows (the Pulse: field). The value is displayed in seconds.

```
IPN875I     MTU: 0 Max Seg: 32684 Pulse: 60s
```

  If the PULSE= parameter is not set in the DEFINE ROUTE statement, then the value displayed is the default.

- Using SET PULSE_TIME does not affect existing route definitions.

Related Commands

DEFINE ROUTE
Adds an entry to the TCP/IP FOR VSE routing table.

QUERY OPTIONS
Displays the current values of modifiable parameters.

SET RETRANSMIT

The SET RETRANSMIT command controls the length of the default interval before TCP/IP FOR VSE enters retransmit mode for unacknowledged data packets.

This command sets the default for both the CRETRANS= and the DRETRANS= parameters in DEFINE ROUTE statements.

Syntax

SET RETransmit={300|*sec300*}

Argument

sec300

Numeric, 0 through 9999999, the number of 300th-second units.

Specifies the time interval to wait for a datagram to be acknowledged. The default is 300, which equals 1000 ms or 1 sec.

Example

```
set retransmit=300
IPN268I RETRANSMIT now set to 300
```

Exposition

The IP transmission protocol requires that each data packet be delivered in a timely fashion or be thrown away. There are no exceptions, and data cannot be queued or delayed in any manner. There is no mechanism that permits IP to inform the sender that a data packet has been lost or discarded.

To allow for IP's habit of discarding data packets, TCP requires an acknowledgment for each data packet sent (from the TCP receiving the packet). If an acknowledgment is not forthcoming, TCP must retransmit the data packet.

Under normal circumstances, TCP/IP FOR VSE notes (on each connection) the time required to receive an acknowledgment. It uses this value to determine when an acknowledgment is overdue and a retransmission is required. The value provided by SET RETRANSMIT is used as the starting point for each new connection.

The proper value for the default retransmission timer is a function of the speed of your network and the capabilities of the hosts on your network. You can use the DIAGNOSE command with the PERFORM option to determine whether you are having retransmission problems.

Notes

The following notes apply to this command:

- The concept of retransmission time has been broken into two sets of two values. The two values are the threshold value until a retransmission is required and the time between retransmissions. Separate sets of values pertain to connection requests and established

connections. All of these values must be set separately using the DEFINE ROUTE command.

- If the CRETRANS= and DRETRANS= parameters are not set on DEFINE ROUTE, the SET RETRANSMIT value is displayed in the DEFINE ROUTE output as follows (the SYN Retran and Data Retran fields). The values are shown in milliseconds, which is the units for these DEFINE ROUTE parameters.

```
IPN876I     SYN Retran: 1000ms Data Retran: 1000ms Fixed: No
```

- The DEFINE ROUTE CRETRANS= and DRETRANS= parameters take precedence over SET RETRANSMIT. Setting the CRETRANS= and DRETRANS= parameters in each DEFINE ROUTE statement is the recommended way of updating these values.

- Using SET RETRANSMIT does not affect existing route definitions.

Related Commands

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

MODIFY ROUTE
Changes values on an existing entry in the Route Table.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY ROUTES
Displays the content of the network routing table or the route taken to reach a specific address.

SET TELNET_TRANSLATE

The SET TELNET_TRANSLATE command specifies the name of a translate table to be used for line-mode Telnet sessions initiated by the CICS line-mode Telnet client, the batch client, programmable Telnet clients, and FTP control connections.

Syntax

SET TELNET_Translate=*name16*

Arguments

name16
The name of the translate table to be used by the Telnet client.

Example

```
 set telnet_translate=us_eng_03
IPN804I Telnet translation will use US_ENG_03
```

Notes

The following notes apply to this command:

- The name of the translate table is not checked for validity until it is needed. If the name is not valid, the system default translate table is used.
- See the *TCP/IP FOR VSE Installation Guide* for more information about defining translate tables.

Related Commands

DEFINE TRANSLATION
Loads and controls ASCII/EBCDIC translation tables.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY TRANSLATES
Displays a list of available translate tables.

SET TELNETD_BUFFERS

The SET TELNETD_BUFFERS command controls the number of buffers in the Telnet daemon buffer pool.

Syntax

SET TELNETD_BUFFers=*num*

Arguments

num

Numeric, 0 through 9999999.

The number of buffers in the pool. The default is 20. Code 0 if no Telnet daemons (with POOL=YES) are to be used. Values higher than the total number of daemons waste storage.

