



iSeries

Operating System/400 Commands  
Starting with STRDBMON (Start Database Monitor)

*Version 5 Release 3*







@server

iSeries

Operating System/400 Commands

Starting with STRDBMON (Start Database Monitor)

*Version 5 Release 3*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 493.

**First Edition (May 2004)**

This edition applies to version 5, release 3, modification 0 of Operating System/400 (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

© Copyright International Business Machines Corporation 1998, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

## Start Database Monitor (STRDBMON) . . . 1

Parameters . . . . .	1
File to receive output (OUTFILE) . . . . .	2
Output member options (OUTMBR) . . . . .	2
Job name (JOB) . . . . .	2
Type of records (TYPE) . . . . .	3
Force record write (FRCRCD) . . . . .	3
Comment (COMMENT) . . . . .	3
Examples . . . . .	3
Error messages . . . . .	4

## Start Data Base Reader (STRDBRDR) . . . 5

Parameters . . . . .	5
Data base file (FILE) . . . . .	5
Member (MBR) . . . . .	6
Job queue (JOBQ) . . . . .	6
Queue for reader messages (MSGQ) . . . . .	7
Reader (RDR) . . . . .	7
Examples . . . . .	7
Error messages . . . . .	7

## Start DFU (STRDFU) . . . . . 9

Error messages for STRDFU . . . . .	9
Parameters . . . . .	9
Option (OPTION) . . . . .	9
DFU program (DFUPGM) . . . . .	10
Data base file (FILE) . . . . .	10
Member (MBR) . . . . .	10
Examples . . . . .	10
Error messages . . . . .	11

## Start Directory Shadowing (STRDIRSHD) . . . . . 13

Parameters . . . . .	13
Examples . . . . .	13
Error messages . . . . .	13

## Start Diskette Reader (STRDKTRDR) 15

Parameters . . . . .	15
Diskette device (DEV) . . . . .	15
Diskette label (LABEL) . . . . .	16
Volume (VOL) . . . . .	16
Job queue (JOBQ) . . . . .	16
Queue for reader messages (MSGQ) . . . . .	17
Reader (RDR) . . . . .	17
Creation date (CRTDATE) . . . . .	17
Code (CODE) . . . . .	18
Examples . . . . .	18
Error messages . . . . .	18

## Start Diskette Writer (STRDKTWTR) . . . 21

Parameters . . . . .	21
Diskette device (DEV) . . . . .	22
Output queue (OUTQ) . . . . .	22

Queue for writer messages (MSGQ) . . . . .	22
Writer (WTR) . . . . .	23
Auto-end options (AUTOEND) . . . . .	23
Spoiled file (FILE) . . . . .	23
Job name (JOB) . . . . .	24
Spoiled file number (SPLNBR) . . . . .	24
Examples . . . . .	24
Error messages . . . . .	24

## Start DNS Query (STRDNSQRY) . . . . . 27

Parameters . . . . .	30
Domain Name Server (DMNNAMSVR) . . . . .	31
Examples . . . . .	31
Error messages . . . . .	31

## Start Disk Reorganization (STRDSKRGZ) . . . . . 33

Parameters . . . . .	33
Auxiliary storage pool ID (ASP) . . . . .	33
ASP device (ASPDEV) . . . . .	33
Time limit (TIMLMT) . . . . .	34
Examples . . . . .	34
Error messages . . . . .	34

## Start Education (STREDU) . . . . . 37

Parameters . . . . .	37
Examples . . . . .	37
Error messages . . . . .	37

## Start 3270 Display Emulation (STREML3270) . . . . . 39

Parameters . . . . .	39
Emulation controller (EMLCTL) . . . . .	40
Emulation device (EMLDEV) . . . . .	41
Emulation location (EMLLOC) . . . . .	41
Display device, batch only (DSPDEV) . . . . .	41
Page Up (Roll Down) key (PAGEUP) . . . . .	41
Page Down (Roll Up) key (PAGEDOWN) . . . . .	42
Test Request key (TESTREQ) . . . . .	42
Cursor Select key (CSRSLT) . . . . .	43
SNA DBCS 3270PC emulation (IGCEMLPC) . . . . .	43
Emulation printer device (EMLPRTDEV) . . . . .	43
Timeout wait for host (INZWAIT) . . . . .	43
Numeric lock keyboard (NUMLCK) . . . . .	44
Handle nulls (NULLS) . . . . .	44
Host signon/logon command (LOGON) . . . . .	44
Wait response (WAITRSP) . . . . .	45
End emulation conditions (ENDCOND) . . . . .	45
Attention emulation menu (ATNEMLMNU) . . . . .	46
Function key program (FKEYPGM) . . . . .	46
Keyboard language type (KBdtype) . . . . .	47
Outgoing translation table (TRNTBLOUT) . . . . .	48
Incoming translation table (TRNTBLIN) . . . . .	48
Configuration entry (EMLCFGE) . . . . .	49

Examples . . . . .	49
Error messages . . . . .	49

**Start EPM Environment (STREPMENV) 53**

Error messages for STREPMENV . . . . .	53
Parameters . . . . .	53
Environment Name (EPMENV). . . . .	53
Environment Program (ROOTPGM) . . . . .	53
Examples . . . . .	54
Error messages . . . . .	54

**Start Font Management Aid (STRFMA) 55**

Error messages for STRFMA. . . . .	55
Parameters . . . . .	55
Examples . . . . .	55
Error messages . . . . .	55

**Start Host Server (STRHOSTSVR) . . . . . 57**

Error messages for STRHOSTSVR . . . . .	57
Parameters . . . . .	58
Server type (SERVER) . . . . .	58
Required protocol (RQDPCL) . . . . .	59
Examples . . . . .	59
Error messages . . . . .	59

**Start IDDU (STRIDD) . . . . . 61**

Parameters . . . . .	61
Examples . . . . .	61
Error messages . . . . .	61

**Start IP over SNA Interface (STRIPSIFC) . . . . . 63**

Parameters . . . . .	63
Internet address (INTNETADR). . . . .	63
Examples . . . . .	63
Error messages . . . . .	63

**Start ITF (STRITF) . . . . . 65**

Parameters . . . . .	65
Remote location (RMTLOCNAME) . . . . .	65
Examples . . . . .	65
Error messages . . . . .	65

**Start Journal (STRJRN) . . . . . 67**

Parameters . . . . .	68
Objects (OBJ) . . . . .	68
File identifier (OBJFID) . . . . .	69
Journal (JRN). . . . .	69
Directory subtree (SUBTREE) . . . . .	69
Name pattern (PATTERN) . . . . .	69
New objects inherit journaling (INHERIT) . . . . .	70
Images (IMAGES) . . . . .	70
Omit journal entry (OMTJRNE). . . . .	71
Examples . . . . .	71
Error messages . . . . .	72

**Start Journal Access Path (STRJRNAP) 73**

Parameters . . . . .	74
Journalled file (FILE) . . . . .	74

Journal (JRN). . . . .	74
Examples . . . . .	75
Error messages . . . . .	75

**Start Journal Object (STRJRNOBJ) . . . . . 77**

Parameters . . . . .	78
Object (OBJ) . . . . .	78
Object type (OBJTYPE) . . . . .	78
Journal (JRN). . . . .	79
Images (IMAGES) . . . . .	79
Examples . . . . .	79
Error messages . . . . .	79

**Start Journal Physical File (STRJRNPFF) 81**

Parameters . . . . .	82
Physical file to be journaled (FILE) . . . . .	82
Journal (JRN). . . . .	82
Record images (IMAGES). . . . .	83
Journal entries to be omitted (OMTJRNE) . . . . .	83
Examples . . . . .	83
Error messages . . . . .	83

**Start Mode (STRMOD) . . . . . 87**

Parameters . . . . .	87
Remote location (RMTLOCNAME) . . . . .	87
Device (DEV). . . . .	87
Mode (MODE) . . . . .	88
Local location (LCLLOCNAME) . . . . .	88
Remote network identifier (RMTNETID). . . . .	88
Examples . . . . .	89
Error messages . . . . .	89

**Start Mail Server Framework (STRMSF) 91**

Parameters . . . . .	91
How to process mail messages (MSGOPT) . . . . .	91
Number of MSF jobs (NBRMSFJOB) . . . . .	91
Examples . . . . .	92
Error messages . . . . .	92

**Start NFS Server (STRNFSSVR) . . . . . 93**

Parameters . . . . .	93
Server daemon (SERVER). . . . .	94
Number of server daemons (NBRSVR) . . . . .	94
Number of block I/O daemons (NBRBIO) . . . . .	94
Retrieve RPC registration (RTVRPCREG) . . . . .	95
Timeout for start of daemon (STRJOBTIMO) . . . . .	95
Examples . . . . .	95
Error messages . . . . .	95

**Start NetWare Connection (STRNTWCNN) . . . . . 97**

Parameters . . . . .	97
Server (SERVER) . . . . .	98
Server type (SVRTYPE) . . . . .	98
NDS tree (NDSTREE) . . . . .	98
Authorized user profile (AUTUSR) . . . . .	98
NDS context (NDSCTX) . . . . .	99
NetWare user (NTWUSER) . . . . .	99
Password (PASSWORD) . . . . .	99

Authorized job (AUTJOB)	99
Connection type (CNNTYPE)	100
Connection idle time (CNNIDLTIME)	100
Examples	100
Error messages	100

### **Start Object Conversion (STROBJCVN) . . . . . 101**

Parameters	101
Library (LIB)	101
Object type (OBJTYPE)	102
Priority (PTY)	102
Examples	103
Error messages	103

### **Start Pass-Through (STRPASTHR) 105**

Parameters	105
Remote location (RMTLOCNAME)	105
APPC device (CNNDEV)	106
Virtual controller (VRTCTL)	106
Virtual display device (VRTDEV)	107
Mode (MODE)	107
Local location (LCLLOCNAME)	107
Remote network identifier (RMTNETID)	107
System request program (SRQ10PGM)	108
User profile (RMTUSER)	108
User password (RMTPWD)	109
Initial program to call (RMTINLPGM)	109
Initial menu (RMTINLMNU)	109
Current library (RMTCURLIB)	110
Display option (PASTHRSCN)	110
Examples	110
Error messages	111

### **Start PC Command (STRPCCMD) . . . . . 115**

Error messages for STRPCCMD	115
Parameters	115
PC command (PCCMD)	115
Pause (PAUSE)	115
Examples	116
Error messages	116

### **Start PC Organizer (STRPCO) . . . . . 117**

Error messages for STRPCO	117
Parameters	117
Text Assist (PCTA)	117
Examples	117
Error messages	117

### **Start Performance Explorer (STRPEX) 119**

Parameters	119
Session ID (SSNID)	119
Option (OPTION)	120
Definition (DFN)	120
Filter (FTR)	120
Examples	120
Error messages	121
Error messages for RMVPEXDFN	121

### **Start Performance Collection (STRPFCOL) . . . . . 123**

Parameters	123
Default collection profile (COLPRF)	123
Cycle collection (CYCCOL)	124
Examples	124
Error messages	124

### **Start Performance Trace (STRPFRTRC) . . . . . 127**

Parameters	127
Trace table size (SIZE)	127
Omit trace points (OMTTRCPNT)	128
Job types (JOBTYPE)	128
Job trace interval (JOBTRCITV)	129
Examples	129
Error messages	129

### **Start Program Export List (STRPGMEXP) . . . . . 131**

Parameters	131
Program level (PGMLVL)	131
Signature level check (LVLCHK)	131
Signature (SIGNATURE)	131
Examples	132
Error messages	132

### **Start Programmer Menu (STRPGMMNU) . . . . . 133**

Parameters	133
Source file (SRCFILE)	133
Source library (SRCLIB)	134
Object library (OBJLIB)	134
Job description (JOBDD)	134
Allow changes (ALWUSRCHG)	134
Option 3 exit program (EXITPGM)	134
Delete option (DLTOPT)	135
Examples	136
Error messages	137

### **Start Program Profiling (STRPGMPRF) 139**

Parameters	139
Examples	139
Error messages	139

### **Start Prestart Jobs (STRPJ) . . . . . 141**

Parameters	141
Subsystem (SBS)	141
Program (PGM)	142
Examples	142
Error messages	142

### **Start Printer Emulation (STRPRTEML) 143**

Parameters	143
Emulation controller (EMLCTL)	144
Emulation device (EMLDEV)	144
Emulation location (EMLLOC)	145
Print device (PRTDEV)	145

Job name (JOB)	145
End Bracket eject (ENDBKTEJT)	145
Printer file (PRTFILE)	146
Timeout wait eject (TIMOUTEJT)	146
Defer printing spool output (DFRPRTOUT)	146
Spool output (SPOOL)	147
Open printer file (OPNPRTF)	147
Print positions per line (NUMCOL)	147
Lines per page (NUMLIN)	148
Lines per inch (LPI)	148
Message queue (MSGQ)	148
Language character set (CHRSET)	149
Submit job (SBMJOB)	149
Job description (JOBDD)	149
Outgoing translation table (TRNTBLOUT)	150
Job queue (JOBQ)	150
End emulation conditions (ENDCOND)	151
Acknowledge form feed (FORMFEED)	151
Configuration entry (EMLCFGE)	151
Examples	152
Error messages	152

**Start Printer Writer (STRPRTWTR) 155**

Parameters	155
Printer (DEV)	156
Output queue (OUTQ)	156
Queue for writer messages (MSGQ)	157
Form type options (FORMTYPE)	157
File separators (FILESEP)	158
Drawer for separators (SEPDRAWER)	158
Writer (WTR)	159
Auto-end options (AUTOEND)	159
Allow direct print (ALWDRTPRT)	159
Align page (ALIGN)	160
Initialize printer (INIT)	160
Spooled file (FILE)	160
Job name (JOB)	161
Spooled file number (SPLNBR)	161
Job system name (JOBSYSNAME)	161
Spooled file created (CRTDATE)	162
Starting page (PAGE)	162
Examples	163
Error messages	163

**Start Query Management Proc (STRQMPC) 165**

Parameters	165
Source member (SRCMBR)	165
Source file (SRCFILE)	166
Relational database (RDB)	166
Connection Method (RDBCNNMTH)	166
User (USER)	167
Password (PASSWORD)	167
Naming convention (NAMING)	167
Allow information from QRYDFN (ALWQRYDFN)	167
Command source file (CMDSRCFILE)	168
Source member (CMDSRCMBR)	168
Display screens (ALWDSPLAY)	169
Examples	169
Error messages	169

**Start Query Management Query (STRQMQR) 171**

Parameters	171
Query management query (QMQR)	172
Output (OUTPUT)	172
Query management report form (QMFORM)	173
File to receive output (OUTFILE)	173
Output member options (OUTMBR)	173
Date and time (DATETIME)	174
Page numbers (PAGNBR)	174
Relational database (RDB)	174
Connection Method (RDBCNNMTH)	175
User (USER)	175
Password (PASSWORD)	175
Naming convention (NAMING)	176
Allow information from QRYDFN (ALWQRYDFN)	176
Set variables (SETVAR)	176
Examples	176
Error messages	177

**Start Query (STRQR) 179**

Parameters	179
Examples	179
Error messages	179

**Start QSH (STRQSH) 181**

Error messages for STRQSH	182
Parameters	182
Command (CMD)	182
Examples	183
Error messages	183

**Start Question and Answer (STRQST) 185**

Parameters	185
Q/A database (QSTDB)	185
Lib containing Q/A database (LIB)	185
Examples	186
Error messages	186

**Start REXX Procedure (STRREXPC) 187**

Parameters	187
Source member (SRCMBR)	187
Source file (SRCFILE)	187
Parameters (PARM)	188
Command environment (CMDENV)	188
Exit program (EXITPGM)	189
Examples	190
Error messages	190

**Start Remote Support (STRRMTSPT) 191**

Parameters	191
Device class (DEVCLS)	192
Display type (DSPTYPE)	192
Display model (DSPMODEL)	193
Address (STNADR)	193
User profile (USRPRF)	193
Resource name (RSRCNAME)	194
Remote location (RMTLOCNAME)	194
Local location (LCLLOCNAME)	194



Remote network identifier (RMTNETID) . . . . .	195
Maximum length of request unit (MAXLENRU) . . . . .	195
Data compression (DTACPR) . . . . .	195
Inbound data compression (INDTACPR) . . . . .	196
Outbound data compression (OUTDTACPR) . . . . .	197
Modem (MODEM) . . . . .	197
Modem remote system (MDMRMTSYS) . . . . .	198
Examples . . . . .	198
Error messages . . . . .	199

<b>Start Remote Writer (STRRMTWTR) . . . . .</b>	<b>201</b>
Parameters . . . . .	201
Output queue (OUTQ) . . . . .	202
Queue for writer messages (MSGQ) . . . . .	202
Form type options (FORMTYPE) . . . . .	203
Writer (WTR) . . . . .	203
Auto-end options (AUTOEND) . . . . .	204
Examples . . . . .	204
Error messages . . . . .	204

<b>Start S/36 Session (STRS36) . . . . .</b>	<b>205</b>
Parameters . . . . .	205
Menu (MENU) . . . . .	205
Current library (CURLIB) . . . . .	205
Procedure or program (PRC) . . . . .	206
Force menu (FRCMNU) . . . . .	206
Examples . . . . .	206
Error messages . . . . .	206

<b>Start S/36 Procedure (STRS36PRC) . . . . .</b>	<b>209</b>
Parameters . . . . .	209
Procedure (PRC) . . . . .	209
Current library (CURLIB) . . . . .	209
Procedure parameters (PARM) . . . . .	210
Examples . . . . .	210
Error messages . . . . .	210

<b>Start Subsystem (STRSBS) . . . . .</b>	<b>211</b>
Parameters . . . . .	212
Subsystem description (SBSD) . . . . .	212
Examples . . . . .	212
Error messages . . . . .	213

<b>Start Search Index (STRSCHIDX) . . . . .</b>	<b>215</b>
Parameters . . . . .	215
Search index (SCHIDX) . . . . .	215
Examples . . . . .	216
Error messages . . . . .	216

<b>Start Support Network (STRSPTN) . . . . .</b>	<b>217</b>
Parameters . . . . .	217
Account (ACCOUNT) . . . . .	217
User ID (SPTUSRID) . . . . .	217
Password (SPTPWD) . . . . .	218
Device description (FEADEV) . . . . .	218
Destination application (DESTAPP) . . . . .	218
Examples . . . . .	218
Error messages . . . . .	218

<b>Start Service Agent (STRSRVAGT) . . . . .</b>	<b>219</b>
Parameters . . . . .	219
Type (TYPE) . . . . .	219
Activation password (ACTPWD) . . . . .	220
Master password (MSTPWD) . . . . .	220
Examples . . . . .	220
Error messages . . . . .	220

<b>Start Service Job (STRSRVJOB) . . . . .</b>	<b>221</b>
Parameters . . . . .	221
Job name (JOB) . . . . .	221
Duplicate job option (DUPJOB OPT) . . . . .	222
Examples . . . . .	222
Error messages . . . . .	222

<b>Start System Service Tools (STRSST) . . . . .</b>	<b>223</b>
Parameters . . . . .	223
Examples . . . . .	223
Error messages . . . . .	223

<b>Start TCP/IP (STRTCP) . . . . .</b>	<b>225</b>
Parameters . . . . .	227
Start application servers (STRSVR) . . . . .	227
Start TCP/IP interfaces (STRIFC) . . . . .	228
Start point-to-point profiles (STRPTPPRF) . . . . .	228
Examples . . . . .	228
Error messages . . . . .	229

<b>Start TCP/IP Interface (STRTCPIFC) . . . . .</b>	<b>231</b>
Parameters . . . . .	231
Internet address (INTNETADR) . . . . .	231
Examples . . . . .	232
Error messages . . . . .	232

<b>Start Point-to-Point TCP/IP (STRTCPPTP) . . . . .</b>	<b>235</b>
Parameters . . . . .	235
Configuration profile (CFGPRF) . . . . .	235
Script dialog output (OUTPUT) . . . . .	236
Restart (RESTART) . . . . .	236
Send inquiry message (SNDINQMSG) . . . . .	236
Autodelete configuration (AUTODLTCFG) . . . . .	237
Examples . . . . .	237
Error messages . . . . .	238

<b>Start TCP/IP Server (STRTCP SVR) . . . . .</b>	<b>239</b>
Parameters . . . . .	239
Server application (SERVER) . . . . .	240
Restart server (RESTART) . . . . .	243
HTTP server (HTTPSVR) . . . . .	244
DNS server (DNSSVR) . . . . .	245
TCM server (TCMSVR) . . . . .	246
ASFTOMCAT server (TOMCATSVR) . . . . .	246
SNTP service (NTPSRV) . . . . .	247
Examples . . . . .	247
Error messages . . . . .	248

<b>Start TIE Session (STRTISSN) . . . . .</b>	<b>249</b>
Parameters . . . . .	249

User ID (SPTUSRID) . . . . .	249	Job queue (JOBQ) . . . . .	278
Password (SPTPWD) . . . . .	249	Routing data (RTGDTA) . . . . .	278
Account (ACCOUNT) . . . . .	249	Request data or command (RQSDTA) . . . . .	279
Examples . . . . .	250	Examples . . . . .	279
Error messages . . . . .	250	Error messages . . . . .	279
<b>Start Trace (STRTRC) . . . . .</b>	<b>251</b>	<b>Transfer Pass-Through (TFRPASTHR) 281</b>	
Parameters . . . . .	251	Parameters . . . . .	281
Session ID (SSNID) . . . . .	252	To job (TOJOB) . . . . .	281
Job name (JOB) . . . . .	253	Examples . . . . .	282
Job types (JOBTYPE) . . . . .	254	Error messages . . . . .	282
Maximum storage to use (MAXSTG) . . . . .	254	<b>Transfer Secondary Job</b>	
Trace full (TRCFULL) . . . . .	254	<b>(TFRSECJOB) . . . . . 283</b>	
Trace type (JOBTRCTYPE) . . . . .	255	Parameters . . . . .	283
Trace type (TRCTYPE) . . . . .	255	Examples . . . . .	283
Trace filter (TRCFTR) . . . . .	256	Error messages . . . . .	283
Watch for message (WCHMSG) . . . . .	256	<b>Trace TCP/IP Route (TRACEROUTE) 285</b>	
Watched message queue (WCHMSGQ) . . . . .	257	Parameters . . . . .	285
Watched job (WCHJOB) . . . . .	257	Remote system (RMTSYS) . . . . .	286
Watch for LIC log entry (WCHLICLOG) . . . . .	258	Range of hops to probe (RANGE) . . . . .	286
Length of time to watch (WCHTIMO) . . . . .	259	Probes sent per hop (PROBES) . . . . .	286
Trace program (TRCPGM) . . . . .	260	Response wait time (WAITTIME) . . . . .	286
Time interval (TRCPGMITV) . . . . .	261	Packet length (in bytes) (PKTLEN) . . . . .	287
Examples . . . . .	261	Output (OUTPUT) . . . . .	287
Error messages . . . . .	263	Data queue (DTAQ) . . . . .	287
<b>Start Trap Manager (STRTRPMGR) 265</b>		Address version format (ADRVERFMT) . . . . .	287
Parameters . . . . .	265	Source IP address (LCLINTNETA) . . . . .	288
Forward Trap (FWDTRP) . . . . .	265	Base remote port (RMTPORT) . . . . .	288
Examples . . . . .	265	Lookup host names (NAMELOOKUP) . . . . .	288
Error messages . . . . .	266	Probing protocol (PROBEPCL) . . . . .	288
<b>Transfer Batch Job (TFRBCHJOB) 267</b>		Allow fragmentation (FRAGMENT) . . . . .	289
Parameters . . . . .	268	Examples . . . . .	289
Job queue (JOBQ) . . . . .	268	Error messages . . . . .	290
Routing data (RTGDTA) . . . . .	268	<b>Trace ASP Balance (TRCASPBAL) . . . 291</b>	
Request data or command (RQSDTA) . . . . .	268	Parameters . . . . .	291
Examples . . . . .	269	Trace option setting (SET) . . . . .	292
Error messages . . . . .	269	ASP number (ASP) . . . . .	292
<b>Transfer Control (TFRCTL) . . . . . 271</b>		ASP device (ASPDEV) . . . . .	292
Parameters . . . . .	271	Time limit (TIMLMT) . . . . .	293
Program (PGM) . . . . .	271	Examples . . . . .	293
Parameter CL variable names (PARM) . . . . .	272	Error messages . . . . .	293
Examples . . . . .	272	<b>Trace Connection (TRCCNN) . . . . . 295</b>	
Error messages . . . . .	272	Parameters . . . . .	295
<b>Transfer to Group Job (TFRGRPJOB) 273</b>		Trace option setting (SET) . . . . .	297
Parameters . . . . .	273	Trace type (TRCTYPE) . . . . .	297
Group job (GRPJOB) . . . . .	273	Trace full (TRCFULL) . . . . .	297
Initial group program (INLGRPPGM) . . . . .	273	Trace table name (TRCTBL) . . . . .	297
Special environment (SPCENV) . . . . .	274	Size (SIZE) . . . . .	298
Restore display (RSTDSP) . . . . .	274	Trace data to be formatted (FMTDTA) . . . . .	298
Text 'description' (TEXT) . . . . .	274	Coded character set identifier (CCSID) . . . . .	299
Examples . . . . .	275	Job name (JOB) . . . . .	299
Error messages . . . . .	275	Spooled file number (SPLNBR) . . . . .	299
<b>Transfer Job (TFRJOB) . . . . . 277</b>		Job system name (JOBSYSNAME) . . . . .	300
Parameters . . . . .	278	Spooled file created (CRTDATE) . . . . .	300
		TCP/IP data (TCPDTA) . . . . .	301

Watch for message (WCHMSG) . . . . .	301
Watched message queue (WCHMSGQ) . . . . .	302
Watched job (WCHJOB) . . . . .	303
Watch for LIC log entry (WCHLICLOG) . . . . .	303
Length of time to watch (WCHTIMO) . . . . .	304
Trace program (TRCPGM) . . . . .	305
Time interval (TRCPGMITV) . . . . .	306
Examples . . . . .	307
Error messages . . . . .	308

**Trace CPI Communications (TRCCPIC) 309**

Parameters . . . . .	309
Trace option setting (SET) . . . . .	310
Maximum storage to use (MAXSTG) . . . . .	310
Trace full (TRCFULL) . . . . .	310
User data length (DTALEN) . . . . .	310
Output (OUTPUT) . . . . .	311
Output file (OUTFILE) . . . . .	311
Output member options (OUTMBR) . . . . .	311
Examples . . . . .	312
Error messages . . . . .	312

**Trace ICF (TRCICF) . . . . . 315**

Parameters . . . . .	315
Trace option setting (SET) . . . . .	315
Maximum storage to use (MAXSTG) . . . . .	316
Trace full (TRCFULL) . . . . .	316
User data length (DTALEN) . . . . .	316
Output (OUTPUT) . . . . .	316
Output file (OUTFILE) . . . . .	317
Output member options (OUTMBR) . . . . .	317
Examples . . . . .	317
Error messages . . . . .	318

**Trace Internal (TRCINT) . . . . . 321**

Parameters . . . . .	321
Trace option setting (SET) . . . . .	324
Trace table name (TRCTBL) . . . . .	325
Trace table size (SIZE) . . . . .	325
Trace full (TRCFULL) . . . . .	326
Trace type (TRCTYPE) . . . . .	326
Job name (JOB) . . . . .	328
Thread ID to include (SLTTHD) . . . . .	329
Server type (SVRTYPE) . . . . .	330
Task name (TASK) . . . . .	330
Task number (TASKNBR) . . . . .	331
Select trace points (SLTTRCPNT) . . . . .	331
Omit trace points (OMTTRCPNT) . . . . .	332
Stop trace points (STOPTRCPNT) . . . . .	332
Job types (JOBTYPE) . . . . .	333
Job trace interval (JOBTRCITV) . . . . .	334
TCP/IP data (TCPDTA) . . . . .	334
Sockets data (SCKDTA) . . . . .	335
Device (DEV) . . . . .	336
Controller (CTL) . . . . .	337
Line (LIN) . . . . .	337
Network interface (NWI) . . . . .	338
Network Server (NWS) . . . . .	339
Hardware Resource (RSRCNAME) . . . . .	339
Device (OUTDEV) . . . . .	340

Task information (TASKINF) . . . . .	340
Output (OUTPUT) . . . . .	340
File to receive output (OUTFILE) . . . . .	340
Output member options (OUTMBR) . . . . .	341
Watch for message (WCHMSG) . . . . .	341
Watched message queue (WCHMSGQ) . . . . .	342
Watched job (WCHJOB) . . . . .	342
Watch for LIC log entry (WCHLICLOG) . . . . .	343
Length of time to watch (WCHTIMO) . . . . .	344
Trace program (TRCPGM) . . . . .	344
Time interval (TRCPGMITV) . . . . .	346
Examples . . . . .	346
Error messages . . . . .	347

**Trace Job (TRCJOB) . . . . . 351**

Parameters . . . . .	351
Trace option setting (SET) . . . . .	352
Trace type (TRCTYPE) . . . . .	352
Maximum storage to use (MAXSTG) . . . . .	353
Trace full (TRCFULL) . . . . .	353
Program to call before trace (EXITPGM) . . . . .	353
Select procedures to trace (SLTPRC) . . . . .	354
Thread ID to include (SLTTHD) . . . . .	354
Output (OUTPUT) . . . . .	355
File to receive output (OUTFILE) . . . . .	355
Output member options (OUTMBR) . . . . .	355
Examples . . . . .	356
Error messages . . . . .	356

**Trace REXX (TRCREX) . . . . . 359**

Parameters . . . . .	359
Trace option setting (SET) . . . . .	359
Examples . . . . .	360
Error messages . . . . .	360

**Trace TCP/IP Application (TRCTCPAPP) . . . . . 361**

Parameters . . . . .	361
TCP/IP application (APP) . . . . .	363
Trace option setting (SET) . . . . .	364
Maximum storage for trace (MAXSTG) . . . . .	364
Trace full action (TRCFULL) . . . . .	365
Additional traces (ADLTRC) . . . . .	365
Trace program (TRCPGM) . . . . .	366
Trace title (TITLE) . . . . .	368
User profile (USER) . . . . .	368
Recipient mail address (MAILADR) . . . . .	369
Recipient host name (HOST) . . . . .	369
Remote network address (RMTNETADR) . . . . .	369
Local network address (LCLNETADR) . . . . .	370
Device description (DEVD) . . . . .	370
Device type (DEVTYPE) . . . . .	371
Trace point (TRCPNT) . . . . .	371
Argument list (ARGLIST) . . . . .	371
Virtual private network server (VPNSVR) . . . . .	372
Certificate services type (CERTTYPE) . . . . .	372
Domain name service (DNS) . . . . .	372
PPP connection profile (PPPCNNPRF) . . . . .	373
TCP/IP data to trace (TCPTRCDTA) . . . . .	373
QOS trace type (QOSTRCTYPE) . . . . .	373

HTTP server instance (HTTPSVR) . . . . .	373
Trace level (TRCLVL) . . . . .	373
Packet rules trace points (PKTTRCPNT) . . . . .	374
Configuration object (CFGOBJ) . . . . .	374
Type (CFGTYPE) . . . . .	374
Watch for message (WCHMSG) . . . . .	374
Watched message queue (WCHMSGQ) . . . . .	375
Watched job (WCHJOB) . . . . .	376
Watch for LIC log entry (WCHLICLOG) . . . . .	376
Length of time to watch (WCHTIMO) . . . . .	377
Time interval (TRCPGMITV) . . . . .	378
Examples . . . . .	378
Error messages . . . . .	379

**Trace TCP/IP Route (TRCTCPRTE) . . . . . 381**

Parameters . . . . .	381
Remote system (RMTSYS) . . . . .	382
Range of hops to probe (RANGE) . . . . .	382
Probes sent per hop (PROBES) . . . . .	382
Response wait time (WAITTIME) . . . . .	382
Packet length (in bytes) (PKTLEN) . . . . .	383
Output (OUTPUT) . . . . .	383
Data queue (DTAQ) . . . . .	383
Address version format (ADRVERFMT) . . . . .	383
Source IP address (LCLINTNETA) . . . . .	384
Base remote port (RMTPORT) . . . . .	384
Lookup host names (NAMELOOKUP) . . . . .	384
Probing protocol (PROBEPCL) . . . . .	384
Allow fragmentation (FRAGMENT) . . . . .	385
Examples . . . . .	385
Error messages . . . . .	386

**Remove Mounted FS (UNMOUNT) . . . . . 387**

Parameters . . . . .	387
Type of file system (TYPE) . . . . .	387
Directory mounted over (MNTOVRDIR) . . . . .	388
Mounted file system (MFS) . . . . .	388
Examples . . . . .	388
Error messages . . . . .	388

**Update Data with Temp Program (UPDDTA) . . . . . 391**

Error messages for UPDDTA . . . . .	391
Parameters . . . . .	391
Data base file (FILE) . . . . .	391
Member (MBR) . . . . .	391
Examples . . . . .	392
Error messages . . . . .	392

**Update Program (UPDPGM) . . . . . 393**

Parameters . . . . .	393
Program (PGM) . . . . .	394
Module (MODULE) . . . . .	394
Replacement library (RPLLIB) . . . . .	395
Bind service program (BNDSRVPGM) . . . . .	396
Bound *SRVPGM library name (SRVPGMLIB) . . . . .	396
Binding directory (BNDDIR) . . . . .	397
Activation group (ACTGRP) . . . . .	397
Creation options (OPTION) . . . . .	398
Listing detail (DETAIL) . . . . .	399

Examples . . . . .	399
Error messages . . . . .	399

**Update Service Program (UPDSRVPGM) . . . . . 401**

Parameters . . . . .	401
Service program (SRVPGM) . . . . .	402
Module (MODULE) . . . . .	403
Export (EXPORT) . . . . .	403
Export source file (SRCFILE) . . . . .	404
Export source member (SRCMBR) . . . . .	404
Replacement library (RPLLIB) . . . . .	405
Bind service program (BNDSRVPGM) . . . . .	405
Bound *SRVPGM library name (SRVPGMLIB) . . . . .	406
Binding directory (BNDDIR) . . . . .	406
Activation group (ACTGRP) . . . . .	407
Creation options (OPTION) . . . . .	407
Listing detail (DETAIL) . . . . .	408
Examples . . . . .	408
Error messages . . . . .	409

**Update System Information (UPDSYSINF) . . . . . 411**

Parameters . . . . .	411
Library (LIB) . . . . .	411
Type of information (TYPE) . . . . .	412
Examples . . . . .	412
Error messages . . . . .	412

**Verify APPC Connection (VFYAPPCNN) . . . . . 413**

Parameters . . . . .	413
Remote location (RMTLOCNAME) . . . . .	413
Mode (MODE) . . . . .	413
Remote user ID (RMTUSER) . . . . .	414
Remote password (RMTPWD) . . . . .	414
Message mode (MSGMODE) . . . . .	414
Packet length (in bytes) (PKTLEN) . . . . .	415
Number of iterations (NBRITER) . . . . .	415
Number of packets (NBRPKT) . . . . .	415
Echo (ECHO) . . . . .	415
Wait time (in seconds) (WAITTIME) . . . . .	416
Examples . . . . .	416
Error messages . . . . .	416

**Verify Communications (VFYCMN) . . . . . 417**

Parameters . . . . .	417
Verification type (VFYTYPE) . . . . .	417
Remote control point (RCPNAME) . . . . .	418
Network ID (NETID) . . . . .	418
User ID (USERID) . . . . .	418
Password (PASSWORD) . . . . .	418
Examples . . . . .	419
Error messages . . . . .	419

**Verify Image Catalog (VFYIMGCLG) . . . . . 421**

Parameters . . . . .	421
Image catalog (IMGCLG) . . . . .	421
Image catalog type (TYPE) . . . . .	421

Sort image catalog (SORT) . . . . .	422
Examples . . . . .	422
Error messages . . . . .	422

**Verify Link supporting LPDA-2 (VFYLNKLPDA) . . . . . 425**

Parameters . . . . .	425
Line (LINE) . . . . .	425
Test (TEST) . . . . .	425
Local DCE address (LCLDCEADR) . . . . .	426
Remote DCE address (RMTDCEADR) . . . . .	426
Output (OUTPUT) . . . . .	426
Number of sequences (SEQCOUNT) . . . . .	427
Remote DTE port (DTEPORT) . . . . .	427
DTE retry (DTERTY) . . . . .	427
DCE retry (DCERTY) . . . . .	427
Link status after test (VRYLNKSTS) . . . . .	428
Examples . . . . .	428
Error messages . . . . .	428

**Verify NetWare Aut Entry (VFYNTWAUTE) . . . . . 431**

Parameters . . . . .	431
Server type (SVRTYPE) . . . . .	431
NDS tree (NDSTREE) . . . . .	431
Server (SERVER) . . . . .	431
User profile (USRPRF) . . . . .	432
Examples . . . . .	432
Error messages . . . . .	432

**Verify OptiConnect Connections (VFYOPCCNN) . . . . . 433**

Parameters . . . . .	433
Examples . . . . .	433
Error messages . . . . .	433

**Verify Optical (VFYOPT) . . . . . 435**

Parameters . . . . .	435
Device (DEV) . . . . .	435
Examples . . . . .	435
Error messages . . . . .	435

**Verify Printer (VFYPRT) . . . . . 437**

Parameters . . . . .	437
Workstation printer device (DEV) . . . . .	437
Times to print (TIMES) . . . . .	438
Examples . . . . .	438
Error messages . . . . .	438

**Verify Service Agent (VFYSRVAGT) 439**

Parameters . . . . .	439
Type (TYPE) . . . . .	439
Error log identifier (ERRLOGID) . . . . .	439
Examples . . . . .	440
Error messages . . . . .	440

**Verify Service Configuration (VFYSRVCFG) . . . . . 441**

Parameters . . . . .	441
Service (SERVICE) . . . . .	441
Examples . . . . .	441
Error messages . . . . .	442

**Verify Tape (VFYTAP) . . . . . 443**

Parameters . . . . .	443
Device (DEV) . . . . .	443
Resource name (RSRCNAME) . . . . .	443
Examples . . . . .	443
Error messages . . . . .	444

**Verify TCP/IP Connection (VFYTCPCNN) . . . . . 445**

Parameters . . . . .	445
Remote system (RMTSYS) . . . . .	445
Remote internet address (INTNETADR) . . . . .	446
Address version format (ADRVERFMT) . . . . .	446
Message mode (MSGMODE) . . . . .	446
Packet length (in bytes) (PKTLEN) . . . . .	447
Number of packets (NBRPKT) . . . . .	447
Wait time (in seconds) (WAITTIME) . . . . .	447
Local internet address (LCLINTNETA) . . . . .	447
Type of service (TOS) . . . . .	447
IP time to live (hop limit) (IPTTL) . . . . .	448
Examples . . . . .	448
Error messages . . . . .	449

**Vary Configuration (VRYCFG) . . . . . 451**

Parameters . . . . .	453
Configuration object (CFGOBJ) . . . . .	453
Type (CFGTYPE) . . . . .	454
Status (STATUS) . . . . .	454
Range (RANGE) . . . . .	455
Vary on wait (VRYWAIT) . . . . .	455
Asynchronous vary off (ASCVRVYOFF) . . . . .	455
Reset (RESET) . . . . .	455
Resource name (RSRCNAME) . . . . .	455
Reset configuration file (RESETCFGF) . . . . .	456
Forced vary off (FRCVRVYOFF) . . . . .	456
Start TCP/IP interfaces (STRICPIFC) . . . . .	456
Submit multiple jobs (SBMMLTJOB) . . . . .	457
Job description (JOBDESC) . . . . .	457
Examples . . . . .	457
Error messages . . . . .	459

**Wait (WAIT) . . . . . 461**

Parameters . . . . .	461
CL var for responding device (DEV) . . . . .	461
Open file identifier (OPNID) . . . . .	462
Examples . . . . .	462
Error messages . . . . .	462

**When (WHEN) . . . . . 465**

Parameters . . . . .	465
Condition (COND) . . . . .	465
Command (THEN) . . . . .	466
Examples . . . . .	466
Error messages . . . . .	467

**Work with Active Jobs (WRKACTJOB) 469**

Parameters . . . . . 469  
Output (OUTPUT). . . . . 469  
Reset status statistics (RESET) . . . . . 469  
Subsystem (SBS) . . . . . 470  
CPU percent limit (CPUPCTLMT) . . . . . 470  
Response time limit (RSPLMT) . . . . . 470  
Sequence (SEQ). . . . . 470  
Job name (JOB). . . . . 471  
Automatic refresh interval (INTERVAL) . . . . . 472  
Examples. . . . . 472  
Error messages . . . . . 472

**Work with Alerts (WRKALR) . . . . . 475**

Parameters . . . . . 475  
Display option (DSPOPT) . . . . . 475  
Period (PERIOD) . . . . . 476  
Alert type (ALRTYPE) . . . . . 477  
Resource name (ALRRSC) . . . . . 478  
Alert resource type (ALRRSCTYPE) . . . . . 478  
User assigned (ASNUSER) . . . . . 478  
Group (GROUP) . . . . . 479  
Output (OUTPUT). . . . . 479  
Detail (DETAIL) . . . . . 479  
Examples. . . . . 480  
Error messages . . . . . 480

**Work with Alert Descriptions (WRKALRD) . . . . . 481**

Parameters . . . . . 481

Message identifier (MSGID) . . . . . 481  
Alert table (ALRTBL). . . . . 481  
Examples. . . . . 482  
Error messages . . . . . 482

**Work with Alert Table (WRKALRTBL) 485**

Parameters . . . . . 485  
Alert table (ALRTBL). . . . . 485  
Examples. . . . . 486  
Error messages . . . . . 486

**Work with APPN Status (WRKAPPNSTS) . . . . . 489**

Parameters . . . . . 489  
Option (OPTION) . . . . . 489  
Attached controller (CTL) . . . . . 489  
Remote network identifier (RMTNETID) . . . . . 490  
Remote control point (RMTLOCNAME) . . . . . 490  
Remote control point (RMTCPNAME) . . . . . 490  
Mode (MODE) . . . . . 491  
Transport connection ID (TCID) . . . . . 491  
Examples. . . . . 491  
Error messages . . . . . 492

**Appendix. Notices . . . . . 493**

Trademarks . . . . . 494  
Terms and conditions for downloading and printing publications . . . . . 495  
Code disclaimer information . . . . . 495

# Start Database Monitor (STRDBMON)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
 Examples  
 Error messages

The Start Database Monitor (STRDBMON) command starts the collection of database performance statistics for a specified job or all jobs on the system. The statistics are placed in a specified database file and member. If the file or member do not exist, one is created based on the QAQQDBMN file in library QSYS. If the file or member do exist, the record format of the file is checked to see if it is the same.