Example

```
 set telnetd_buffers=10
IPN268I Telnetd buffers now set to 10
```

Notes

The following notes apply to this command:

- Telnet daemons defined with the POOL=YES operand share their buffers. Each daemon obtains a buffer only when it is actually transferring data.
- An increase in the number of buffers is effective immediately.
- A decrease in the number of buffers is effective immediately if the buffers to be released are not in use.

Related Commands

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY STATISTICS
Displays a summary of stack-related information.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

SET TELNETD_BUFSIZE

The SET TELNETD_BUFSIZE command controls the size of buffers in the Telnet daemon buffer pool.

Syntax

SET TELNETD_BUFSize=*num*

Arguments

num

Numeric, 8192 through 9999999.

The default size (8192) of Telnet buffers should be sufficient for all normal 3270 operations. In rare cases where screen sizes are maximized and screen complexity is high, it may be necessary to increase the buffer size.

Example

```
 set telnetd_bufsize=8192
IPN268I Telnetd buffers now set to 0
IPN268I TELNETD_BUFSIZE now set to 8192
IPN268I Telnetd buffers now set to 10
```

Notes

The following notes apply to this command:

- Daemons defined with POOL=NO obtain their buffers at startup. Before a change in buffer size will take effect for a specific daemon, it must be recycled.
- When the buffer size is changed, all existing buffers in the buffer pool are released and then reallocated in the correct size. If a buffer is in use, it will be released when it is returned to the pool.

Related Commands

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon.

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY STATISTICS
Displays a summary of stack-related information.

QUERY TELNETDS
Displays TN3270 and TN3270E daemons.

SET TELNETD_BUFFERS
Determines the size of the buffer pool shared by TN3270 daemons.

SET WINDOW

The SET WINDOW command controls the default amount of data that can be received from a remote host before it must wait for an acknowledgment.

This command sets the default for the WINDOW= parameter on the DEFINE ROUTE command.

Syntax

SET WINdow={65535|*num*}

Arguments

num

Numeric, 1500 through 65535.

The number of bytes that can be received before an acknowledgment must be awaited. The default is 65535.

Example

```
 set window=65535
IPN268I WINDOW now set to 65535
```

Exposition

The "window" is the number of bytes that the remote host is authorized to send. Because each byte in a TCP connection is sequentially numbered, the window is actually the sequence number of the byte following the last position in the window. One of the rules is that this end point can never move backward (shrinking window).

Under normal circumstances, as data is received, it is acknowledged and the window is further extended. If the data arrives faster than it can be accommodated, it still must be acknowledged in a timely fashion, but the end point of the window is not advanced. Eventually, the end point is reached and the window is "closed."

Once the window is closed, no more data can be sent until it has been re-opened. One of the rules is that when a window is re-opened, it must be fully re-opened to its maximum size.

Communication links have become faster, and available VSE storage is plentiful. To increase overall transmission speed, TCP/IP FOR VSE buffers well beyond the window size, keeping the "available window" at its maximum value with every acknowledgement.

When selecting a window size, you must also consider both the MTU and MSS values.

Notes

The following notes apply to this command:

- The window value affects only inbound traffic.
- The window size must be at least MTU-40.
- See the *TCP/IP FOR VSE Installation Guide* for more information on the TCP/IP window.
- The DEFINE ROUTE WINDOW= parameter takes precedence over SET WINDOW. Setting WINDOW= in each DEFINE ROUTE statement is the recommended way of updating the value.
- If the WINDOW= parameter is not set in DEFINE ROUTE statements, the SET WINDOW value is displayed in the DEFINE ROUTE output as follows (the RWin: field). The value is the number of bytes.

```
IPN884I     RWin: 65535
```

- Using SET WINDOW does not affect existing route definitions.

Related Commands

DEFINE ROUTE
Adds an entry to the TCP/IP routing table.

QUERY OPTIONS
Displays the current values of modifiable parameters.

SHUTDOWN

The SHUTDOWN command directs TCP/IP FOR VSE to terminate all operations.

Syntax

SHUTdown [Immediate]

Arguments

Immediate
When specified, the shutdown process bypasses most of the normal shutdown processes. Only items that may prevent a normal restart are performed.

Example 1

```
 shutdown
IPN205A Respond "YES" for normal TCP/IP shutdown
Reply to IPN205A is YES
IPN146I TCP/IP Beginning Shutdown
IPN597I Shutdown Stage: 1: Connection Terminations
TCP913I Event Deleted: AUTOLPR1
TCP913I Event Deleted: AUTOFTP2
TCP913I Event Deleted: AUTOFTP1
TCP901I Shutdown Client Automation Daemon
LPD903I Daemon Shutdown LPD
FTP912I FTP01 not accepting connections on port 21
FTP908I Daemon Shutdown FTP Id: FTP01 Port: 21
IPN597I Shutdown Stage: 2: Daemon Terminations
IPN597I Shutdown Stage: 3: Link Driver Terminations
IPT101I Link Driver Processor LCS stopping
IPI101I IP Processor Collector stopping
IPN597I Shutdown Stage: 4: Network termination (max. 10 seconds)
IPN597I Shutdown Stage: 5: API termination (max. 10 seconds)
IPN597I Shutdown Stage: 6: System Task Terminations
IPI101I IP Processor Prep 2 stopping
IPI101I IP Processor Prep 1 stopping
IPI101I IP Processor Raw 1 stopping
IPI101I IP Processor Raw 2 stopping
IPI101I IP Processor Input IP stopping
IPI101I IP Processor Output IP stopping
IPT101I TCP Processor Input stopping
IPT101I TCP Processor Output stopping
IPT101I UDP Processor Input stopping
IPT101I UDP Processor Output stopping
IPT101I ICMP Processor Input stopping
IPT101I ICMP Processor Output stopping
IPT101I IGMP Processor Input stopping
IPT101I IGMP Processor Output stopping
IPT101I Internet Link Level (ILL) Processor Input 802.2 stopping
IPT101I Internet Link Level (ILL) Processor Output 802.2 stopping
IPN597I Shutdown Stage: 7: Residual Task Termination
```

(continued)

```
IPN597I Shutdown Stage: 8: Halt Device I/O
IPN597I Shutdown Stage: 9: Security Exit Shutdown
IPN597I Shutdown Stage: 10: Terminate File I/O
IPN597I Shutdown Stage: 11: Subtask Terminations
IPN597I Shutdown Stage: 12: Operating System Interface Removal
IPN597I Shutdown Stage: 13: Printing Statistics
IPN597I Shutdown Stage: 14: Terminating Console Logging
```

Example 2

```
 shutdown immediate
IPN205A Respond "YES" for IMMEDIATE TCP/IP shutdown
Reply to IPN205A is YES
IPN146I TCP/IP Beginning Shutdown
IPN597I Shutdown Stage: 3: Link Driver Terminations
IPT101I Link Driver Processor LCS stopping
IPN597I Shutdown Stage: 8: Halt Device I/O
IPN597I Shutdown Stage: 11: Subtask Terminations
IPN597I Shutdown Stage: 12: Operating System Interface Removal
IPN597I Shutdown Stage: 14: Terminating Console Logging
```

Notes

The following notes apply to this command:

- The operator is always prompted to confirm an Immediate-mode shutdown.
- During shutdown processing, the stack notifies all applications and daemons. It then waits for them to close all connection and terminate. The stack continues to wait until no additional progress is made. This step is skipped when “Immediate” is specified.
- Following application notification, all remaining connections are severed with a RESET. This step is skipped when “Immediate” is specified.
- SVA-based storage that is not intended to survive the stack’s shutdown is released.

Related Commands

CONNECT_STATS
Controls the printing of connection statistics during shutdown.

DOWNCHECK
Controls the safety prompt for the SHUTDOWN command.

QUIESCE
Prevents new connections while permitting existing connections to continue.

SINGLEDEST

The SINGLEDEST command determines how automation processing handles multiple reports queued for the same destination (host).

Syntax

SINGledest {ON|OFf}

Arguments

ON

This indicates that reports with the same script name will be treated as though they are going to the same printer. This does not cause all reports to be delivered single-threaded, but it directs the delivery logic to not send a report to a destination if there is already a report that uses the same script name in transit. This is the default.

Note the following:

- To ensure that reports are issued single threaded across all destinations, use SINGLE=YES on the DEFINE EVENT command.
- The automation client assumes each script that has a different name has a different destination. The client does not know which destination may be set within a script.
- If ORDER=YES on the DEFINE EVENT command, then all reports will arrive in order, based on the POWER queue number. Otherwise, they will be delivered in the order that they are encountered and complete. For some clients, especially Email, order is unnecessary. For others, such as LPD, it may or may not be important, depending on the site's needs.
- If ORDER=YES and any one report is hung (the server has stopped responding, for example), report processing stops because reports must arrive in order.