## Restrictions:

1. You cannot specify \*ALL on the JOB parameter if the monitor has already been started with the JOB(\*ALL) attribute. Only one specific monitor can be started on a specific job at a time. For example, STRDBMON JOB(\*) followed by another STRDBMON JOB(\*) within the same job is not allowed unless the job was ended using ENDDDBMON between the two STRDBMON command operations.
2. QTEMP cannot be specified as the library on the OUTFILE parameter unless JOB(\*) was also specified.
3. This command is conditionally threadsafe. For multithreaded jobs, this command is not threadsafe and may fail when the OUTFILE parameter is a distributed file or is a Distributed Data Management (DDM) file of type \*SNA.
4. JOB(\*ALL) requires the file specified for the OUTFILE parameter to be in a library that resides in the system ASP.

Top

## Parameters

Keyword	Description	Choices	Notes
OUTFILE	File to receive output	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
JOB	Job name	Single values: *, *ALL Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
TYPE	Type of records	<i>*SUMMARY, *DETAIL</i>	Optional
FRCRCD	Force record write	0-32767, <i>*CALC</i>	Optional
COMMENT	Comment	<i>Character value, *BLANK</i>	Optional

Top

---

## File to receive output (OUTFILE)

Specifies the file to which the performance statistics are to be written. If the file does not exist, it is created based on model file QAQQDBMN in library QSYS.

This is a required parameter.

### Qualifier 1: File to receive output

*name* Specify the name of the file.

### Qualifier 2: Library

**\*LIBL** All libraries in the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to be searched.

Top

---

## Output member options (OUTMBR)

Specifies the name of the database file member that receives the output of the command.

### Element 1: Member to receive output

**\*FIRST**

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter. If the member already exists, you have the option to add new records to the end of the existing member or clear the member and then add the new records.

*name* Specify the name of the file member that receives the output. If it does not exist, the system creates it.

### Element 2: Replace or add records

**\*REPLACE**

The system clears the existing member and adds the new records.

**\*ADD** The system adds the new records to the end of the existing records.

Top

---

## Job name (JOB)

Specifies the job for which the database monitor is to be started.

### Single values

**\*** The data monitor for the job running the STRDBMON command is to be started.

**\*ALL** All jobs on the system are monitored, including jobs waiting on job queues.

### Qualifier 1: Job name



*name* Specify the name of the job whose database monitor is to be started. If no job user name or job number qualifiers are specified, all of the jobs currently in the system are searched for the specified simple job name. If duplicates of the specified job name are found, you need to specify a job user name or job number that uniquely identifies the job to be changed.

#### Qualifier 2: User

*name* Specify the name of the user of the job whose database monitor is to be started.

#### Qualifier 3: Number

*000000-999999*

Specify the number of the job whose database monitor is to be started.

Top

---

## Type of records (TYPE)

Specifies the type of database records to place in the outfile.

### \*SUMMARY

Only the summary database monitor records are collected.

### \*DETAIL

Both detail and summary database monitor records are collected.

Top

---

## Force record write (FRCRCD)

Specifies the number of records to be held in the buffer before forcing the records to be written to the file.

### \*CALC

The command will calculate the number of records to be held in the buffer.

### *number-of-records*

Specify the number of records to be held. Valid values range from 0 through 32767.

Top

---

## Comment (COMMENT)

Specifies the description that is associated with the database monitor record whose ID is 3018.

### \*BLANK

Text is not specified.

### *character-value*

Specify up to 100 characters of text.

Top

---

## Examples

### Example 1: Starting Database Monitoring for All Jobs

```
STRDBMON  OUTFILE(QGPL/FILE1)  OUTMBR(MEMBER1 *ADD)
          JOB(*ALL)  FRCRCD(10)
```

This command starts database monitoring for all jobs on the system. The performance statistics are added to the member named MEMBER1 in the file named FILE1 in the QGPL library. Ten records will be held before being written to the file.

### Example 2: Starting Database Monitoring for a Specific Job

```
STRDBMON  OUTFILE(*LIBL/FILE3)  OUTMBR(MEMBER2)
          JOB(134543/QPGMR/DSP01)  FRCRCD(20)
```

This command starts database monitoring for job number 134543. The job name is DSP01 and was started by the user named QPGMR. The performance statistics are added to the member named MEMBER2 in the file named FILE3. Twenty records will be held before being written to the file.

### Example 3: Starting Database Monitoring for a Specific Job to a File in a Library in an Independent ASP

```
STRDBMON  OUTFILE(LIB41/DBMONFILE)  JOB(134543/QPGMR/DSP01)
```

This command starts database monitoring for job number 134543. The job name is DSP01 and was started by the user named QPGMR. The performance statistics are added to the member name DBMONFILE (since OUTMBR was not specified) in the file named DBMONFILE in the library named LIB41. This library may exist in more than one independent auxiliary storage pool (ASP); the library in the name space of the originator's job will always be used.

•

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1321

Job &1 user &2 job number &3 not found.

#### CPF436B

&1 can not be specified on the OUTFILE parameter.

#### CPF436C

Job &4 is already being monitored.

#### CPF436E

Job &1 user &2 job number &3 is not active.

Top

---

## Start Data Base Reader (STRDBRDR)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Database Reader (STRDBRDR) command starts a spooling reader using a database file; the reader reads a batch input stream from the database and places the jobs onto one or more job queues. This command specifies the name of the database file and member from which the input stream is read, the name of the reader, and the names of the job queue and message queue that are used.

More than one reader can be active at the same time (as determined by the spooled subsystem description). Each database reader must have a unique reader name, and the specified file or member must be available. The reader can also be held or canceled by using the Hold Reader (HLDRDR) command or End Reader (ENDRDR) command.

Because each reader runs independently of the job that started it, the user can continue doing other work on the system after he has started a reader.

**Restriction:** The specified database file either must consist of single-field records and must have an arrival sequence access path, or it must be a standard database source file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST</i>	Optional, Positional 2
JOBQ	Job queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Job queue	<i>Name, QBATCH</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MSGQ	Queue for reader messages	Single values: *REQUESTER Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Queue for reader messages	<i>Name, QSYSOPR</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
RDR	Reader	<i>Name, *FILE</i>	Optional

Top

---

### Data base file (FILE)

This is a required parameter.

Specifies the name of the database file from which the input stream is to be read. The file must be available for allocation to the spooling reader before the reader can be started.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the file.

*library-name*

Specify the library where the file is located.

*data-base-file-name*

Specify the name of the file that contains the input stream read by the reader.

Top

---

## Member (MBR)

Specifies the name of the member in the specified file that contains the input stream.

The possible values are:

**\*FIRST**

No member name is specified; the first member in the file is used.

*member-name*

Specify the name of the member that contains the input stream to be read.

Top

---

## Job queue (JOBQ)

Specifies the name of the job queue where the spooling reader will place entries. This value is used if \*RDR is specified on the **Job queue** prompt (JOBQ parameter) of the Batch Job (BCHJOB) command. (Note that the job queue for each job within this input stream can be different.)

The possible values are:

**QBATCH**

The job entry is placed on the QBATCH job queue.

*job-queue-name*

Specify the name of the job queue to be used by this reader.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the job queue.

*library-name*

Specify the library where the job queue is located.

Top

---

## Queue for reader messages (MSGQ)

Specifies the name of the message queue that will receive any messages that are created by the reader.

The possible values are:

### QSYSOPR

The messages are sent to the system operator's message queue (QSYSOPR).

### \*REQUESTER

The messages are to be sent to the workstation message queue of the workstation of the user who started this reader. If this value is used in a batch job, the message is changed to the system operator's message queue (QSYSOPR).

### *message-queue-name*

Specify the name of the message queue that you want to receive any messages.

The possible library values are:

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

### \*CURLIB

The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

### *library-name*

Specify the library where the message queue is located.

Top

---

## Reader (RDR)

Specifies the name of the spooling reader to be started. Each reader name must be unique.

The possible values are:

\*FILE The name of the reader is the same as the name of the database file that is specified on the **Data base file** prompt (FILE parameter).

### *reader-name*

Specify the name that identifies the reader to be started.

Top

---

## Examples

```
STRDBRDR FILE(QGPL/BILLING)
```

This command starts a spooled reader that reads its input from the database file named BILLING, which is in the QGPL library. The reader name is also BILLING because the RDR parameter was not specified. The first member in the BILLING file contains the input stream to be processed. The default job queue QBATCH and the system-supplied system operator's message queue QSYSOPR are used by the database reader.

Top

---

## Error messages

### \*ESCAPE Messages

- CPF1338**  
Errors occurred on SBMJOB command.
- CPF2207**  
Not authorized to use object &1 in library &3 type \*&2.
- CPF3301**  
Reader &1 already started.
- CPF3307**  
Job queue &1 in &2 not found.
- CPF3330**  
Necessary resource not available.
- CPF3362**  
Objects in QTEMP not valid for parameter values.
- CPF3363**  
Message queue &1 in library &2 not found.
- CPF3364**  
File &1 in library &2 not database file or DDM file.
- CPF9812**  
File &1 in library &2 not found.
- CPF9815**  
Member &5 file &2 in library &3 not found.

[Top](#)

---

## Start DFU (STRDFU)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Control Language (CL) command STRDFU starts the data file utility (DFU).

---

## Error messages for STRDFU

### \*ESCAPE Messages

#### DFU0005

The command failed.

#### DFU0018

The Run option is not valid.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OPTION	Option	<i>Element list</i>	Optional, Positional 1
	Element 1: DFU option	1-5, <u>*SELECT</u>	
	Element 2: Run option	1-2, <u>*NONE</u>	
DFUPGM	DFU program	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: DFU program	<i>Name</i> , <u>*PRV</u>	
	Qualifier 2: Library	<i>Name</i> , <u>*PRV</u> , *LIBL, *CURLIB	
FILE	Data base file	<i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Data base file	<i>Name</i> , <u>*PRV</u> , *SAME	
	Qualifier 2: Library	<i>Name</i> , <u>*PRV</u> , *LIBL, *CURLIB	
MBR	Member	<i>Name</i> , <u>*PRV</u> , *FIRST	Optional, Positional 4

Top

---

## Option (OPTION)

Specifies the option to use as a value for the DFU main menu.

The possible values are:

### \*SELECT

The DFU main menu appears. You can select an option from the menu.

### **first-menu-option**

Type a number between 1 and 5. Type 1 to run, 2 to create, 3 to change, or 4 to delete a DFU program. Type 5 to run a temporary DFU program.

### **second-menu-option**

If you type 1 (to run a DFU program), you can also type a second option. The second option must be 1 or 2. Type 1 to change data, or 2 to display data without changing it.

Top

---

## **DFU program (DFUPGM)**

Specifies the name of the DFU program to be run, created, changed, or deleted. You do not use this parameter if you select option 5 (to run a temporary DFU program).

The possible values are:

**\*PRV** DFU will use the program that was used in your last DFU session.

### **program-name**

Type the qualified name of the DFU program to be used.

Top

---

## **Data base file (FILE)**

Specifies the database file you want to change or display.

The possible values are:

**\*PRV** DFU will use the file that was used in your last DFU session.

### **\*SAME**

DFU will use the file that was used to define the program. You can specify \*SAME only if you select option 1 (to run a DFU program) or option 3 (to change a DFU program).

### **file-name**

Type the qualified name of the data file you want DFU to process. (If you do not specify a library name, \*LIBL is used.)

**Note:** Make sure the data-file name is different from the DFU program name.

Top

---

## **Member (MBR)**

Specifies the member in the file you want to change or display.

The possible values are:

**\*PRV** DFU will use the member that was used in your last DFU session.

### **member-name**

Type the name of the member you want DFU to process.

Top

---

## **Examples**

None

Top



---

## Error messages

### \*ESCAPE Messages

#### DFU0005

The command failed.

#### DFU0018

The Run option is not valid.

[Top](#)



---

# Start Directory Shadowing (STRDIRSHD)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Directory Shadowing (STRDIRSHD) command submits a job to start the directory shadowing environment in the system work subsystem (QSYSWRK). The system administrator can use this command to restart the directory shadowing environment if it is not already active. Only one active directory shadowing environment per system is allowed. If the directory shadowing environment is already active, a warning message is issued.

The system work subsystem (QSYSWRK) automatically starts the directory shadowing environment as a prestart job when the subsystem is started.

To ensure the job submitted with this command is successful, use the Work with Jobs (WRKJOB) command using the job number returned in the message after issuing the STRDIRSHD command.

There are no parameters for this command.

**Restriction:** You must have job control (\*JOBCTL) authority to use this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

STRDIRSHD

This command submits a job to start the directory shadowing environment in the system work subsystem QSYSWRK.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

CPF89A8

Unable to start job that controls directory shadowing.

[Top](#)



---

## Start Diskette Reader (STRDKTRDR)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Diskette Reader (STRDKTRDR) command starts a spooling reader to the specified diskette unit to read a batch input stream and place it on the appropriate job queue. This command specifies the name of the diskette unit from which the input stream is read, the volume ID and data file name of the input stream, and the names of the job queue and message queue to be used. More than one reader can be active at the same time (as determined by the spooling subsystem description).

**Restriction** Because each reader runs independently of the job that started it, the users can continue doing other work on the system after they have started a reader. This command cannot be used to read data files of diskettes that are in the save/restore format.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEV	Diskette device	<i>Name</i>	Required, Positional 1
LABEL	Diskette label	<i>Character value</i>	Required, Positional 2
VOL	Volume	Single values: *NONE Other values (up to 50 repetitions): <i>Character value</i>	Optional, Positional 4
JOBQ	Job queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Job queue	<i>Name</i> , <u>QBATCH</u>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MSGQ	Queue for reader messages	Single values: *DEV, *REQUESTER Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Queue for reader messages	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
RDR	Reader	<i>Name</i> , *DEV	Optional, Positional 3
CRTDATE	Creation date	<i>Date</i> , *NONE	Optional
CODE	Code	*EBCDIC, *ASCII	Optional

Top

---

### Diskette device (DEV)

This is a required parameter.

Specifies the name of the diskette unit that is used to read the input stream.

*diskette-device-name*

Specify the name of the diskette unit you will be using.

---

## Diskette label (LABEL)

This is a required parameter.

Specifies the data file identifier of the file that contains the input stream.

### *data-file-identifier*

Specify the name of the data file on diskette that contains the input data stream. The data file identifier cannot exceed 8 alphanumeric characters.

Top

---

## Volume (VOL)

Specifies the volume identifiers of the diskette volumes that contain the input stream to be processed. The volumes must be placed on a disk drive in the same order that the volume identifiers are specified on this parameter.

The possible values are:

### \*NONE

No volume identifier is specified. Use the volume that is placed in the diskette reader.

### *volume-identifier*

Specify the identifiers of one or more volumes in the order that they are mounted and read. Each identifier can have 6 alphanumeric characters or less. A blank is used as a separator character when multiple volumes are listed.

You can enter multiple values for this parameter. If you are on an entry display and you need additional entry fields to enter these multiple values, type a plus sign (+) in the entry field opposite the phrase "+ for more" and press Enter.

Top

---

## Job queue (JOBQ)

Specifies the name of the job queue on which the spooling reader places job entries. This value is used if \*RDR is specified on the **Job queue** prompt (JOBQ parameter) of the Batch Job (BCHJOB) command. (Note that, if \*RDR is not specified, the job queue for each job in the input stream can be different.)

**Note:** If both the user identified in the job description of the job being read and the user processing the Start Diskette Reader (STRDKTRDR) command are not authorized to the job queue on which the job should be placed, the job ends and a diagnostic message is placed in the job log. The input stream, continues to be processed, starting with the next job. If either user is authorized to the job queue, the job runs without error.

The possible values are:

### QBATCH

The job entry is placed on the QBATCH job queue. This will only occur if there is no override in the input stream.

### *job-queue-name*

Specify the name of the job queue to which each job read is sent.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the job queue.

*library-name*

Specify the library where the job queue is located.

Top

---

## Queue for reader messages (MSGQ)

Specifies the name of the message queue where messages that are created by the diskette reader will be sent.

The possible values are:

**\*DEV D**

The messages are sent to the message queue that was specified in the device description of the device named on the **Diskette device** prompt (DEV parameter).

**\*REQUESTER**

The messages are sent to the workstation message queue of the workstation of the user who started the reader. This value becomes \*DEV D for batch jobs.

*message-queue-name*

Specify the name of the message queue to which diskette reader messages will be sent.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the message queue.

*library-name*

Specify the library where the message queue is located.

Top

---

## Reader (RDR)

Specifies the name of the spooling reader that you are starting. Each reader name must be unique.

**\*DEV** The name of the reader is the same as the name of the diskette unit that is specified on the **Diskette device** prompt (DEV parameter).

*reader-name*

Specify the name of the reader that you are starting.

Top

---

## Creation date (CRTDATE)

Specifies when the diskette data file was created on diskette. The creation date should not be specified unless you want the system to check this date against the diskette date.

The possible values are:

### \*NONE

The creation date is not specified; no check is made.

### *creation-date*

Specify the creation date of the data file that is read. The date must be specified in the job date format.

Top

---

## Code (CODE)

Specifies the type of character code used to read the diskette data into the job queue.

The possible values are:

### \*EBCDIC

The diskette data file is written in the EBCDIC character code.

### \*ASCII

The diskette data file is written in the ASCII character code.