OFf

Automation processing does not assume that reports are going to the same destination, even if the script names are the same. This will cause asynchronous delivery of multiple reports, so long as "SINGLE=NO" is set in the DEFINE EVENT. Please note that the maximum number of asynchronous report deliveries is set by SET MAX_*xxx*_EVENTS commands. Do not attempt to deliver multiple reports to the same destination unless you know that the destination(s) can accept simultaneous transmissions.

Example

```
 singledest on
IPN249I Value for Single Destination set to on
```

Notes

The following notes apply to this command:

- EMAIL events are not under the control of the SINGLEDEST setting.
- If any destination LPD is a physical printer, always use SINGLEDEST ON because a printer cannot accept more than one report at a time.
- Some servers are not true LPD servers, but simulate them and may or may not be able to accept multiple reports that will then be forwarded to a true LPD server, which improves performance. Determine if this is the case in your environment.

Related Commands

DEFINE EVENT
Monitors a POWER class for automatic report distribution.

QUERY EVENTS
Displays the status of automation processing.

SET MAX_EMAIL_EVENTS
Establishes the maximum number of simultaneous Email events for automation processing.

SET MAX_FTP_EVENTS
Establishes the maximum number of simultaneous FTP events for automation processing.

SET MAX_LPR_EVENTS
Establishes the maximum number of simultaneous LPR events for automation processing.

SPINCHECK

The SPINCHECK command disables the TCP/IP FOR VSE loop-detection mechanism. You should use this command only if you must allow extremely CPU-intensive processes to complete.

Syntax

SPINcheck {ON|OFF}

Arguments

ON

Directs TCP/IP FOR VSE to monitor the CPU usage of internal tasks and pseudo tasks. If a loop condition is detected, diagnostic information is produced and TCP/IP FOR VSE terminates. This is the default.

OFF

The TCP/IP FOR VSE loop-detection routines are disabled.

Example

```
 spincheck off
IPN268I SPINCHECK now set to OFF
```

Related Commands

QUERY OPTIONS

Displays the current values of modifiable parameters.

START

The START command restarts the TCP/IP FOR VSE engine or a communications link.

Syntax

START [LINKid=*id*]

Arguments

(Null argument)
The TCP/IP FOR VSE dispatcher resumes execution.

id
The communications link with this ID is started.

Example 1

```
 start
IPN499I TCP/IP processing started
```

Example 2

```
 start linkid=xmem_link
IPN477I Task 0028 has resumed. IPNLRAW D:1 R:0
```

Notes

The following notes apply to this command:

- After the START command executes, the TCP/IP engine resumes at the point where the preceding STOP command was issued.
- Timeouts and disconnections may occur after executing START. This is normal and results from delays introduced during the stopped period.
- The START command is used to start communications links that were defined with the STOPPED parameter.

Related Commands

DEFINE LINK
Creates a link between TCP/IP and a network or to a directly connected stack.

STOP
Stops the TCP/IP dispatching engine.

STEALTH

Controls how the stack handles connection requests that cannot be matched with a listen connection.

Syntax

STEALTH {ON|OFF}

Arguments

ON

Refused connection requests are quietly dropped without replying with a RESET.

OFF

Connection requests that cannot be matched with a listen connection are forcefully rejected with a RESET. This is the default behavior.

Example

```
 stealth on
IPN268I Stealth mode now set to ON
```

Notes

Stealth mode is useful in deterring hacking attempts. The remote host is not presented with any evidence that anything exists at the address being tried. When a port scan is attempted, each port being tested requires the maximum amount of time to "fail."

Related Commands

QUERY PORTQUEUE

Displays statistics associated with queued connection requests.

QUERY OPTIONS

Displays the current values of modifiable parameters.

STOP

The STOP command causes the TCP/IP engine to (1) temporarily stop dispatching pseudo tasks or (2) suspend attempts to activate a specific communications link.

Syntax

STOP [LINKid=*id*]

Arguments

(Null argument)
If no argument is specified, the entire TCP/IP dispatcher is placed in the stopped state until a START command is issued.

id
The link driver with this ID is placed into a stopped state. The remainder of the TPC/IP product continues to function.

Example 1

```
 stop
IPN500W All TCP/IP processing stopped
```

Example 2

```
 stop linkid=link3172
IPN478I Task 0020 suspended in phase IPNLOSA2. Disp: 223
```

Notes

The following notes apply to this command:

- During the stopped state, the socket collection mechanism continues to function. This means that system and partition GETVIS accumulates until a START command is issued and requests can be processed.
- The STOP command is useful when producing a static dump with the DUMP command.
- If you need to dynamically add a 3172 link, issuing the STOP command permits you to enter the DEFINE LINK command and the related DEFINE ADAPTER commands. If no STOP is issued, the system dynamically configures the link before the DEFINE ADAPTER commands can be entered. See also DEFINE LINK for a caution statement about defining a link that was deleted.
- Use STOP LINKID= only with CTCA and cross-partition (IPNET) links. Stopping a link that is driving an LCS device causes excessive error recovery, and a link restart is likely to fail.

Related Commands

DEFINE ADAPTER
Creates an adapter definition within the scope of a DEFINE LINK.

DEFINE LINK
Creates a link between TCP/IP and a network or to a directly connected stack.

QUERY LINKS
Displays the status of network links.

START
Starts the TCP/IP dispatching engine or a “stopped” network link.

SUSPEND

The SUSPEND command suspends a TCP/IP FOR VSE pseudo task from further execution.

Syntax

SUSPend *taskid*

Arguments

taskid
Is the identifying hexadecimal task number of the pseudo task to be suspended.

Example

```
 suspend 000d
IPN478I Task 000D suspended in phase IPNIGARB. Disp: 3,855
```

Notes

The following notes apply to this command:

- The suspended pseudo task is not changed in any way except that it is not eligible for dispatch.
- You can use the QUERY TASKS command to obtain the task numbers of TCP/IP for VSE internal pseudo tasks.
- The task number is prepended to each console message displayed by TCP/IP for VSE tasks.
- Do not suspend and resume TCP/IP for VSE pseudo tasks unless you understand the effects of these actions.

Related Commands

QUERY TASKS
Displays a list of pseudo tasks.

RESUME
Resumes processing of a suspended pseudo task.

TRACERT

The TRACERT command identifies each hop between the stack and the remote host.

Syntax

TRACert *host*

Arguments

host
Specifies the IP address of the target host. This may be an actual numeric address or a symbolic name that can be resolved to an IP address.

Example

See the *TCP/IP FOR VSE Messages* manual for an explanation of each message in this example.

```
 tracert csi-international.com
TCP915I Tracing route to 012.182.034.227 (csi-international.com)
TCP910I Hop: 010.089.128.001 at milliseconds: 00008.
TCP910I
TCP910I Hop: 024.095.086.193 at milliseconds: 00012.
TCP910I GIG3-10.WOTNOH1-SWT401.COLUMBUS.RR.COM
TCP910I Hop: 065.025.129.155 at milliseconds: 00013.
TCP910I TGE1-2.CNVLOH1-SWT401.COLUMBUS.RR.COM
TCP910I Hop: 065.025.129.157 at milliseconds: 00011.
TCP910I TGE1-2.GRVWOH1-SWT401.COLUMBUS.RR.COM
TCP910I Hop: 065.025.129.159 at milliseconds: 00012.
TCP910I TGE2-1.CLMBOH1-RTR2.COLUMBUS.RR.COM
TCP910I Hop: 065.025.137.245 at milliseconds: 00013.
TCP910I TGE1-0-0.CLBOH1-RTR0.MWRTN.RR.COM
TCP910I Hop: 066.109.006.068 at milliseconds: 00025.
TCP910I AE-4-0.CR0.CHI30.TBONE.RR.COM
TCP910I Hop: 066.109.006.155 at milliseconds: 00022.
TCP910I AE-1-0.PR0.CHI10.TBONE.RR.COM
TCP910I Hop: 004.059.028.109 at milliseconds: 00021.
TCP910I XE-10-2-0.EDGE2.CHICAGO2.LEVEL3.NET
TCP910I Hop: 004.069.138.190 at milliseconds: 00023.
TCP910I VLAN52.EBR2.CHICAGO2.LEVEL3.NET
TCP910I Hop: 004.069.140.193 at milliseconds: 00035.
TCP910I AE-5-5.EBR2.CHICAGO1.LEVEL3.NET
TCP910I Hop: 004.068.101.168 at milliseconds: 00169.
TCP910I AE-24-56.CAR4.CHICAGO1.LEVEL3.NET
TCP910I Hop: 192.205.033.185 at milliseconds: 00024.
TCP910I
TCP910I Hop: 012.122.133.138 at milliseconds: 00034.
TCP910I CR1.CGCIL.IP.ATT.NET
TCP910I Hop: 012.123.007.025 at milliseconds: 00034.
TCP910I GAR4.CLBOH.IP.ATT.NET
TCP910I Hop: 012.090.239.238 at milliseconds: 00041.
TCP910I
```