Top

---

## Examples

```
STRDKTRDR  DEV(QDKT) LABEL(OCT24) VOL(SALES)
```

This command starts the spooled reader named QDKT, which reads diskette input from the device QDKT. Because \*DEV was the default on the unspecified RDR parameter, the device name QDKT is also used as the reader name. The reader reads its input from the data file named OCT24 whose volume identifiers must be SALES. The default job queue QBATCH and the message queue QSYSOPR are used by the diskette reader.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1338

Errors occurred on SBMJOB command.

#### CPF2207

Not authorized to use object &1 in library &3 type \*&2.

#### CPF3301

Reader &1 already started.

#### CPF3307

Job queue &1 in &2 not found.

#### CPF3330

Necessary resource not available.

#### CPF3347

Device &1 not found.

#### CPF3362

Objects in QTEMP not valid for parameter values.



**CPF3363**

Message queue &1 in library &2 not found.

**CPF3367**

Device &1 not diskette device.

[Top](#)



## Start Diskette Writer (STRDKTWTR)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Start Diskette Writer (STRDKTWTR) command starts a spooling writer to the specified diskette unit. The writer takes spooled files from an output queue and produces (writes) the output on the diskette unit. This command specifies the names of the diskette unit and the writer and the names of the output and message queues to be used.

More than one writer can be active at the same time (as determined by the spooling subsystem description). The writer can also be held or stopped if the Hold Writer (HLDWTR) or End Writer (ENDWTR) command is used.

Because each writer runs independently of the job that started it, the user can continue doing other work on the system after he has started a writer.

Top

### Parameters

Keyword	Description	Choices	Notes
DEV	Diskette device	Name	Required, Positional 1
OUTQ	Output queue	Qualified object name	Required, Positional 2
	Qualifier 1: Output queue	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
MSGQ	Queue for writer messages	Single values: *REQUESTER, *DEV Other values: Qualified object name	Optional, Positional 4
	Qualifier 1: Queue for writer messages	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
WTR	Writer	Name, *DEV	Optional, Positional 3
AUTOEND	Auto-end options	Element list	Optional, Positional 5
	Element 1: Automatically end writer	*NO, *YES	
	Element 2: If yes, when to end	*NORDYF, *FILEEND	
FILE	Spooled file	Name, *NONE	Optional
JOB	Job name	Single values: * Other values: Qualified job name	Optional
	Qualifier 1: Job name	Name	
	Qualifier 2: User	Name	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, *ONLY, *LAST	Optional

Top

---

## Diskette device (DEV)

This is a required parameter.

Specifies the name of the diskette unit which will receive the spooled output.

*diskette-device-name*

Specify the name of the diskette unit to which the spooled output will be sent.

Top

---

## Output queue (OUTQ)

This is a required parameter.

Specifies the name of the output queue from which the writer processes spooled files. The output queue must be available before the writer can be started.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the output queue.

*library-name*

Specify the library where the output queue is located.

Top

---

## Queue for writer messages (MSGQ)

Specifies the name of the message queue to which messages created by the diskette writer will be sent.

The possible values are:

**\*DEV**

The messages are to be sent to the message queue that was specified in the device description of the device named on the **Diskette device** prompt (DEV parameter).

**\*REQUESTER**

The messages are to be sent to the workstation message queue of the workstation of the user who started the writer. This value is not valid for batch jobs.

*message-queue-name*

Specify the name of the message queue where the diskette writer messages will go.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the message queue.

*library-name*

Specify the library where the message queue is located.

Top

---

## Writer (WTR)

Specifies the name of the writer being started. Each writer name must be unique.

The possible values are:

**\*DEV** The name of the writer is to be the same as the name of the diskette unit that was specified on the **Diskette device** prompt (DEV parameter).

*writer-name*

Specify the name that of the writer to be started.

Top

---

## Auto-end options (AUTOEND)

Specifies whether the diskette writer should stop automatically when there are no more spooled files to be written to the diskette unit.

### Element 1: Stop Writer Option

**\*NO** The writer does not end when the last available entry has been removed from the output queue; it waits for another spooled file entry to be put on the queue.

**\*YES** The writer automatically ends after it has reached the state specified on the second part of this parameter.

### Element 2: Conditions for Stopping Writer

**\*NORDYF**

The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

**\*FILEEND**

The writer stops after it has finished processing one spooled file.

Top

---

## Spooled file (FILE)

Specifies the name of the first (or only) spooled file processed by the spooling writer and written to diskette.

The possible values are:

**\*NONE**

No spooled file name is specified; the first spooled file that becomes available on the output queue is processed first.

*spooled-file-name*

Specify the name of the spooled file that is the first (or only) file to be written to diskette.

Top

---

## Job name (JOB)

Specifies the name of the job that created the spooled file being written to diskette. This parameter is ignored when \*NONE is specified on the **Spooled file** prompt (FILE parameter).

The possible values are:

\*  
\_ The job that issued this Start Diskette Writer (STRDKTWTR) command is the job that created the spooled file.

*job-name*

Specify the name of the job that created the spooled file.

*user-name*

Specify the user name that identifies the user profile under which the job is run.

*job-number*

Specify the system-assigned job number.

Top

---

## Spooled file number (SPLNBR)

Specifies the number of the spooled file that is processed first. This parameter is valid only if a spooled file name is specified on the **Spooled file** prompt (FILE parameter).

The possible values are:

\*ONLY

Only one spooled file from the job on the specified output queue has the specified name; no spooled file number is needed.

\*LAST

The highest-numbered spooled file with the specified file name from the job on the specified output queue is the file processed first.

*spooled-file-number*

Specify the number of the specified file from the job on the specified output queue that is to be processed first.

Top

---

## Examples

```
STRDKTWTR  DEV(QDKT)  OUTQ(QDKT)  AUTOEND(*YES)
```

This command starts a spooling writer to the diskette drive. The files written on the diskettes are on the IBM-supplied output queue QDKT. When all the files have been written (no more entries are on the QDKT output queue), the writer is automatically ended and the diskette drive is available for other uses.

Top

---

## Error messages

\*ESCAPE Messages

**CPF0906**

A duplicate job named &3/&2/&1 was found.

**CPF1338**  
Errors occurred on SBMJOB command.

**CPF1764**  
Writer already started for device &1.

**CPF1842**  
Cannot access system value &1.

**CPF2207**  
Not authorized to use object &1 in library &3 type \*&2.

**CPF3303**  
File &1 not found in job &5/&4/&3.

**CPF3305**  
Output queue &1 in library &2 assigned to another writer.

**CPF3309**  
No files named &1 are active.

**CPF3310**  
Writer &1 already started.

**CPF3330**  
Necessary resource not available.

**CPF3340**  
More than one file with specified name found in job &5/&4/&3.

**CPF3342**  
Job &5/&4/&3 not found.

**CPF3343**  
Duplicate job names found.

**CPF3347**  
Device &1 not found.

**CPF3357**  
Output queue &1 in library &2 not found.

**CPF3362**  
Objects in QTEMP not valid for parameter values.

**CPF3363**  
Message queue &1 in library &2 not found.

**CPF3367**  
Device &1 not diskette device.

**CPF3418**  
Duplicate file &1 number &2 found in job.

**CPF3463**  
Output queue for device &1 not found.

**CPF3478**  
File &1 not found in job &5/&4/&3 on output queue &6 in library &7.

Top





---

## Start DNS Query (STRDNSQRY)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

Start DNS Query (STRDNSQRY), and its alias NSLOOKUP, start the NSLookup (Name Server Lookup) tool.

NSLookup is an interactive query tool that allows you to retrieve information from, or test the response of a DNS server. You can verify that a DNS server is responding correctly before you configure your system to use it. You can also retrieve DNS information about hosts, domains, and DNS servers.

**Note:** NSLookup asks for (queries) information from DNS servers. To begin a NSLookup query session, an active DNS server must be designated the 'default' server for the query session. The default server is the DNS server that NSLookup sends all queries to unless you tell it otherwise. All references in the following help to 'the default server', or 'the default DNS server', refer only to the default DNS server for the current NSLookup query session.

NSLookup retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its default DNS server for the session: 1. The DNS server your system is configured to use, or 2. The DNS server that is running on your local system.

If neither of these conditions exist, NSLookup will not be able to retrieve any information until you specify a DNS server to query. DMNNAMSVR allows you to start the query session and set the DNS server of your choice as the default server for the session.

There are two parameters for this command:

1. HOSTNAME
2. DMNNAMSVR

These parameters are used with STRDNSQRY to specify a default DNS server for the query session or, to request information about a specific host on session start up. Help for these parameters follows the list of session subcommands.

Following is a list of NSLookup subcommands that can be used once the query session is started.

### NAME

Show the IP address of the host NAME. Substitute a host name for NAME. The current or 'default' DNS server is queried.

### NAME1 NAME2

Show the IP address of the host NAME (NAME1), but query NAME2 for the information instead of the current (default) DNS server (where NAME2 is name of a DNS server).

Allows you to direct the query to a DNS server other than the current or 'default' DNS server for the query session.

### help (or ?)

Displays a list of subcommands for the STRDNSQRY (NSLOOKUP) tool.

### server NAME

Change the default (current) DNS server to NAME (where NAME is the name of a DNS server), using the current (default) DNS server.

**lserver NAME**

Change the default (current) DNS server to NAME (where NAME is the name of a DNS server), using the initial default DNS server.

Useful if you switched default DNS servers during your query session, and the current DNS server cannot resolve the new DNS server name. **lserver** allows you to make the switch using your initial default DNS server instead of the current one. If the initial DNS server also cannot resolve the new DNS name, substitute the IP address for the name, if you know it. If you do not know the IP address for the new DNS server, try restarting the NSLookup session using the DMNNAMSVR parameter to specify the new DNS server as the default server for the query session.

**root** Makes the root DNS server the default DNS server for the query session. The root DNS server is defined by the 'set root=NAME' option.

**set** The set subcommand allows you to set values for query session options. Valid option values for the set subcommand are:

**set all** Show the current values for all of the session options. If no option values have been set, the default values for each option are shown.

**set debug**  
Show debugging information.

**set nodebug**  
Do not show debugging information.

**set d2** Show exhaustive (verbose) debugging information.

**set nod2**  
Do not show exhaustive (verbose) debugging information.

**set defname**  
Append the default domain name to each query. The default domain name is defined by the 'set domain=NAME' option.

**set nodefname**  
Do not append the default domain name to each query.

**set search**  
Use the srchlist option instead of the defname option. Uses the list of domain names defined by the 'set srchlist=N1/N2/N3...' option.

**set nosearch**  
Do not use the srchlist option.

**set recurse**  
Query other DNS servers if the default server does not have the information.

**set norecurse**  
Do not query other DNS servers if the default server does not have the information.

**set vc** Use TCP for queries instead of UDP.

**set novc**  
Do not use TCP for queries instead of UDP.

**set ignoretc**  
Do not retry query using TCP if UDP reply is truncated.

**set noignoretc**  
Retry query using TCP if UDP reply is truncated.

**set domain=NAME**

Set default domain name to NAME (substitute a domain name for NAME). Defines the default domain name used by the 'set defname' option.

**set srchlist=N1/N2/N3...**

Creates a list of domain names to append to each query. Each domain name in the list is appended to the query until a reply is received, or there are no more names in the list. Substitute domain names for N1, N2, N3, etc.

**set root=NAME**

Set root server to NAME (substitute a DNS server name for NAME). Defines the server used by the 'root' subcommand.

**set retry=X**

Set the number of retries to X (where X is a numerical value).

**Note:** The default value for number of retries is 1. The retry value works together with the timeout value, which is the time in seconds that NSLookup waits before making the first retry. Retry values are usually set to 1 or 2.

**set timeout=X**

Set initial timeout interval to X seconds (where X is a numerical value).

**Note:** timeout=X determines how long NSLookup waits before making the first retry if no reply is received on the first query. The timeout value doubles after each unsuccessful retry. The default value is 5 seconds.

**set type=X**

Determines the type of DNS record that the DNS server will use to answer the query. Substitute 'X' for one of the following DNS record types:

**A** IP Address record. This is the default value.

**ANY** Any record type that exists for the subject of the query.

**CNAME**

Canonical Name record. Returns a list of aliases for the true (canonical) host name if any exist.

**HINFO**

Host information. Information about the CPU type and operating system of subject of the query.

**MX** Mail Exchange record.

**NS** Name server (DNS server) information for the zone

**PTR** Pointer record. Returns a host name for an IP address.

**SOA** Start of Authority record.

**TXT** Text record.

**WKS** Well-known services or applications available on this host.

**Note:** This type of information record is not usually available.

**set port=X**

Use TCP/IP port 'X' to query the DNS server, where 'X' is a TCP/IP port number. The default value is port 53.

**Note:** The well known port number for DNS servers is 53 and most DNS servers use it. You do not normally need to set the port value unless the DNS server you want to query

is not using port 53. Other ports are sometimes used under special circumstances. To query DNS server that is not using port 53, set the port value to the same port number the DNS server is using.

**ls** List. The list subcommand is used to display information or write it to a file. It is used with additional values to determine the kind of information displayed or written, and if written, the path and file name of the file to write the information to. Values for the ls subcommand are:

**ls DOMAIN > FILE**

Write a list of IP addresses in DOMAIN to FILE. Substitute the name of the domain for DOMAIN, and the full path and filename to write to for FILE.

```
ls company.us.com > /temp/filename.extension
```

**ls -a DOMAIN**

List all canonical (true) names and aliases in DOMAIN (substitute a domain name for DOMAIN).

**ls -h DOMAIN**

List HINFO (CPU type and operating system) for DOMAIN (substitute a domain name for DOMAIN).

**ls -s DOMAIN**

List the well-known services available on DOMAIN (substitute a domain name for DOMAIN).

**ls -d DOMAIN**

List all available records for DOMAIN (substitute a domain name for DOMAIN). Includes all DNS record types.

**ls -t TYPE DOMAIN**

List all DNS TYPE records for DOMAIN. Substitute a DNS record type for TYPE, and a domain name for DOMAIN. See the 'set type=X' subcommand for a list of DNS record types.

**view FILE**

Display the contents of ls output FILE (substitute the ls output file name for FILE).

**exit** End the query session. Then hit enter to return to the command line.

Top

---

## Parameters

Keyword	Description	Choices	Notes
HOSTNAME	Host	Character value, <u>*NONE</u>	Optional, Positional 1
DMNNAMSVR	Domain Name Server	Character value, <u>*CFG</u>	Optional

Top

Top

---

## Domain Name Server (DMNNAMSVR)

Specify the name or the IP address of the DNS server that NSLookup will use as its default server for the query session.

**Note:** NSLookup retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its default DNS server for the session: 1. The DNS server your system is configured to use, or 2. The DNS server that is running on your local system.

If neither of these conditions exist, NSLookup will not be able to retrieve any information until you specify a DNS server to query. DMNNAMSVR allows you to start the query session and set the DNS server of your choice as the default server for the session.

Use the DMNNAMSVR parameter of the STRDNSQRY command to specify a default DNS server for your NSLookup query session. You can specify any DNS server your TCP/IP network has access to. Or, if you want to test the response of a DNS server prior to designating it for use by your system, specify that server.

Valid values for the DMNNAMSVR parameter are:

**\*CFG** Use the DNS server that is currently designated for use by this system.

*domain-name-server-name*

Specify the name of a DNS server.

*domain-name-server-IP-address*

Specify the IP address of a DNS server.

Top

---

## Examples

Top

---

## Error messages

Unknown

Top



---

## Start Disk Reorganization (STRDSKRGZ)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Disk Reorganization (STRDSKRGZ) command allows the user to start the disk reorganization function for one or more auxiliary storage pools (ASPs). The user specifies a time limit that the function is to run for each ASP being reorganized. A message will be sent to the system history (QHST) log when the reorganization function is started for each ASP.

Unused space will be collected together within the ASP. This allows future large disk allocations to be done more efficiently.

**Restriction:** You must have \*ALLOBJ special authority to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
ASP	ASP number	Single values: *ALL Other values (up to 32 repetitions): 1-32	Optional, Positional 2
ASPDEV	ASP device	Values (up to 32 repetitions): <i>Name</i> , *ALLAVL	Optional
TIMLMT	Time limit	1-9999, *NOMAX	Required, Positional 1

Top

---

### Auxiliary storage pool ID (ASP)

Specifies for which auxiliary storage pools the disk reorganization function is to be started.

**\*ALL** Disk reorganization will be started for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

*auxiliary-storage-pool-number*

Specify the ASP for which disk reorganization is to be started. Valid ASP numbers are 1 to 32. Up to 32 ASP numbers may be specified.

Top

---

### ASP device (ASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device for which the disk reorganization is to be started. A value must be specified for the ASP parameter or the ASPDEV parameter.

**\*ALLAVL**

Disk reorganization will be started for all ASP devices that currently have a status of 'Available'.

### *auxiliary-storage-device-name*

Specify the name of the independent ASP device for which disk reorganization is to be started. Up to 32 ASP device names may be specified.

Top

---

## Time limit (TIMLMT)

Specifies the amount of time, in minutes, that the reorganization function is allowed to run. When the time limit is reached the function ends. The time limit specified is for each ASP being reorganized. For example, if ASP(\*ALL) is specified and the machine has four ASP's configured and TIMLMT(60) is specified, four reorganization functions are started and each can run 60 minutes. If reorganization of any ASP has not completed after 60 minutes, it will be forced to end. This allows you to do disk reorganization incrementally.

### **\*NOMAX**

There is no time limit for the reorganization function. For a large ASP that has many small unused disk areas, the reorganization function can take a long time to complete. If you start the reorganization function with \*NOMAX and you want to force the function to end, you can use the End Disk Reorganization (ENDDSKRGZ) command.

### *time-limit*

Specify the time limit that the reorganization function is allowed to run. Valid values range from 1 to 9999 minutes.

Top

---

## Examples

### **Example 1: Starting Disk Reorganization for ASP 1**

```
STRDSKRGZ ASP(1) TIMLMT(*NOMAX)
```

This command allows the user to start the disk reorganization function for ASP 1. The reorganization function will run until the ASP has been reorganized or it is ended through the ENDDSKRGZ command.

### **Example 2: Starting Disk Reorganization with a Time Limit**

```
STRDSKRGZ ASP(*ALL) TIMLMT(60)
```

This command allows the user to start the disk reorganization function for each ASP on the system. Each reorganization function will have a time limit of sixty minutes. After sixty minutes, any reorganization functions which have not completed will be ended.

### **Example 3: Starting Disk Reorganization for an ASP Device**

```
STRDSKRGZ ASPDEV(MYASP1) TIMLMT(*NOMAX)
```

This command allows the user to start the disk reorganization function for ASP device MYASP1. The reorganization function will run until the ASP has been reorganized or it is ended through the ENDDSKRGZ command.

Top

---

## Error messages

### \*ESCAPE Messages



**CPF1888**

Disk reorganization for ASP &1 already started.

**CPF1890**

\*ALLOBJ authority required for requested operation.

**CPF1891**

Disk reorganization cannot be started on an ASP with only one unit.

Top



---

## Start Education (STREDU)

### Where allowed to run:

- Interactive job (\*INTERACT)
- Interactive program (\*IPGM)
- Interactive REXX procedure (\*IREXX)

Threadsafe: No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Education (STREDU) command starts the online education session.

There are no parameters for this command.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

STREDU

This command shows the following menus:

- The **Start Education Administration** menu is shown for the Administrator.
- The **Select Course Option** menu is shown for the new student that was enrolled by the Administrator.
- The **Specify your Name** data entry screen is shown for the new student that was not enrolled.
- The **Select Course Option** menu is shown for the enrolled student.

[Top](#)

---

## Error messages

None

[Top](#)



## Start 3270 Display Emulation (STREML3270)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Start 3270 Display Emulation (STREML3270) command starts a 3270 device emulation session for a display device to a binary synchronous communications (BSC) or Systems Network Architecture (SNA) host system. The user can type this command on the command line or from any display station that allows CL commands to be specified. This command can also be issued from a batch job by specifying the display device (DSPDEV) parameter.

The STREML3270 command can be in a CL program specified as the INLPGM for a user profile that is run when the user signs on the display device.

More information on device emulation is in the 3270 Device Emulation Support book, SC41-5408.