(continued)

```
TCP910I Hop: 012.182.034.227 at milliseconds: 00042.
TCP910I CSI-INTERNATIONAL.COM
TCP910I TRACERT was successful.
```

Notes

The following notes apply to this command:

- ICMP Echo (ping) must be permitted on the network for TRACERT to function.
- Each hop displays the IP address and the domain name associated with the address. If the domain name is blank, either none exists or none is returned by the reverse DNS lookup.

Related Commands

DISCOVER
Determines the “best” MTU size to a remote host.

PING
Issues an ICMP Echo (PING) request.

TRAFFIC

The TRAFFIC command controls how TCP/IP FOR VSE handles IP and non-IP traffic. This can be helpful in diagnosing performance problems.

Syntax

TRAFFic {ON|OFF|FULL}

Arguments

ON
IP traffic is processed by the stack. This is the default and recommended setting.

OFF
No traffic is passed to the stack. All inbound datagrams are discarded.

FULL
All traffic is passed to the stack, including any non-IP traffic. This includes IPX and NetBEUI, for example.

Example

```
 traffic full
IPN268I TRAFFIC now set to FULL
```

Notes

The following notes apply to this command:

- Most network control units are capable of filtering out non-IP traffic. If your controller is equipped with this capability, enable it.
- Non-IP traffic increases CPU consumption.
- Non-IP traffic is not an issue with CTC, cross-partition (IPNET), OSA-Express, or CLAW links.
- You can issue the QUERY STATS command to determine if non-IP traffic is coming into your TCP/IP for VSE system.

Related Commands

QUERY OPTIONS
Displays the current values of modifiable parameters.

QUERY STATISTICS
Displays a summary of stack-related information.

TRUST

The TRUST command allows you to assign or remove an IP address's "trusted" status. A "trusted" IP address is never subject to having its access to VSE revoked based on suspicious activity.

Syntax

TRUSt {ADD|DELete} ,IPaddress=*ip4addr*

Arguments

ADD
The specified address is marked as "trusted."

DELete
The specified address is no longer marked as "trusted."

ip4addr
Is the IP address that is to have its "trusted" status updated.

Example

```
 trust add,ipaddress=192.168.1.66

 query trust
IPN253I << TCP/IP Trusted IP Addresses >>
IPN127I 192.168.1.66
IPN127I 192.168.1.1
12.182.34.168
```

Related Commands

ACCESS
Controls access to VSE by IP address

QUERY TRUSTED
Displays the currently trusted IP addresses

UPCASE

The UPCASE command specifies to display all TCP/IP FOR VSE console traffic in upper case.

Syntax

```
Upcase {ON|OFF}
```

Arguments

ON
All messages displayed on the console are shifted to upper case.

OFF
All messages displayed on the console are in mixed case. This is the default.

Example

```
 upcase on
IPN246I UPPER CASE MESSAGE TRANSLATION SET TO ON

 UPCASE OFF
IPN246I Upper Case message translation set to off
```

Notes

The following notes apply to this command:

- Some data displayed by TCP/IP for VSE console commands is case sensitive.
- The UPCASE command does not work for non-console output, such as the output from an FTP batch job. Some TCP/IP for VSE clients have options to force upper-case translation for messages.
- See the *TCP/IP FOR VSE User Guide* for information on how to enable upper-case message translation in the FTP clients.

Related Commands

DEFINE LOG
Creates a system log file.

MODIFY LOG
Changes characteristics of a system log file.

QUERY LOGS
Displays available consoles and logs, along with their properties.

QUERY OPTIONS
Displays the current values of modifiable parameters.

VERIFY_MEMORY

The VERIFY_MEMORY command disables or enables the storage monitor function. The storage monitor aggressively checks the memory for overlays or corruption caused by incorrectly executing tasks or processes.