Top

### Parameters

Keyword	Description	Choices	Notes
EMLCTL	Emulation controller, or	<i>Name</i>	Optional, Positional 1
EMLDEV	Emulation device, or	<i>Name</i>	Optional, Positional 2
EMLLOC	Emulation location	<i>Communications name</i>	Optional, Positional 3
DSPDEV	Display device, batch only	<i>Name</i> , *CURRENT	Optional
PAGEUP	Page Up (Roll Down) key	*PA2, *PA1, *PA3, *NONE, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *CLEAR, *ERASEINP, *CSRSLT	Optional
PAGEDOWN	Page Down (Roll Up) key	*PA1, *PA2, *PA3, *NONE, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24, *CLEAR, *ERASEINP, *CSRSLT	Optional
TESTREQ	Test Request key	*DFT, *CLEAR, *ERASEINP	Optional
CSRSLT	Cursor Select key	*NONE, *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24	Optional
IGCEMLPC	SNA DBCS 3270PC emulation	*NO, *YES	Optional
EMLPRTDEV	Emulation printer device	<i>Name</i> , *NONE, *EMLCTL, *EMLLOC	Optional
INZWAIT	Timeout wait for host	1-32767, 120, *NOMAX	Optional
NUMLCK	Numeric lock keyboard	*EMLDEV, *NO, *YES	Optional
NULLS	Handle nulls	*BLANK, *REMOVE	Optional
LOGON	Host signon/logon command	<i>Character value</i> , *NONE	Optional
WAITRSP	Wait response	*NO, *YES	Optional

Keyword	Description	Choices	Notes
ENDCOND	End emulation conditions	Single values: <b>*NONE</b> Other values (up to 2 repetitions): *DACTLU, *UNBIND	Optional
ATNEMLMNU	Attention emulation menu	<b>*YES</b> , *NO	Optional
FKEYPGM	Function key program	Single values: <b>*NONE</b> Other values: <i>Element list</i>	Optional
	Element 1: Program	<i>Qualified object name</i>	
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , *CURLIB	
	Element 2: Function keys	Single values: *ALLFKEYS Other values (up to 24 repetitions): *F1, *F2, *F3, *F4, *F5, *F6, *F7, *F8, *F9, *F10, *F11, *F12, *F13, *F14, *F15, *F16, *F17, *F18, *F19, *F20, *F21, *F22, *F23, *F24	
KBDTYPE	Keyboard language type	<b>*DSPDEV</b> , *SYSVAL, *LCL, AGB, AGE, AGI, AGM, ALI, ALM, BGB, BGE, BLI, BLM, BRB, BRE, CAB, CAE, CAI, CAM, CLB, CLE, CSB, CSE, CYB, DMB, DME, DMI, DMM, ESB, FAB, FAE, FAI, FAM, FNB, FNE, FNI, FNM, FQB, FQI, GKB, GNB, GNE, HIB, HNB, HNE, ICB, ICE, ICI, ICM, INB, INI, IRB, ITB, ITE, ITI, ITM, JEB, JEL, JKB, JPB, JPE, JUB, KAB, KOB, LAE, LTB, LVB, MKB, MKE, NCB, NCE, NEB, NEE, NEI, NEM, NWB, NWE, NWI, NWM, PLB, PKE, PLE, PRB, PRE, PRI, PRM, RCB, RMB, RME, ROB, ROE, RUB, RUE, SFI, SFM, SGI, SGM, SKB, SKE, SPB, SPE, SPI, SPM, SQB, SQE, SSB, SSI, SSE, SSM, SWB, SWE, SWI, SWM, TAB, THB, THE, TKB, TKE, TRB, TRE, UAE, UKB, UKE, UKI, UKM, USB, USE, USI, USM, VNE, YGI, YGM, *TRNTBL	Optional
TRNTBLOUT	Outgoing translation table	Single values: <b>*KBDTYPE</b> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Outgoing translation table	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , *CURLIB	
TRNTBLIN	Incoming translation table	Single values: <b>*KBDTYPE</b> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Incoming translation table	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , *CURLIB	
EMLCFGE	Configuration entry	<i>Name</i> , QEMDFTCFGE, <b>*NONE</b>	Optional

Top

## Emulation controller (EMLCTL)

Specifies the name of a binary synchronous communications (BSC) controller description or Systems Network Architecture (SNA) controller description that has attached 3270 emulation device descriptions. When this parameter is specified, the requesting display device is linked to an available 3270 emulation device on the emulation controller. At least one device attached to the controller must be available, and the requester of the command must be authorized to use the controller and device.

Either this parameter, the **Emulation device** prompt (EMLDEV parameter), or the **Emulation location** prompt (EMLLOC parameter) is required.

Top

---

## Emulation device (EMLDEV)

Specifies the name of a binary synchronous communications (BSC) or any System Network Architecture (SNA) device emulation (3278) that is linked to the requesting display device to emulate a 3270 display device. The requester must be authorized to this device, and the device must be available.

Either this parameter, the **Emulation controller (EMLCTL)** parameter, or the **Emulation location (EMLLOC)** parameter is required.

Top

---

## Emulation location (EMLLOC)

Specifies the remote location name that describes the location of the 3270 display emulation devices. This name is defined during device description configuration, and it refers to the remote location where communication takes place. When this parameter is specified, the requesting display device is linked to an available 3270 emulation device referred to by the remote location. At least one of the emulation devices referred to by the remote location must be available, and the requester of the command must be authorized to use the device. A remote location can refer to as many as 1,016 emulation display devices.

Either this parameter, the **Emulation controller** prompt (EMLCTL parameter), or the **Emulation device** prompt (EMLDEV parameter) is required.

Top

---

## Display device, batch only (DSPDEV)

Specifies the name of the display device used for display emulation when the command is in a batch job. The 3270 support tries to acquire the display device by this name; if the display device is acquired, the 3270 device emulation is active on that display device.

### \*CURRENT

The current display device name is used for device emulation. This parameter is used when the command is in an interactive job.

### *display-device-name*

Specify the display device name used for device emulation. This parameter is used when the command is in a batch job.

Top

---

## Page Up (Roll Down) key (PAGEUP)

Specifies a 3270 function for the Page Up (Roll Down) key on the 5250 type keyboard when 3270 device emulation is active. This assignment is in effect when the number of input fields is not larger than the maximum number of input fields.

### \*PA-key

Specify the 3270 PA key assigned to the Page Up (Roll Down) key. The default is the \*PA2 key.

### \*NONE

No function is assigned to the Page Up (Roll Down) key. When there are fewer input fields on the display than allowed by the 5250 display device, this key has no function.

**\*F-key** Specify the 3270 F key assigned to the Page Up (Roll Down) key.

**\*CLEAR**

The 3270 CLEAR key is assigned to the Page Up (Roll Down) key.

**\*ERASEINP**

The 3270 ERASE INPUT key is assigned to the Page Up (Roll Down) key.

**\*CSRSLT**

The 3270 CURSOR SELECT key is assigned to the Page Up (Roll Down) key and does not allow the real Cursor Select key to be used.

Top

---

## Page Down (Roll Up) key (PAGEDOWN)

Specifies an added function for the Page Down (Roll Up) key on the 5250 type keyboard when 3270 device emulation is active. This assignment is in effect when the number of input fields is not larger than the maximum number of input fields.

**\*PA-key**

Specify the 3270 PA key assigned to the Page Down (Roll Up) key. The default is the \*PA1 key.

**\*NONE**

No function is assigned to the Page Down (Roll Up) key. When there are fewer input fields on the display than allowed by the 5250 display device, the key has no function.

**\*F-key** Specify the 3270 F key assigned to the Page Down (Roll Up) key.

**\*CLEAR**

The 3270 CLEAR key is assigned to the Page Down (Roll Up) key.

**\*ERASEINP**

The 3270 ERASE INPUT key is assigned to the Page Down (Roll Up) key.

**\*CSRSLT**

The 3270 CURSOR SELECT key is assigned to the Page Down (Roll Up) key and does not allow the real Cursor Select key to be used.

Top

---

## Test Request key (TESTREQ)

Specifies an added function for the Test Request key on the 5250 keyboard when 3270 device emulation is active.

**\*DFT** Normal function is assigned to the Test Request key. This is the system default. The normal function depends on whether the 3270 emulation display device uses binary synchronous communications (BSC) or Systems Network Architecture (SNA) protocol. BSC defaults to a 3270 Test Request function, while SNA defaults to a 3270 System Request function.

**\*CLEAR**

The 3270 CLEAR key is assigned to the Test Request key.

**\*ERASEINP**

The 3270 ERASE INPUT key is assigned to the Test Request key.

Top



---

## Cursor Select key (CSRSLT)

Specifies one of several physical function keys to be used as the Cursor Select key. When 3270 emulation is active, the specified key can be used to select or reject selectable fields.

### \*NONE

A physical function key is not assigned to emulate the cursor select key. The real Cursor Select key is used.

**\*F-key** Specify the function key assigned to emulate the Cursor Select key. The use of the real Cursor Select key is not allowed.

Top

---

## SNA DBCS 3270PC emulation (IGCEMLPC)

Specifies whether System Network Architecture (SNA) double-byte character set (DBCS) 3270PC emulation or 3270 device emulation is used. This parameter is valid only when using a S/55 Personal Computer.

**\*NO** SNA DBCS 3270PC emulation is not used.

**\*YES** SNA DBCS 3270PC emulation is used.

Top

---

## Emulation printer device (EMLPRTDEV)

Specifies the emulation printer device that is used for SNA DBCS 3270PC emulation. The printer emulation device is selected after the device emulation is selected. This parameter is valid only when **\*YES** is specified on the **SNA DBCS 3270PC emulation** prompt (IGCEMLPC parameter).

### \*NONE

SNA DBCS 3270PC printer emulation with device emulation is not used. No printer emulation device is selected.

### *emulation-device-description-name*

Specify the printer emulation device with the selected display emulation device for SNA DBCS 3270PC emulation.

### **\*EMLCTL**

The first available printer emulation device on the specified controller on the **Emulation controller** prompt (EMLCTL parameter) is used.

### **\*EMLLOC**

The first available printer emulation device from the specified location on the **Emulation location** prompt (EMLLOC parameter) is used.

Top

---

## Timeout wait for host (INZWAIT)

Specifies the initial amount of time (in seconds) that 3270 emulation waits for the first display data from the host system. If the host system does not send the first display in this time, the emulation session is ended, and a message is returned to the requester.

**120** 3270 emulation waits 120 seconds for the first display from the host system.

#### **\*NOMAX**

There is no limit on the amount of time 3270 emulation waits for the first display from the host system. This value can be used when the user is not sure when the host system is active to this session. The request can be ended by using the system request and ending request functions.

#### ***number-of-seconds***

Specify the length of time (in seconds) that the 3270 emulation waits for the first display from the host system. Valid values range from 1 through 32767 seconds.

Top

---

## **Numeric lock keyboard (NUMLCK)**

Specifies whether numeric input fields will only allow numeric data on a 5250 keyboard.

#### **\*EMLDEV**

Numeric lock is specified in the EMLNUMLCK field of the emulation device description. You can use the DSPDEVD command to display the current EMLNUMLCK value for the emulation device. The value can be changed using the BSC commands CRTDEVBSC or CHGDEVBSC or the SNA commands CRTDEVHOST or CHGDEVHOST.

**\*NO** 3270 emulation will allow any data to be typed in the numeric input fields.

**\*YES** 3270 emulation will only allow numeric data to be typed in the numeric input fields. Numeric data that can be typed include the characters 0 through 9, and symbols " + - , . " and the blank symbol, which is the character *b* with a slash on the stem.

Top

---

## **Handle nulls (NULLS)**

Specifies how beginning and embedded nulls within the 3270 data stream sent from a 5250 display station are handled. Beginning nulls are those that occur before a character that is not null. Embedded nulls are those that occur between characters that are not null.

#### **\*BLANK**

Beginning and embedded nulls are changed to blanks within the 3270 data stream.

#### **\*REMOVE**

Beginning and embedded nulls are removed from the 3270 data stream.

Top

---

## **Host signon/logon command (LOGON)**

Specifies the sign-on text that is sent to the host system after SNA 3270 emulation is started. This text can be used to sign on to a specific host application.

This parameter is not allowed if specified for BSC 3270 emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation.

#### **\*NONE**

No text is sent to the host system after 3270 emulation is started.

#### ***host-logon-command***

Specify text that is sent to the host system after 3270 emulation is started. The text must be enclosed in apostrophes if it contains blanks or other special characters. All apostrophes within the text must be represented by two apostrophes. A maximum of 256 characters can be specified.

---

## Wait response (WAITRSP)

Specifies whether the 3270 emulation device waits until the data received is shown on the workstation display to send a positive response to the host system. The response time recorded by the AS/400 system may be longer than the time recorded by the host when the emulation device does not wait.

- \*NO** The emulation device does not wait to send a positive response. It sends the response as soon as the data is received to the workstation display.
- \*YES** The emulation device waits until the data received is shown on the workstation display to send a positive response.

---

## End emulation conditions (ENDCOND)

Specifies additional ways in which the SNA 3270 device emulation session can end.

This parameter is not allowed if specified for BSC 3270 emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation.

The possible values are:

**\*NONE**

No additional ways to end 3270 device emulation are requested.

**\*DACTLU**

The 3270 emulation session will end if it receives an SNA DACTLU from the host system. Please consider the following before selecting this end condition:

- There are certain host system applications that issue a DACTLU before starting, such as Time Sharing Option (TSO), which will end the 3270 emulation session before the desired application is accessed. This end condition should be avoided when trying to access these applications.

**\*UNBIND**

The 3270 emulation session will end if it receives an SNA UNBIND from the host system. Consider the following items before selecting this end condition:

- This end condition should be used only when you need to access one host application for the duration of the session. An UNBIND will occur while switching from one application to the next, and the 3270 session will end before accessing the second application.
- This end condition should only be used when the communication path to the host system is a simple one. A simple communication path is one that only involves accessing the AS/400 system where the Start 3270 Display Emulation (STREML3270) command is run, and accessing the host system that contains the desired application. Intermediate systems can exist along this simple path as long as they are not accessed. If intermediate systems are accessed, an UNBIND will occur while switching from one system to the next, and the 3270 display emulation session will end before accessing the desired application.
- There are certain host system applications that issue an UNBIND before starting, such as Time Sharing Option (TSO), which will end the 3270 display emulation session before the desired application is accessed. This end condition should be avoided when trying to access these applications.

---

## Attention emulation menu (ATNEMLMNU)

Specify whether you want the Select 3270 Emulation Option for SNA menu or the Select 3270 Emulation Option for BSC menu to be displayed when the Attention key is pressed.

This parameter is not allowed if specified for either SNA 3270 display station pass-through or SNA DBCS 3270PC emulation.

The possible values are:

- \*YES** The Select 3270 Emulation Option for SNA or BSC menu is displayed when you press the Attention key.
- \*NO** The Select 3270 Emulation Option for SNA or BSC menu is not displayed when you press the Attention key. The attention program (if any) currently active in the job will get control when the Attention key is pressed. You can still display the Select 3270 Emulation Option for SNA or BSC menu by pressing the System Request key sequence and then selecting the **Display 3270 emulation options menu** option from the System Request menu.

Top

---

## Function key program (FKEYPGM)

Specifies a user-exit program and one or more function keys that call the program. When a specified function key is pressed during the 3270 display emulation session and is sent to the host system, the user-exit program is called. When the user-exit program ends, control is returned to the 3270 display emulation session at the point where the function key was pressed.

This parameter is not valid if specified for either BSC 3270 display emulation, SNA 3270 display station pass-through, or SNA DBCS 3270PC emulation.

The user-exit program is called only if the function key is successfully sent to the host system. If the function key fails to be received, an error reset message appears at the bottom of the display suggesting you try again.

The AS/400 user-exit program must be coded to allow for input parameters. The following parameters are passed to the program in the specified order:

1. The function key identifier (10 characters). The identifier of the function key that was pressed. If function key 1 is pressed, the parameter value is \*F1. If function key 2 is pressed, the parameter value is \*F2, and so on, up to function key 24. The value is left-justified within the parameter.
2. The display name (10 characters). The name of the display on which the 3270 display emulation running. The value is left-justified within the parameter.
3. The cursor location (6 characters). The screen location of the cursor at the time the function key was pressed. The first three characters are the row position of the cursor location. The second three characters are the column position of the cursor location. For example, if the cursor location is row 24, column 1 when the function key is pressed, the value of the parameter is 024001. The row and column can be extracted from the variable using substring logic.

The possible **program name** values are:

- \*NONE** A user-exit program is not associated with any function key.

### *program-name*

Specify the name and library of the user-exit program that is called when one of the specified function keys is pressed. The program cannot be a system program.

The possible library values are:

**\*LIBL** All libraries in the user and system portions of the job's library list are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the program. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library to be searched.

The possible **function key** values are:

**\*ALLFKEYS**

All function keys call the specified user-exit program.

*function-key*

Specify a function key to call the user-exit program. A maximum of 24 values can be specified on this parameter.

Top

---

## Keyboard language type (KBDTYPE)

Specifies the 3-character keyboard language identifier which represents a specific full character identifier (CHRID - comprised of a character set and code page) that is used on the display station. To determine the full CHRID from the keyboard language identifier, see the CHRID Values table in the Create Device Display (CRTDEV DSP) command in the CL Reference.

This parameter does not apply when running SNA 3270 display station pass-through.

**\*DSPDEV**

If a local display device is specified for the STREML3270 display device (DSPDEV) parameter, then use the specified display's device description current KBDTYPE value. If a remote display device is specified for the STREML3270 DSPDEV parameter, then use the current QKBDTYPE system value.

**\*SYSVAL**

Use the current QKBDTYPE system value. This value is valid for both local and remote displays.

**\*LCL**

The display device that requested 3270 device emulation is a local display device. The keyboard type is determined from the display device description.

**\*TRNTBL**

Allows user-defined translation tables to be used. The character translation is defined in the translation tables specified by the **Outgoing translation table** prompt (TRNTBLOUT parameter) and the **Incoming translation table** prompt (TRNTBLIN parameter).

If a local display device is specified for the STREML3270 Display Device (DSPDEV) parameter, then use the specified display's device description current KBDTYPE value to determine the CHRID to be used. If a remote display device is specified for the STREML3270 DSPDEV parameter, then use the current QKBDTYPE system value.

*keyboard-language-identifier*

Specify the keyboard language identifier to be used.

Top

---

## Outgoing translation table (TRNTBLOUT)

Specifies the outgoing translation table that is used to translate characters sent from the host system to 3270 Emulation. If \*TRNTBL is specified on the **Keyboard language type** prompt (KBDTYPE parameter), the **Incoming translation table** prompt (TRNTBLIN parameter), must also be specified.

### \*KBDTYPE

Translation is done using the language specified on the **Keyboard language type** prompt (KBDTYPE parameter).

### *table-name*

Specify the name and library of the table that is used for outgoing translation.

The possible library values are:

\*LIBL All libraries in the user and system portions of the job's library list are searched until the first match is found.

### \*CURLIB

The current library for the job is used to locate the outgoing translation table. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library to be searched.

Top

---

## Incoming translation table (TRNTBLIN)

Specifies the incoming translation table that is used to translate characters sent from 3270 Emulation to the host system. If \*TRNTBL is specified on the **Keyboard language type** prompt (KBDTYPE parameter), the **Outgoing translation table** prompt (TRNTBLOUT parameter), must also be specified.

### \*KBDTYPE

Translation is done using the language specified on the **Keyboard language type** prompt (KBDTYPE parameter).

### *table-name*

Specify the name and library of the table used for incoming translation.

The possible library values are:

\*LIBL All libraries in the user and system portions of the job's library list are searched until the first match is found.

### \*CURLIB

The current library for the job is used to locate the incoming translation table. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library to be searched.

Top

---

## Configuration entry (EMLCFGE)

Specifies whether a configuration entry is used for this session. Configuration entries indicate 3270 emulation configuration options. Configuration entries are created with the Add Emulation Configuration Entry (ADDEMLCFGE) command.

The possible values are:

### \*NONE

No configuration entry is named and the configuration entry defaults are used.

### **QEMDFTCFGE**

The default configuration entry QEMDFTDFGE is used. This entry is shipped with configuration entry defaults, and can be updated with the Change Emulation Configuration Entry (CHGEMLCFGE) command.

### *configuration-entry-name*

Specify the name of the configuration entry to be used. If the configuration entry named does not exist in the configuration file, the configuration entry defaults are used.

Top

---

## Examples

```
STREML3270 EMLCTL(TSOHOST) PAGEUP(*F7) PAGEDOWN(*F8)
```

This command places the display device into an emulation session that uses the first available device on the controller description TSOHOST for which the user has authority. When there are fewer input fields on the display than the maximum allowed by the 5250 display device and the Page Up key is pressed, an F7 key value is sent to the host system. When the Page Down key is pressed, an F8 key value is sent to the host system.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF2619**

Table &1 not found.

#### **CPF269A**

Library parameter is not set to "QSYS " on call.

#### **CPF269B**

T.61 conversion table not found.

#### **CPF85EA**

Screen address received from host is larger than screen size.

#### **CPF85EB**

3270 device emulation session ended.

#### **CPF85EC**

Specifying text on the LOGON parameter is not supported.

#### **CPF85ED**

Values other than ENDCOND(\*NONE) are not supported.

#### **CPF85E2**

3270 display emulation is already active at this job.

**CPF85E4**  
Not authorized to translation table &1 in library &2.

**CPF85E5**  
3270 emulation device &1 is reserved for device &2.

**CPF85E6**  
Translation table &1 in library &2 was not found.

**CPF8503**  
Emulation controller &1 not found.

**CPF8504**  
Controller &1 does not support 3270 emulation.

**CPF8505**  
Emulation device &2 not found.

**CPF8506**  
Emulation location &1 not found.

**CPF8507**  
Display emulation cannot open required file.

**CPF8508**  
Host system did not respond.

**CPF851A**  
Maximum number of shift in and shift out characters exceeded.

**CPF8510**  
Internal error occurred on device &1.

**CPF8511**  
Emulation ended by errors on device &2.

**CPF8512**  
Emulation ended because device &2 was held.

**CPF8513**  
Emulation ended by errors on device &2.

**CPF8514**  
Error recovery stopped on device &1.

**CPF8515**  
3270 emulation session ended by host.

**CPF8516**  
No match between host and device &2.

**CPF8517**  
Received more than maximum number of fields allowed.

**CPF8518**  
Emulation ended because of internal failure in system.

**CPF8519**  
Function check in 3270 emulation.

**CPF852A**  
Values other than FKEYPGM(\*NONE) are not supported.

**CPF852B**  
Program &1 not found.



**CPF852C**  
Not authorized to program &1.

**CPF8521**  
Not authorized to controller &1.

**CPF8522**  
Not authorized to emulation device &2.

**CPF8523**  
\*NO on the ATNEMLMNU parameter is not supported.

**CPF8524**  
Emulation cannot open its required display file.

**CPF8525**  
KBDTYPE(\*LCL) not allowed for remote display devices.

**CPF8526**  
No 3270 display emulation devices available.

**CPF8527**  
Emulation device &2 not available.

**CPF8528**  
Device &2 is not a display emulation device.

**CPF8530**  
Not authorized to use any display emulation device.

**CPF8533**  
Display device not specified in a batch job.

**CPF8534**  
Display device &1 is not available.

**CPF8535**  
Display device &1 not found.

**CPF8536**  
Not authorized to display device &1.

**CPF8539**  
&1 keyboard type not supported by 3270 emulation.

**CPF8546**  
No 3270 display emulation devices available.

**CPF8547**  
No 3270 printer emulation devices available.

**CPF8550**  
Emulation ended due to time-out internal failure.

**CPF8551**  
Emulation ended with error code &1.

**CPF8552**  
Emulation ended because of return code.

**CPF8553**  
BSC controller or device not allowed without translation.

**CPF8565**  
Emulation device &1 not found.

**CPF8568**

Device &1 not printer emulation device.

**CPF8569**

Not authorized to use any printer emulation device.

**CPF8571**

No 3270 printer emulation devices available.

**CPF8572**

Emulation device &1 not available.

[Top](#)

---

## Start EPM Environment (STREPMENV)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start EPM Environment (STREPMENV) command is used with extended program model (EPM) languages to create a user-controlled environment. The C/400\*, FORTRAN/400\*, and AS/400\* Pascal languages are part of the extended program model.

You can use this command to create a run-time environment for an EPM language entry point that you are calling from another AS/400 language program. See the Extended Program Model User's Guide and Reference for more detailed information on the EPM and this command.

---

### Error messages for STREPMENV

None

Top

---

### Parameters

Keyword	Description	Choices	Notes
EPMENV	Environment Name	<i>Character value</i>	Required, Positional 1
ROOTPGM	Environment Program	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Environment Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

### Environment Name (EPMENV)

Specifies the name of the user-controlled environment that is to be created. The environment name must be unique.

*environment-name*

Enter a name for the environment.

Top

---

### Environment Program (ROOTPGM)

Specifies the name of the program and library that contains the environment definition information that is necessary in order to create the EPM run-time environment.

*program-name*

Enter the name of the EPM language program object that contains the environment definition information.

The possible values for library are:

**\*LIBL** The library list is searched to locate the specified program object.

**\*CURLIB**

The current library is searched to locate the specified program object.

*library-name*

Enter the name of the library that contains the specified program object.

Top

---

## Examples

None

Top

---

## Error messages

None

Top

---

## Start Font Management Aid (STRFMA)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Font Management Aid (STRFMA) command displays the Font Management Aid (FMA) menu. From this menu, you can "Work With" user defined characters (24 X 24 dot matrix) in the DBCS font table with a workstation font file (\$SYS1Z24.FNT). "Work With" means to copy user defined characters in the workstation font file to DBCS font table, or copy user defined characters in DBCS font table to workstation font file. FMA is also used to get a copy of the workstation user-font/dictionary file from the other workstation.

There are no parameters for this command.

---

### Error messages for STRFMA

None

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

None

[Top](#)

---

### Error messages

None

[Top](#)



---

## Start Host Server (STRHOSTSVR)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Host Server (STRHOSTSVR) command is used to start the optimized host server daemons and the server mapper daemon.

There is one server daemon for each of the host server types. In addition, there is one server mapper daemon for all host servers which provides support for client applications to obtain a particular host server daemon's port number. This port number is then used by the client application to connect to the host server's daemon. The daemon accepts the incoming connection request and routes it to the server job for further processing.

The daemons are batch jobs submitted to either the QSYSWRK or QSERVER subsystem, depending on the value or values specified for the SERVER keyword. All daemon jobs are submitted to the QSYSWRK subsystem with the exception of the \*DATABASE and \*FILE server daemons which are submitted to the QSERVER subsystem.

In order for the server daemons and the server mapper daemon to start successfully, the QSYSWRK subsystem and, for \*DATABASE and \*FILE server, the QSERVER subsystem must be active. If the required subsystem is not active, then the submission of the daemon job will fail. Additionally, the QUSRWRK subsystem or the user-defined subsystem must be active in order to start the associated server jobs. All associated server jobs can run in the QUSRWRK subsystem or a user-defined subsystem, except for the following:

- Server jobs QPWFERSVSO and QPWFERSVSS - these jobs run in the QSERVER subsystem or a user-defined subsystem
- Server job QPWFERSVS2 - this job runs in the QSERVER subsystem
- Server job QIWVPPJT - this job runs in the QSYSWRK subsystem
- Server job QTFPJTCP - this job runs in the QSERVER subsystem.

There are no server jobs associated with the server mapper daemon.

The server daemons must be active in order to allow client applications to establish a connection with the host server using sockets communication support. Once started, the server daemons and the server mapper daemon remain active until they are ended explicitly using the End Host Server (ENDHOSTSVR) command or an error occurs.

### Restrictions:

- This command is used only for enabling client applications to communicate with any of the host servers using sockets communication support. This command does not start any of the APPC host servers; these are started as a result of a program start request.
- Only one server daemon can be active for a specific server type. Requests to start a server daemon that is already active will result in an informational message to the user issuing this command.

---

## Error messages for STRHOSTSVR

### \*ESCAPE Messages

#### PWS300D

Unable to start host server daemon jobs.

---

## Parameters

Keyword	Description	Choices	Notes
SERVER	Server type	Single values: *ALL Other values (up to 8 repetitions): *CENTRAL, *DATABASE, *DTAQ, *FILE, *NETPRT, *RMTCMD, *SIGNON, *SVRMAP	Required, Positional 1
RQDPCL	Required protocol	*ANY, *NONE, *TCP	Optional

---

## Server type (SERVER)

Specifies the host server daemons to be started by this command.

The possible values are:

**\*ALL** All of the server daemons and the server mapper daemon are started.

**\*CENTRAL**

The central server daemon is started in the QSYSWRK subsystem. The daemon job is named QZSCSRVSD. The associated server job is named QZSCSRVS.

**\*DATABASE**

The database server daemon is started in the QSERVER subsystem. The daemon job is named QZDASRVSD. The associated server jobs are named QZDASOINIT, QZDASSINIT, and QTFPJTCP.

**\*DTAQ**

The data queue server daemon is started in the QSYSWRK subsystem. The daemon job is named QZHQSRVD. The associated server job is named QZHQSSRV.

**\*FILE**

The file server daemon is started in the QSERVER subsystem. The daemon job is named QPWFSERVSD. The associated server jobs are named QPWFSERVSO, QPWFSERVSS, and QPWFSERVS2.

**\*NETPRT**

The network print server daemon is started in the QSYSWRK subsystem. The daemon job is named QNPSEVRD. The associated server jobs are named QNPSEVRVS and QIWVPPJT.

**\*RMTCMD**

The remote command and distributed program call server daemon is started in the QSYSWRK subsystem. The daemon job is named QZRCSRVD. The associated server job is named QZRCSRVS.

**\*SIGNON**

The signon server daemon is started in the QSYSWRK subsystem. The daemon job is named QZSOSGND. The associated server job is named QZSOSIGN.

**\*SVRMAP**

The server mapper daemon is started in the QSYSWRK subsystem. The daemon job is named QZSOSMAPD.



---

## Required protocol (RQDPCL)

Specifies which communication protocols are required to be active for the host server daemons to start.

### Single Values

**\*ANY:** The TCP/IP communication protocol must be active at the time the STRHOSTSVR command is issued. If TCP/IP is not active, escape message PWS300D will be issued and the host server daemons will not be started. A diagnostic message (PWS3008) will be issued if TCP/IP is found to be inactive.

**\*NONE:** No communication protocols need to be active at the time the STRHOSTSVR command is issued for the host server daemons to start. No messages will be issued for protocols which are inactive.

**\*TCP:** The TCP/IP communication protocol must be active at the time the STRHOSTSVR command is issued. If TCP/IP is not active, diagnostic message PWS3008 and escape message PWS300D will be issued and the host server daemons will not be started.

Top

---

## Examples

None

Top

---

## Error messages

### \*ESCAPE Messages

#### PWS300D

Unable to start host server daemon jobs.

#### PWS3006

Errors occurred starting server daemon jobs.

Top



---

## Start IDDU (STRIDD)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Interactive Data Definition Utility (STRIDD) command runs the main Interactive Data Definition (IDDU) menu. From this menu, you can select from options that allow you to work with data definitions, data dictionaries, files, and libraries, or use related commands and office tasks.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

STRIDD

This commands displays the main IDDU menu.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Start IP over SNA Interface (STRIPSIFC)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start IP over SNA Interface (STRIPSIFC) command is used to start an AF\_INET sockets over SNA interface (an IP address by which this local host is known on the SNA transport).

**Restriction:** Only eight (8) AF\_INET sockets over SNA interfaces can be active on a single host. If the maximum number of interfaces is active and you want to start another interface, you must first end one or more interfaces using the End IP over SNA Interfaces (ENDIPSIFC) CL command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	<i>Character value</i>	Required, Positional 1

Top

---

### Internet address (INTNETADR)

Specifies the internet address of an inactive (ended) interface that had previously been added to the IP over SNA configuration with the ADDIPSIFC CL command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

This is a required parameter.

Top

---

### Examples

```
STRIPSIFC INTNETADR('9.5.1.248')
```

This command activates (starts) the interface with IP address 9.5.1.248.

Top

---

### Error messages

#### \*ESCAPE Messages

##### CPFA10F

IP over SNA interface &1 not started.

Top



---

## Start ITF (STRITF)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Interactive Terminal Facility (STRITF) command allows the user to send and receive data and file members for 5250 work stations using asynchronous communications. You can also send documents using the Interactive Terminal Facility (ITF). Before you can use ITF, you must start asynchronous communications.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	<i>Communications name</i>	Required, Positional 1

[Top](#)

---

### Remote location (RMTLOCNAME)

Specifies the name of the remote location with which you want to communicate. This name is the same as the remote location name specified during configuration.

This is a required parameter.

[Top](#)

---

### Examples

```
STRITF CHICAGO
```

This command allows the user to communicate with the remote location CHICAGO.

[Top](#)

---

### Error messages

None

[Top](#)





---

## Start Journal (STRJRN)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Start Journal (STRJRN) command is used to start journaling changes (made to an object or list of objects) to a specific journal. The object types which are supported through this interface are Data Areas (\*DTAARA), Data Queues (\*DTAQ), Byte Stream Files (\*STMF), Directories (\*DIR), and Symbolic Links (\*SYMLNK). Only objects of type \*STMF, \*DIR or \*SYMLNK that are in the root ('/'), QOpensys, and user-defined file systems are supported. For more information about the possible journal entries which can be sent, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

The user can specify that only the after image or both the before and the after images of an object of type \*DTAARA be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command.

After journaling begins for the object, the user should save the journaled object to preserve its journal attribute information. Also, the object must be saved because, for example, journaled changes cannot be applied to a version of the object that was saved before journaling was in effect.

For other ways to start journaling see the following commands:

- Access Paths - Start Journal Access Path (STRJRNAP)
- Physical Files - Start Journal Physical File (STRJRNPF)
- Other Objects - Start Journal Object (STRJRNOBJ)

### Restrictions:

- The object must not be journaling changes to another journal.
- The maximum number of objects that can be associated with one journal is 250,000. This maximum includes objects whose changes are currently being journaled, objects for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.
- The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
- Byte stream files that are currently memory mapped or byte stream files that are being used as IXS network storage spaces cannot be journaled.
- Objects that are internally marked as not eligible for journaling cannot be journaled. The system may mark system working directories that are created inside of user directories as not eligible for journaling.
- For data areas, only local external data area objects may be journaled. The special data areas (\*LDA, \*GDA, \*PDA) and DDM data areas cannot be journaled.
- For data queues, only local data queues are supported. DDM data queues cannot be journaled.

- At least one of parameter OBJ or OBJFID must be specified.

Top

## Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Name	<i>Path name</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
OBJFID	File identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
JRN	Journal	<i>Path name</i>	Optional
SUBTREE	Directory subtree	*ALL, <u>*NONE</u>	Optional
PATTERN	Name pattern	Values (up to 20 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value, *</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
INHERIT	New objects inherit journaling	<u>*NO</u> , *YES	Optional
IMAGES	Images	<u>*AFTER</u> , *BOTH	Optional
OMTJRNE	Omit journal entry	<u>*NONE</u> , *OPNCLOSYN	Optional

Top

## Objects (OBJ)

Specifies a maximum of 300 objects for which changes are to be journaled. Only objects whose path name identifies an object of type \*STMF, \*DIR, \*SYMLNK, \*DTAARA or \*DTAQ are supported.

### Element 1: Name

*'object-path-name'*

Specify the path name of the object for which changes are to be journaled.

A pattern can be specified in the last part of the path name. An asterisk (\*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed. If the path name begins with the tilde character, then the path is assumed to be relative to the appropriate home directory.

Additional information about path name patterns is in the Integrated file system information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

### Element 2: Include or omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

#### \*INCLUDE

The objects that match the object name pattern are to be journaled, unless overridden by an \*OMIT specification.

### **\*OMIT**

The objects that match the object name pattern are not be journaled. This overrides an \*INCLUDE specification and is intended to be used to omit a subset of a previously selected path.

Top

---

## **File identifier (OBJFID)**

Specifies a maximum of 300 file identifiers (FID) for which changes are to be journaled. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type \*STMF, \*DIR, \*SYMLNK, \*DTAARA or \*DTAQ are supported.

### *file-identifier*

Objects identified with the FID are journaled.

Top

---

## **Journal (JRN)**

Specifies the path name of the journal that receives the journaled changes.

### *'journal-path-name'*

Specify the path name of the journal that receives the journaled changes.

Top

---

## **Directory subtree (SUBTREE)**

Specifies whether the directory subtrees are included in the start journal operation.

**Note:** This parameter is ignored if the OBJ parameter is not specified.

**Note:** This parameter is ignored unless object-path-name is a directory (\*DIR) object.

### **\*NONE**

Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

**\*ALL** All objects that meet the selection criteria are processed in addition to the entire subtree of each directory that matches the selection criteria. The subtree includes all sub-directories and the objects within those sub-directories.

Top

---

## **Name pattern (PATTERN)**

Specifies a maximum of 20 patterns to be used to include or omit objects for the start journal operation. If this parameter is not specified, the default will be to match all patterns.

Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern. An asterisk (\*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes. Symbolic links within the path name will not be followed.

If this parameter is not specified, the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

**Note:** This parameter is ignored if the OBJ parameter is not specified.

#### Element 1: Pattern

'\*' All objects that match the input OBJ parameter are to be included into the start journal operation or omitted from the start journal operation.

#### *name-pattern*

Specify the pattern to either include or omit objects for the start journal operation.

#### Element 2: Include or omit

The second element specifies whether names that match the pattern should be included or omitted from the operation. Note that in determining whether a name matches a pattern, relative name patterns are always treated as relative to the current working directory.

**Note:** The SUBTREE parameter specifies whether the subtrees are included or omitted.

#### \*INCLUDE

The objects that match the object name pattern are included into the start journal operation unless overridden by an \*OMIT specification.

#### \*OMIT

The objects that match the object name pattern are not to be included into the start journal operation. This overrides an \*INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

---

## New objects inherit journaling (INHERIT)

Specifies whether new objects created within a journaled directory should inherit the journal options and the journal state of its parent directory.

\*NO New objects created within the directory will not inherit the journal options and journal state of the parent directory.

\*YES New objects created within the directory will inherit the journal options and journal state of the parent directory.

Top

---

## Images (IMAGES)

Specifies the kinds of images that are written to the journal receiver for changes to objects.

#### \*AFTER

Only *after* images are generated for changes to objects.

#### \*BOTH

The system generates both *before* and *after* images for changes to objects.

**Note:** The value \*BOTH is only valid for \*DTAARA objects.

Top

---

## Omit journal entry (OMTJRNE)

Specifies the journal entries that are omitted.

### \*NONE

No entries are omitted.

### \*OPNCLOSYN

Open, close and force entries are omitted. Open, close and force operations on the specified objects do not generate open, close and force journal entries. This prevents the use of TOJOB0 and TOJOB1 entries on the Apply Journalized Changes (APYJRNCHG) command, but it saves some storage space in the journal receivers.

**Note:** The value \*OPNCLOSYN is only valid for \*DIR and \*STMF objects.

Top

---

## Examples

### Example 1: Start Journaling with Omit of Directory

```
STRJRN  OBJ('/mypath' *INCLUDE)
         ('/mypath/myobject' *OMIT))
        JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
```

This command journals all changes to all objects supported by this command within the first-level of directory '/mypath' except '/mypath/myobject' to journal '/QSYS.LIB/MYLIB.LIB/JRNLA.JRN'. None of the objects within the subdirectories of '/mypath' will be journaled.

Only the *after* images of updated records are written to the journal.

### Example 2: Start Journaling with Pattern Matching

```
STRJRN  OBJ('/mypath' *INCLUDE)
         ('/mypath/myobject.txt' *OMIT))
        JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN') SUBTREE(*ALL)
        PATTERN('*.*.TXT' *INCLUDE)) OMTJRNE(*OPNCLOSYN)
```

This command journals changes to all objects that match pattern '\*.txt' in directory '/mypath' except object '/mypath/myobject.txt'. The open, close and force entries are not journaled.

Only the *after* images of updated records are written to the journal.