Syntax

VERIFY_MEMORY {ON|OFF}

Arguments

ON

The TCP/IP FOR VSE multi-tasking engine tests every storage element at every dispatch for validity. This checking increases both the aggregate storage used and the CPU consumed. This is the default and the recommended setting.

OFF

Disables the storage testing.

Example

```
 VERIFY_MEMORY ON
IPN268I VERIFY_MEMORY now set to ON

 VERIFY_MEMORY OFF
IPN268I VERIFY_MEMORY now set to OFF
```

Notes

The following notes apply:

- When VERIFY_MEMORY ON is in effect, the storage manager, which normally places random “guard bytes” in the area immediately adjacent to each storage element, also tests whether those guard bytes are in place when freeing a storage block. If the bytes are missing, then a storage-overlay problem exists, and the TCP/IP stack issues a Vital-level error message to report the problem.
- Although this testing is expensive in terms of CPU required, it can catch most cases of storage corruption while the offending program is still in control.
- Cycling the stack with STORMON coded in the EXEC card’s PARM field also enables storage monitoring. The stack also displays the memory location of the storage manager phase (IPN900D).

Related Commands

QUERY STOR

Displays detailed information on memory use in the TCP/IP FOR VSE partition.

WAITFOR

The WAITFOR command suspends TCP/IP FOR VSE initialization until the specified event has occurred.

You can issue this command more than once, with a different parameter for each instance. There is no default, and so a missing parameter will result in the command not being performed.

Syntax

WAITfor {VTam|IPNEt|TIME[=]*nn*}

Arguments

VTam
Initialization pauses until the stack has detected that VTAM has initialized.

IPNEt
This parameter does nothing, and the request to wait is ignored. It was used in earlier releases and is included for compatibility.

TIME=*nn*
Initialization pauses *nn* seconds.

Note: "WAIT TIME *nn*" (without the **"="** delimiter) is accepted.

Example

```
 WAITFOR VTAM
IPN394I Testing for an active VTAM partition
IPN395I Waiting for VTAM to become active
IPN395I Waiting for VTAM to become active
IPN396I VTAM is active and functioning
```

Notes

If WAITFOR is encountered as a command after the stack's dispatcher has become active, it is ignored.

Related Commands

DEFINE TELNETD
Creates a TN3270 or TN3270E daemon

INCLUDE
Includes a library member in the initialization parameter set.

4

Deprecated Commands

The following TCP/IP FOR VSE commands are no longer supported and should not be used in 2.2.*x*. This list is cumulative and may include commands that were deprecated in earlier releases.

Some of these obsolete commands may appear to be accepted by the stack. Their settings are no longer referenced, however, and their actions no longer have any effect. Commands that fall into this category will be removed in future updates, so it is best to delete them from scripts now.

Command	Last Used	Notes
ACTIVATE	1.5E	Replaced by RESUME
ARPDELETE	1.5E	ARP processing redesigned
ATTACH		
CONNECT_QUEUE		No longer required
DEBUG		Replaced by DIAGNOSE DEBUG
DEFINE FILESYS		Replaced by EXEC FILESYS utility
DELETE SECURITY	1.5E	Replaced by SECURITY
DIAGNOSE_SECEXIT		Replaced by DIAGNOSE
DISPATCH		
DISPATCH_TIME		
DYNAMIC_ROUTE		No longer required
FIXED_RETRANSMIT		No longer required
FULL_CETIERROR	1.5E	Hardware no longer supported
FULL_SEGMENT		No longer required
HOLD		No longer required

Command	Last Used	Notes
LISTIDCAMS		No longer required
LOCALECB		No longer required
MATCH_MESSAGE		No longer required
NFS		No longer required
QUERY SUSPENDED	1.5E	
REDISPATCH		
RELEASE	1.5E	
REUSE		No longer required
REXX_SUPPORT		No longer required
SECURITY_ARP		Replaced by SECURITY
SECURITY_IP		Replaced by SECURITY
SET ARP_TIME	1.5E	ARP processing redesigned
SET CLOSE_DEPTH	1.5E	Inbound message control redesigned
SET CONNECT_SEQ		Replaced by CONNECT_SEQUENCE
SET CONSOLE_PORT		Function has been replaced by the SEE-TCP/IP FOR VSE optional feature
SET MESSAGE INFO		Replaced by MODIFY LOG
SET REUSE_SIZE	1.5E	Replaced by the IBBLOK command
SET TRANSFER_BUFFERS		Replaced by DEFINE FTPD
SOCKCHECK		No longer required