### Example 3: Start Journaling with Omit by Pattern

```
STRJRN  OBJ('/mypath/my*' *INCLUDE))
        JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
        PATTERN('*.*.DTA*' *OMIT))
```

This command journals changes to all objects within the first-level directories that match the pattern for path '/mypath/my\*' and will omit all objects that match pattern '\*.DTA\*' (objects of type \*DTAARA and \*DTAQ).

Only the *after* images of updated records are written to the journal.

### Example 4: Start Journaling using File Identifiers

```
STRJRN  OBJFID(00000000000000007E09BDB000000009
              00000000000000009E09BDB00000000A)
        JRN('/QSYS.LIB/MYLIB.LIB/JRNLA.JRN')
```

This command journals all changes to the objects represented by the specified file identifiers to journal '/QSYS.LIB/MYLIB.LIB/JRNLA.JRN'.

Only the *after* images of updated records are written to the journal.

#### Example 5: Start Journaling on a Set of Data Queues

```
STRJRN  OBJ('/QSYS.LIB/MYLIB.LIB/MYDATA*.DTAQ')  
        JRN('/QSYS.LIB/MYLIB.LIB/MYJRN.JRN')
```

This command starts the journaling of all changes to the objects of type \*DTAQ in library MYLIB that begin with the characters 'MYDATA'.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA0D4

File system error occurred. Error number &1.

#### CPF700A

&1 of &2 objects have started journaling.

#### CPF705A

Operation failed due to remote journal.

#### CPF9801

Object &2 in library &3 not found.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9803

Cannot allocate object &2 in library &3.

#### CPF9810

Library &1 not found.

#### CPF9820

Not authorized to use library &1.

#### CPF9825

Not authorized to device &1.

#### CPF9830

Cannot assign library &1.

#### CPF9873

ASP status is preventing access to object.

#### CPF9875

Resources exceeded on ASP &1.

Top

---

## Start Journal Access Path (STRJRNAP)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Start Journal Access Path (STRJRNAP) command is used to start journaling the access paths for all members of a database file to a specified journal. Any new member that is later added to the file also has its access path journaled.

If a physical file is specified, journaling can be started for its access paths. When access path journaling is started for a physical file, only the access paths for the physical file members are journaled. Journaling for any logical file access paths is started only when access path journaling is started for the logical file.

The journal entries created after running this command cannot be used in any apply or remove journaled changes operation. These entries are used only to recover the access path without rebuilding it after an abnormal system operation ending.

If you start journaling your access paths, consider specifying RCVSIZOPT(\*RMVINTENT) on either the Create Journal (CRTJRN) or the Change Journal (CHGJRN) command for this journal. This will reduce the additional storage required to do access path journaling.

If you do not want the overhead of managing the access path journaling yourself, consider taking advantage of the system-managed access-path protection support. For more information, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>, and the Edit Recovery for Access Paths (EDTRCYAP) or the Change Recovery for Access Paths (CHGRCYAP) command.

For other ways to start journaling see the following commands:

- Integrated file system objects - Start Journal (STRJRN)
- Physical files - Start Journal Physical File (STRJRNPF)
- Other objects - Start Journal Object (STRJRNOBJ)

### Restrictions:

- Before journaling an access path, all physical files over which the access path is built must first be journaled to the same journal that is used to journal the access path. Even if all physical file members for a particular physical file are removed from the access path of a logical file, all physical files must still be journaled to the same journal before journaling the access path.
- All access paths to be journaled must specify MAINT(\*IMMED) or MAINT(\*DLY).
- If only after images are being journaled for the physical file members, the system automatically starts journaling the before and after images for the physical file once journaling is started for any access path built over the physical file. When journaling ends for the access paths, the system automatically stops journaling the before images for the physical file and again only journals the after images.
- Overrides are not applied to files specified on the FILE parameter.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

---

## Parameters

Keyword	Description	Choices	Notes
FILE	Journalled file	Values (up to 50 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Journalled file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
JRN	Journal	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

### Journalled file (FILE)

Specifies a maximum of 50 database files whose access paths are journalled.

This is a required parameter.

#### Qualifier 1: Journalled file

*file-name*

Specify the name of the file.

#### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the name of the library to be searched.

Top

---

### Journal (JRN)

Specifies the journal that receives the file change journal entries.

This is a required parameter.

#### Qualifier 1: Journal

*journal-name*

Specify the name of the journal.

#### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.



### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

### ***library-name***

Specify the name of the library to be searched.

Top

---

## **Examples**

```
STRJR NAP FILE(MYFILE) JRN(MYLIB/JRNLA)
```

This command journals all access paths for all members in file MYFILE (found using the library search list) to journal JRNLA in library MYLIB.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF6971**

Damage prevents object &1 from being journaled.

#### **CPF6972**

Cannot allocate access path for file &1 in &2.

#### **CPF7003**

Entry not journaled to journal &1. Reason code &3.

#### **CPF7004**

Maximum number of objects journaled to journal &1.

#### **CPF7008**

Cannot start or end access path journaling for file &1.

#### **CPF7009**

Not all based-on files being journaled to &3.

#### **CPF7011**

Not enough storage or resources.

#### **CPF703C**

DDL transaction prevents journaling operation.

#### **CPF703D**

DDL transaction prevents journaling operation.

#### **CPF703E**

DDL transaction prevents journaling operation.

#### **CPF7030**

Object of type \*&3 already being journaled.

#### **CPF7031**

Cannot allocate member &3 file &1 in &2.

#### **CPF7033**

Start or end journaling failed for member &3.

#### **CPF7034**

Logical damage of file &1 in &2.

**CPF7035**  
Object &1 in &2 already known to journal.

**CPF705A**  
Operation failed due to remote journal.

**CPF7079**  
Access path journaling for file &1 not started.

**CPF708D**  
Journal receiver found logically damaged.

**CPF7084**  
Object of type \*&6 could not be journaled.

**CPF709D**  
Cannot start journaling object of type \*&7.

**CPF9801**  
Object &2 in library &3 not found.

**CPF9802**  
Not authorized to object &2 in &3.

**CPF9803**  
Cannot allocate object &2 in library &3.

**CPF9812**  
File &1 in library &2 not found.

**CPF9820**  
Not authorized to use library &1.

**CPF9822**  
Not authorized to file &1 in library &2.

**CPF9825**  
Not authorized to device &1.

**CPF9830**  
Cannot assign library &1.

**CPF9873**  
ASP status is preventing access to object.

**CPF9875**  
Resources exceeded on ASP &1.

Top

---

## Start Journal Object (STRJRNOBJ)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Start Journal Object (STRJRNOBJ) command is used to start journaling changes (made to an object or list of objects) to a specific journal. The object types which are supported through this interface are Data Areas (\*DTAARA) and Data Queues (\*DTAQ). For more information about journal entries which can be sent, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.

Additionally, the user can specify that only the after image or both the before and the after images of an object of type \*DTAARA be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command.

After journaling begins for the object, the user should save the journaled object to preserve its journal attribute information. Also, the object must be saved because, for example, journaled changes cannot be applied to a version of the object that was saved before journaling was in effect.

For other ways to start journaling see the following commands:

- Access paths - Start Journal Access Path (STRJRNAP)
- Integrated file system objects - Start Journal (STRJRN)
- Physical files - Start Journal Physical File (STRJRNPF)

### Restrictions:

- The object must not be journaling changes to another journal.
- The maximum number of objects that can be associated with one journal is 250,000. This maximum includes objects whose changes are currently being journaled, objects for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>.
- The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
- For data areas, only local external data area objects may be journaled. The special data areas (\*LDA, \*GDA, and \*PDA) and DDM data areas cannot be journaled.
- For data queues, only local data queues are supported. DDM data queues cannot be journaled.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OBJ	Object	Values (up to 300 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OBJTYPE	Object type	*DTAARA, *DTAQ	Required, Positional 2
JRN	Journal	<i>Qualified object name</i>	Required, Positional 3
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
IMAGES	Images	*AFTER, *BOTH	Optional

Top

---

### Object (OBJ)

Specifies a maximum of 300 qualified object names for which changes are to be journaled.

This is a required parameter.

#### Qualifier 1: Object

##### *object-name*

Specify the name of the object for which changes are to be journaled.

#### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

##### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

##### *library-name*

Specify the name of the library to be searched.

Top

---

### Object type (OBJTYPE)

Specifies the object type for which changes are to be journaled.

This is a required parameter.

##### **\*DTAARA**

Changes for data area objects are to be journaled.

##### **\*DTAQ**

Changes for data queue objects are to be journaled.

Top

---

## Journal (JRN)

Specifies the journal that receives the journaled changes.

This is a required parameter.

### Qualifier 1: Journal

*journal-name*

Specify the name of the journal that receives the journaled changes.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the name of the library to be searched.

Top

---

## Images (IMAGES)

Specifies the kinds of images that are written to the journal receiver for changes to objects.

**\*AFTER**

Only *after* images are generated for changes to objects.

**\*BOTH**

The system generates both *before* and *after* images to the journal for changes to objects.

**Note:** The value **\*BOTH** is only valid for **\*DTAARA** objects.

Top

---

## Examples

```
STRJRNOBJ OBJ(DTALIB/MYDTAARA) OBJTYPE(*DTAARA)
          JRN(MYLIB/JRNLA)
```

This command journals all changes to data area MYDTAARA in library DTALIB to journal JRNLA in library MYLIB. Only the *after* images of updates are written to the journal.

Top

---

## Error messages

**\*ESCAPE Messages**

**CPF6979**

Journal &1 in library &2 is unusable.

**CPF700A**

&1 of &2 objects have started journaling.

**CPF705A**

Operation failed due to remote journal.

- CPF9801**  
Object &2 in library &3 not found.
- CPF9802**  
Not authorized to object &2 in &3.
- CPF9803**  
Cannot allocate object &2 in library &3.
- CPF9810**  
Library &1 not found.
- CPF9820**  
Not authorized to use library &1.
- CPF9825**  
Not authorized to device &1.
- CPF9830**  
Cannot assign library &1.
- CPF9873**  
ASP status is preventing access to object.
- CPF9875**  
Resources exceeded on ASP &1.

Top

---

## Start Journal Physical File (STRJRNPf)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
Examples  
Error messages

The Start Journal Physical File (STRJRNPf) command is used to start journaling changes made to a specific database physical file to a specific journal. Changes in new members added to the file are also journaled.

The user can specify that only the after image or both the before and after images of records in the journaled physical file be journaled. Before images are necessary to remove journaled changes using the Remove Journaled Changes (RMVJRNCHG) command. In addition, the system will automatically capture the before images for a database file if the file is opened under commitment control. For more information about commitment control, see the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

After journaling begins for the file, and after any new members are added to the file, the user should run the Save Changed Object (SAVCHGOBJ) command with OBJTYPE(\*FILE) and OBJJRN(\*YES) specified. The file must be saved because journaled changes cannot be applied to a version of the file that was saved before journaling was in effect.

When the file being journaled is a distributed file, the STRJRNPf command is also distributed if journaling was successfully started locally. Even if the distribution request fails, the local file remains journaled.

For other ways to start journaling see the following commands:

- Access paths - Start Journal Access Path (STRJRNAP)
- Integrated file system objects - Start Journal (STRJRN)
- Other objects - Start Journal Object (STRJRNOBJ)

### Restrictions:

- The file must not be journaling changes to another journal.
- Overrides are not applied to files specified on the FILE parameter.
- The maximum number of objects that can be associated with one journal is 250,000. This maximum includes physical file members whose changes are currently being journaled, members for which journaling was ended while the current receiver was attached, and journal receivers that are or were associated with the journal while the current journal receiver is attached. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal management topic.
- In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type \*SNA.
- If the file has Large Object (LOB) columns, and the total of the LOB columns plus the record size is greater than 15,761,440 bytes, then the file can only be journaled to a journal with RCVSIZOPT(\*MAXOPT2) or RCVSIZOPT(\*MAXOPT3) specified.

---

## Parameters

Keyword	Description	Choices	Notes
FILE	Physical file to be journaled	Values (up to 50 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Physical file to be journaled	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
JRN	Journal	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
IMAGES	Record images	*AFTER, *BOTH	Optional
OMTJRNE	Journal entries to be omitted	*NONE, *OPNCLO	Optional

---

### Physical file to be journaled (FILE)

Specifies a maximum of 50 physical files whose changes are written to the journal.

This is a required parameter.

#### Qualifier 1: Physical file to be journaled

*file-name*

Specify the name of a physical file.

#### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

\*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the name of the library to be searched.

---

### Journal (JRN)

Specifies the journal that will receive the file change journal entries.

This is a required parameter.

#### Qualifier 1: Journal

*journal-name*

Specify the name of the journal.

#### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.



### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, QGPL is used.

### ***library-name***

Specify the name of the library to be searched.

Top

---

## **Record images (IMAGES)**

Specifies the kinds of record images to be written to the journal for changes to records in the file.

### **\*AFTER**

Only *after* images are written to the journal for changes to records in this file.

### **\*BOTH**

The system writes both *before* and *after* images to the journal for changes to records in this file.

Top

---

## **Journal entries to be omitted (OMTJRNE)**

Specifies the journal entries that are omitted.

### **\*NONE**

No journal entries are omitted.

### **\*OPNCLO**

*Open* and *close* entries are omitted. *Open* and *close* operations on the specified file members do not create *open* and *close* journal entries. This prevents the use of TOJOB0 and TOJOB C entries on the Apply Journalized Changes (APYJRNCHG) and Remove Journalized Changes (RMVJRNCHG) commands, but it saves some storage space in the attached receivers.

Top

---

## **Examples**

```
STRJRNP  FILE(MYFILE)  JRN(MYLIB/JRNLA)
```

This command journals all changes to all members of file MYFILE (as found using the library search list) to journal JRNLA in library MYLIB. Only the *after* images of updated records are written to the journal.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF6971**

Damage prevents object &1 from being journaled.

#### **CPF6979**

Journal &1 in library &2 is unusable.

#### **CPF700D**

File &1 in library &2 not journaled.

**CPF7002**  
File &1 in library &2 not a physical file.

**CPF7003**  
Entry not journaled to journal &1. Reason code &3.

**CPF7004**  
Maximum number of objects journaled to journal &1.

**CPF7011**  
Not enough storage or resources.

**CPF703C**  
DDL transaction prevents journaling operation.

**CPF703D**  
DDL transaction prevents journaling operation.

**CPF703E**  
DDL transaction prevents journaling operation.

**CPF7030**  
Object of type \*&3 already being journaled.

**CPF7031**  
Cannot allocate member &3 file &1 in &2.

**CPF7033**  
Start or end journaling failed for member &3.

**CPF7034**  
Logical damage of file &1 in &2.

**CPF7035**  
Object &1 in &2 already known to journal.

**CPF704B**  
Journaling started locally but distributed requests failed.

**CPF705A**  
Operation failed due to remote journal.

**CPF708D**  
Journal receiver found logically damaged.

**CPF7084**  
Object of type \*&6 could not be journaled.

**CPF709D**  
Cannot start journaling object of type \*&7.

**CPF9801**  
Object &2 in library &3 not found.

**CPF9802**  
Not authorized to object &2 in &3.

**CPF9803**  
Cannot allocate object &2 in library &3.

**CPF9810**  
Library &1 not found.

**CPF9812**  
File &1 in library &2 not found.

**CPF9820**

Not authorized to use library &1.

**CPF9822**

Not authorized to file &1 in library &2.

**CPF9825**

Not authorized to device &1.

**CPF9830**

Cannot assign library &1.

**CPF9873**

ASP status is preventing access to object.

**CPF9875**

Resources exceeded on ASP &1.

[Top](#)



---

## Start Mode (STRMOD)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Mode (STRMOD) command starts one or all modes currently in use for an advanced program-to-program communications (APPC) remote location. The user can use STRMOD in either the reset or ended state; it is required only after an End Mode (ENDMOD) command has ended a mode. More information is in the APPC Programming book, SC41-5443.

**Restriction:** The user must have operational authority for the APPC device to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	<i>Communications name</i>	Required, Positional 1
DEV	Device	<i>Name, *<u>LOC</u></i>	Optional, Positional 2
MODE	Mode	<i>Communications name, *<u>NETATR</u>, *ALL</i>	Optional
LCLLOCNAME	Local location	<i>Communications name, *<u>LOC</u>, *NETATR</i>	Optional
RMTNETID	Remote network identifier	<i>Communications name, *<u>LOC</u>, *NETATR, *NONE</i>	Optional

Top

---

### Remote location (RMTLOCNAME)

Specifies the remote location name.

This is a required parameter.

Top

---

### Device (DEV)

Specifies the device description name.

The possible values are:

**\*LOC** The device description is determined by the system.

*device-name*

Specify the name of the device description.

Top

---

## Mode (MODE)

Specifies the mode that is to be started.

The possible values are:

### \*NETATR

The mode in the network attributes is used.

**\*ALL** All modes currently in use for the remote location are to be started.

- For a device description automatically created by the APPN support or a device description manually created with the APPN parameter specified as \*YES, \*ALL indicates that any modes that have been used while the remote location was active, but are not currently started, are to be started.
- For a device description manually created with the APPN parameter specified as \*NO, \*ALL specifies that all configured modes for the specified remote location are to be started.

### **BLANK**

The mode name (consisting of 8 blank characters) is used.

### *mode-name*

Specify a mode name.

**Note:** SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

---

## Local location (LCLLOCNAME)

Specifies the local location name.

The possible values are:

\*LOC The local location name is determined by the system.

### \*NETATR

The LCLLOCNAME value specified in the system network attributes is used.

### *local-location-name*

Specify the name of your location. The local location name is specified if you want to indicate a specific local location name for the remote location.

Top

---

## Remote network identifier (RMTNETID)

Specifies the remote network ID used with the remote location.

The possible values are:

\*LOC The system selects the remote network ID.

### \*NETATR

The remote network identifier specified in the network attributes is used.

### **\*NONE**

No remote network identifier (ID) is used.

*remote-network-id*

Specify the name of the remote network ID.

Top

---

## Examples

```
STRMOD  RMTLOCNAME (APPCRLOC)  DEV (APPCDEV)  MODE (APPCMODE)
        RMTNETID (CHICAGO)
```

This command starts a mode named APPCMODE for a remote location named APPCRLOC, a device named APPCDEV, and a remote network ID of CHICAGO.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF598B

The &1 command failed for one or more modes.

Top





---

# Start Mail Server Framework (STRMSF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Mail Server Framework (STRMSF) command starts the mail server framework jobs in the system work subsystem (QSYSWRK).

Top

---

## Parameters

Keyword	Description	Choices	Notes
MSGOPT	How to process mail messages	*RESUME, *RESET, *CLEAR	Optional, Positional 1
NBRMSFJOB	Number of MSF jobs	1-99, <u>3</u>	Optional, Positional 2

Top

---

## How to process mail messages (MSGOPT)

Specifies how the mail server framework processes existing mail server framework messages.

The possible values are:

### \*RESUME

All existing mail server framework messages continue processing from the point the mail server framework previously ended.

### \*RESET

All existing mail server framework messages are processed as if they were just created.

### \*CLEAR

All existing mail server framework messages are deleted. This option should only be used when a software error is reported with the mail server framework or its associated exit point programs.

Top

---

## Number of MSF jobs (NBRMSFJOB)

Specifies the number of mail server framework jobs to start. This option allows concurrent processing of several mail server framework messages.

The possible values are:

3 Three jobs are started.

### *number-of-jobs*

Specify the number of jobs you want handling mail server framework messages. The valid values range from 1 through 99.

---

## Examples

### Example 1: Starting One Mail Server Framework Job

```
STRMSF  NBRMSFJOB(1)
```

This command starts one mail server framework job in a normal manner, processing any mail server framework messages at the point at which processing was interrupted.

### Example 2: Restarting Mail Server Framework Jobs

```
STRMSF  NBRMSFJOB(3)  MSGOPT(*RESET)
```

This command starts three mail server framework jobs and any mail server framework messages which were partially handled by previous mail server framework jobs are processed again from the beginning.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFAFAA

STRMSF did not complete successfully.

#### CPFAFAD

Mail Server Framework currently active.

#### CPFAFA0

Errors detected on MSF internal message index.

#### CPFAFA1

Errors detected on MSF internal message queue.

#### CPFAFFF

Internal system error in program &1.

Top

---

## Start NFS Server (STRNFSSVR)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Network File System Server (STRNFSSVR) command starts one or all of the following Network File System (NFS) server daemons. For more information about these daemon jobs, see OS/400 Network File System book, SC41-5714

SERVER(\*ALL) should be used, which will start the daemons in the following order. (This order is the recommended order for starting the Network File System daemons.)

- The Remote Procedure Call (RPC) RPCBind daemon
- The block input/output (I/O) (BIO) daemon
- The server (SVR) daemon
- The mount (MNT) daemon
- The network status monitor (NSM) daemon
- The network lock manager (NLM) daemon

If just one daemon is to be started, be sure the appropriate order for starting NFS daemons and the possible consequences of starting daemons in an order other than that specified above are understood. For more information about starting NFS daemons, see OS/400 Network File System book, SC41-5714

If the user attempts to start a daemon or daemons that are already running, they will not cause the command to fail, and it will continue to start other daemons that were requested to start. The command will issue diagnostic message CPDA1BA or CPDA1BD if the daemon is already running. However, for best results, end NFS daemons before attempting the STRNFSSVR command.

To determine if an NFS daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following jobs:

QNFSRPCD The RPCBind daemon  
QNFSBIOD The block I/O (BIO) daemon  
QNFSNFSD The NFS server (SVR) daemon  
QNFSMNTD The mount (MNT) daemon  
QNFSNSMD The network status monitor (NSM) daemon  
QNFSNLMD The network lock manager (NLM) daemon

### Restrictions:

1. The user must have input/output (I/O) system configuration (\*IOSYSCFG) special authority to use this command.
2. The user must be enrolled in the system distribution directory. Use the Add Directory Entry (ADDDIRE) command to enroll the user.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SERVER	Server daemon	*ALL, *RPC, *BIO, *SVR, *MNT, *NSM, *NLM	Required, Positional 1

Keyword	Description	Choices	Notes
NBRSVR	Number of server daemons	1-20, <u>1</u>	Optional, Positional 2
NBRBIO	Number of block I/O daemons	1-20, <u>1</u>	Optional
RTVPCREG	Retrieve RPC registration	*NO, *YES	Optional
STRJOBTIMO	Timeout for start of daemon	1-3600, <u>30</u> , *NOMAX	Optional

Top

---

## Server daemon (SERVER)

Specifies the Network File System (NFS) daemon jobs to be started by this command. The specified daemon should not already be running.

- \***ALL** All NFS daemons will be started.
- \***RPC** The NFS RPCBind daemon will be started.
- \***BIO** Starts NFS block input/output (I/O) daemons. Additional daemons will be started if the number specified on the **Number of block I/O daemons (NBRBIO)** parameter is greater than the number of block I/O daemons already running on the system.
- \***SVR** Starts NFS server daemons. Additional daemons will be started if the number specified on the **Number of server daemons (NBRSVR)** parameter is greater than the number of server daemons already running on the system.
- \***MNT** The NFS mount daemon will be started.
- \***NSM** The NFS network status monitor daemon will be started.
- \***NLM** The NFS network lock manager daemon will be started.

This is a required parameter.

Top

---

## Number of server daemons (NBRSVR)

Specifies the number of NFS server (\*SVR) daemon jobs the user wants to have running. Additional daemons will be started if the number specified on this parameter is greater than the number of server daemons already running on the system. This parameter can only be used if SERVER(\*SVR) or SERVER(\*ALL) is specified.

- 1 One NFS server daemon job should be started if there are not already any NFS server daemons running.
- 1-20 Specify the number of NFS server daemon jobs the user wants to have running.

Top

---

## Number of block I/O daemons (NBRBIO)

Specifies the number of NFS block input/output (I/O) (\*BIO) daemon jobs the user wants to have running. Additional daemons will be started if the number specified on this parameter is greater than the number of block I/O daemons already running on the system. This parameter can only be used if SERVER(\*BIO) or SERVER(\*ALL) is specified.

- 1 One NFS block I/O daemon job should be started if there are not already any NFS block I/O daemons running.
- 1-20** Specify the number of NFS block I/O daemon jobs the user wants to have running.

Top

---

## Retrieve RPC registration (RTVPCREG)

Specifies whether to retrieve previously recorded registration information when the RPCBind daemon is started. If registration information is retrieved, any services already registered with the RPCBind daemon do not have to re-register with the RPCBind daemon. This parameter can only be used if SERVER(\*RPC) or (SERVER(\*ALL)) is specified.

- \*NO Do not retrieve registration information.
- \*YES Retrieve registration information.

Top

---

## Timeout for start of daemon (STRJOBTIMO)

Specifies the number of seconds to wait for each daemon to successfully start. If a daemon has not started within the timeout value, the command will fail.

- 30 Default seconds before timeout.
- \*NOMAX  
Wait forever for daemons to start; do not timeout.
- 1-3600** Specify a number of seconds to wait for daemons to start before timing out and failing the command. Timeout values less than 30 seconds are rounded up to 30 seconds.

Top

---

## Examples

### Example 1: Start All NFS Daemons

```
STRNFSSVR SERVER(*ALL) STRJOBTIMO(*NOMAX)
```

This command starts all NFS daemons, and waits forever for them to start. No daemons should be previously running.

### Example 2: Start Only One Daemon

```
STRNFSSVR SERVER(*MNT)
```

This command starts the NFS mount daemon, and waits up to the default of 30 seconds for it to start. The mount daemon should not be already running, and other daemons have been started in the appropriate order.

Top

---

## Error messages

None

Top



---

## Start NetWare Connection (STRNTWCNN)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start NetWare Connection (STRNTWCNN) command starts an authenticated connection to the specified server. This command is used to:

- Start a connection when there is no password in the authentication entry for the server (the password value is \*STRNTWCNN).
- Start a connection using authentication information different from that specified in the user profile authentication entry.
- To specify a preferred server for NetWare operations that might otherwise use a different server.

If a password is stored in the authentication entry, this command is not needed; the system will start and end connections as necessary.

Connections can be authorized to be used only by the current job (AUTJOB(\*)), or can be authorized to be used by any job running under the specified user profile (AUTJOB(\*ANY)), for example, a batch job. Each user profile can only have one connection per job open to a given server at a time.

For NetWare Directory Services trees, connections may be started to each server within the tree. The special value \*ALL may be used to start connections to all servers. This could be used, for example, when copying from one server in the tree to another, or copying between partitions of the NDS tree. If multiple connections exist, operations that do not require a specific server will be performed using the first connection started.

The NetWare backup services require that each user perform a separate authentication in addition to a normal login. The connection type (CNNTYPE) parameter allows you to specify that a \*SAVRST authentication be performed for this purpose.

**Restrictions:** To start a connection for another user profile (specified in AUTUSR parameter), you need to have \*USE authority to that user profile.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SERVER	Server	Character value, *ALL, *ANY	Required, Positional 1
SVRTYPE	Server type	*SERVER, *NDS	Optional
NDSTREE	NDS tree	Character value, *SERVER	Optional
AUTUSR	Authorized user profile	Name, *CURRENT	Optional
NDSCTX	NDS context	Character value, *AUTE, *USRPRF	Optional
NTWUSER	NetWare user	Character value, *AUTE, *USRPRF	Optional
PASSWORD	Password	Character value, *NONE, *AUTE	Optional
AUTJOB	Authorized job	*, *ANY	Optional
CNNTYPE	Connection type	Single values: *ALL Other values (up to 2 repetitions): *USER, *SAVRST	Optional

Keyword	Description	Choices	Notes
CNNIDLTIME	Connection idle time	1-9999, <u>*NOMAX</u>	Optional

Top

---

## Server (SERVER)

Specify the server to which the connection is to be started.

**\*ANY** For \*NDS servers, start a connection to any server within the NDS tree.

**\*ALL** For \*NDS servers, start connections to all servers within the NDS tree.

*character-value*

Specify the name of the server to which the connection is to be started.

Top

---

## Server type (SVRTYPE)

Specifies the type of server.

**\*SERVER**

Use system information about the specified server to determine the server type. This value cannot be used when \*ANY or \*ALL is specified for the **Server (SERVER)** parameter.

**\*NDS** The server is in a NetWare Directory Services tree.

Top

---

## NDS tree (NDSTREE)

For \*NDS servers, specifies the name of the NetWare Directory Services (NDS) tree to which the connection is to be started.

**\*SERVER**

Use system information about the specified server to determine the NDS tree name. This value cannot be used when \*ANY or \*ALL is specified for the SERVER parameter.

*character-value*

Specify the name of NDS tree.

Top

---

## Authorized user profile (AUTUSR)

Specifies the user profile authorized to use the connection.

**\*CURRENT**

The connection can only be used by jobs running under the current user profile.

*name* Specify the name of the user profile. The connection can only be used by jobs running under the specified user profile.

Top



---

## NDS context (NDSCTX)

Specifies, for \*NDS servers, the NetWare Directory Services context in which the NetWare user name is defined.

### \*AUTE

Use the authentication entry for this tree in the user profile to determine the NDS directory context.

### \*USRPRF

Use the authorized user profile name as the NDS directory context name.

### *character-value*

Specify the name of NDS directory context.

Top

---

## NetWare user (NTWUSER)

Specifies the NetWare user name used to authenticate the user.

### \*AUTE

Use the authentication entry for the specified server or NDS tree in the user profile to determine the NetWare user name.

### \*USRPRF

Use the authorized user profile name as the NetWare user name.

### *character-value*

Specify the NetWare user name.

Top

---

## Password (PASSWORD)

Specifies the password to be used to verify the authority for the specified user.

### \*AUTE

Use the password in the authentication entry for this server or NDS tree in the user profile.

### \*NONE

No password is used to verify authority.

### *character-value*

Specify the password to be used.

Top

---

## Authorized job (AUTJOB)

Specifies the job authorized to use the connection. The connection can be used either by the current job, or by any job running under the specified user profile name.

### \*

The connection can only be used by the current job. The connection will be closed when the job ends.

\*ANY The connection can be used by any job running under the specified user profile. The connection must be ended by an End NetWare Connection (ENDNTWCNN) request.

**Notes:**

- The QNETWARE file system requires a connection for the current job (AUTJOB(\*)).
- If a connection cannot be started using an authentication entry, printing to a NetWare print server requires a connection for any job (AUTJOB(\*ANY)).
- Connections for AUTJOB(\*) are also ended if the QFPZAUT activation group is reclaimed (RCLACTGRP command).

Top

---

## Connection type (CNNTYPE)

Specifies the types of authentication to be performed. NetWare backup services require a separate authentication in addition to a normal user login.

### Single values

**\*ALL** Both \*USER and \*SAVRST authentications will be performed.

### Other values (up to 2 repetitions)

#### \*USER

A normal user authentication will be done. This will allow the user to use administration and file system services other than save and restore.

#### \*SAVRST

Authentication to NetWare backup services will be performed. This option requires that the appropriate NetWare backup NLMs (TSAxxx) be loaded on the server.

Top

---

## Connection idle time (CNNIDLTIME)

Specifies the idle time after which the connection will be closed. The idle time is the time since the most recent request to the server.

#### \*NOMAX

No idle time limit is specified. The connection will remain open until ended by an End NetWare Connection (ENDNTWCNN) request, or until the job is ended, whichever is sooner.

**1-9999** Specify the number of minutes of allowed idle time.

Top

---

## Examples

None

Top

---

## Error messages

### \*ESCAPE Messages

#### **FPE0226**

Connection not started to server &1.

Top

---

## Start Object Conversion (STROBJCVN)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Object Conversion (STROBJCVN) command converts user objects from the format used in the previous version, release, and modification level of an IBM-supported operating system to the format required for use in the current operating system. The following AS/400 objects are converted with this command:

1. Programs (original program model (OPM) and Integrated Language Environment (ILE))
2. Service programs
3. Modules
4. Database files (physical and logical) on Replacing-a-release upgrade methods.

The user objects that are not converted with this command are automatically converted when they are first specified for use.

**Note:** Using objects that are not yet converted will degrade performance of an operation.

### Restrictions:

1. Programs, service programs, and modules must have all observable creation data to be converted. Unobservable creation data cannot be used by this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *ALLUSR, *PTY	Required, Positional 1
OBJTYPE	Object type	*ALL, *FILE, *ALLPGM	Optional, Positional 2
PTY	Priority	1-99, *ALL	Optional, Positional 3

Top

---

## Library (LIB)

Specifies the user library whose objects are to be converted to the format of the current operating system.

This is a required parameter.

The possible values are:

### \*ALLUSR

All user libraries are converted. All libraries with names that do not begin with the letter Q are converted except for the following:

#CGULIB    #DFULIB    #RPGLIB    #SEULIB  
#COBLIB    #DSULIB    #SDALIB

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also converted:

QDSNX      QPFRRDATA    QUSER38      QUSRVxRxMx  
QGPL      QRCL          QUSRINFSKR  
QGPL38    QS36F        QUSRSYS

**Note:** A different library name, of the form QUSRVxRxMx, can be created by the user for each release that IBM supports. VxRxMx is the version, release, and modification level of the library.

**\*PTY** Convert the set of libraries at the priority specified in the PTY parameter. The object conversion priority for a library can be set using the Plan Object Conversion task.

#### *library-name*

Specifies the name of the user library whose objects are converted.

**Note:** Although only one library can be specified on this command, repeated calls of this command in a CL program can be used to convert a set of user libraries.

Top

---

## Object type (OBJTYPE)

Specifies which objects in the library should be converted to the format of the current operating system.

This is a required parameter.

The possible values are:

**\*ALL** All \*PGM, \*SRVPGM, \*MODULE, and database files in the specified library are converted to the format of the current operating system.

**\*FILE** Only database file member objects in the specified library are converted to the format of the current operating system.

#### **\*ALLPGM**

Only \*PGM and \*SRVPGM objects in the specified library are converted to the format of the current operating system.

Top

---

## Priority (PTY)

Specifies the priority of libraries to be converted to the format of the current operating system. The priority groups are set on the source system during upgrade preparation. This parameter is valid only when LIB(\*PTY) is specified.

The priority for a library can be set with the Plan Object Conversion task of the Upgrade Assistant for OS/400.

This is a required parameter.

The possible values are:

**\*ALL** All user libraries are converted to the format of the current operating system. Specifying this value with LIB(\*PTY) is the same as specifying LIB(\*ALLUSR).

#### *priority-number*

Specify the libraries to be converted to the format of the current operating system.

**Note:** Although only one priority can be specified on this command, repeated calls of this command in a CL program can be used to convert multiple sets of libraries with different priorities.

Top

---

## Examples

```
STROBJCVN LIB(LIB1)
```

This command converts all OPM and ILE programs, service programs, modules, and physical and logical database files contained in the LIB1 user library to the format of the current operating system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPDA972

Errors occurred while converting object &1 in library &2 type \*&3.

#### CPFA96D

No objects converted in library &1.

#### CPFA960

Library &1 is not a user library.

#### CPFA972

Not all eligible objects in library &1 converted.

#### CPFB0C1

Not all eligible objects in priority &1 converted.

#### CPFB0C2

No objects converted in priority &1.

#### CPF9810

Library &1 not found.

Top



---

## Start Pass-Through (STRPASTHR)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Pass-Through (STRPASTHR) command allows you to pass through to a target system where you can sign on as if you were attached locally. For information on configuring or operating the pass-through function, refer to the Remote Work Station Support book.

### Restriction:

1. This command cannot be entered at a work station with a display that has 12 lines by 80 characters.

Top

---

## Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	<i>Communications name</i> , *CNNDEV	Required, Positional 1
CNNDEV	APPC device	Single values: * <u>LOC</u> Other values (up to 16 repetitions): <i>Name</i>	Optional
VRTCTL	Virtual controller	<i>Name</i> , * <u>NONE</u>	Optional, Positional 2
VRTDEV	Virtual display device	Single values: * <u>NONE</u> Other values (up to 32 repetitions): <i>Name</i>	Optional, Positional 3
MODE	Mode	<i>Communications name</i> , * <u>NETATR</u>	Optional
LCLLOCNAME	Local location	<i>Communications name</i> , * <u>LOC</u> , *NETATR	Optional
RMTNETID	Remote network identifier	<i>Communications name</i> , * <u>LOC</u> , *NETATR, *NONE	Optional
SRQ10PGM	System request program	Single values: * <u>SRQMNU</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: System request program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
RMTUSER	User profile	<i>Character value</i> , * <u>NONE</u> , *CURRENT	Optional
RMPWD	User password	<i>Character value</i> , * <u>NONE</u>	Optional
RMTINLPGM	Initial program to call	<i>Name</i> , * <u>RMTUSRPRF</u> , *NONE	Optional
RMTINLMNU	Initial menu	<i>Name</i> , * <u>RMTUSRPRF</u> , *SIGNOFF	Optional
RMTCURLIB	Current library	<i>Name</i> , * <u>RMTUSRPRF</u>	Optional
PASTHRSCN	Display option	<i>Character value</i> , * <u>YES</u> , *NO	Optional

Top

---

## Remote location (RMTLOCNAME)

Specifies one of the following:

- If (\*LOC) is specified for the **APPC device** prompt (CNNDEV parameter), the RMTLOCNAME parameter specifies the name of the remote location that is the target of the pass-through session.

- If any devices are specified on the CNNDEV parameter, the RMTLOCNAME parameter specifies the first system to do intermediate pass-through routing.

*remote-location-name*

Specify the remote location name that is the target of the pass-through session or the first system that does intermediate pass-through routing. APPN determines the route to this location.

**\*CNNDEV**

Use the APPC devices specified with the **APPC device** prompt (CNNDEV parameter).

Top

---

## APPC device (CNNDEV)

Specifies the names of the device descriptions that connect the first system to do pass-through routing with the target system. If \*CNNDEV is specified on the Remote location prompt (RMTLOCNAME parameter), the first device specified on this parameter is on the source system. If the name of a remote location is specified on the Remote location prompt (RMTLOCNAME parameter), the first device specified on this parameter is on the system in that remote location. If another system connects the source system to the target system and pass-through must establish the intermediate sessions, then you must specify a list of APPC device descriptions. The APPC device names must be listed in the order that the systems are passed through to get to the target system.

You can enter multiple values for this parameter.

**\*LOC** The **Remote location** prompt (RMTLOCNAME parameter) specifies the name of the remote location that is the target of the pass-through session.

*device-name(s)*

Specify the names of the device descriptions that complete the route from the source system to the target system. Up to 16 names can be specified.

Top

---

## Virtual controller (VRTCTL)

Specifies the name of the virtual controller on the remote system that is used to do pass-through jobs. If you specify a virtual controller, one of the virtual display devices attached to it is selected for the pass-through job. A device on the target system is selected based on a comparison of device type and model.

This entry is mutually exclusive with an entry on the **Virtual display device** prompt (VRTDEV parameter); \*NONE must be specified when one or more devices are specified on the **Virtual display device** prompt (VRTDEV parameter).

**\*NONE**

No controller is specified. If \*NONE is also specified on the **Virtual display device** prompt (VRTDEV parameter), you are requesting automatic configuration.

*virtual-controller-name*

Specify the name of the virtual controller description on the target system.

Top



---

## Virtual display device (VRTDEV)

Specifies one or more devices on the target system that are connected to a virtual controller used for the pass-through session. A device on the target system from the list is selected based on a comparison of device type and model.

This entry is mutually exclusive with an entry on the **Virtual controller** prompt (VRTCTL parameter); \*NONE must be specified when a controller name is specified on the **Virtual controller** prompt (VRTCTL parameter).

You can enter multiple values for this parameter.

### \*NONE

No device names are specified. If \*NONE is also specified on the **Virtual controller** prompt (VRTCTL parameter), you are requesting automatic configuration.

### *virtual-display-device-name*

Specify the names of the virtual display device descriptions on the target system. Up to 32 names can be specified.

Top

---

## Mode (MODE)

Specifies the mode name used.

### \*NETATR

The mode in the network attributes is used.

### *mode-name*

Specify a mode name.

Top

---

## Local location (LCLLOCNAME)

Specifies the local location name.

\*LOC The local location name is chosen by the system.

### \*NETATR

The LCLLOCNAME value specified in the system network attributes is used.

### *APPN-location-name*

Specify the local location name that is associated with the source system. If the local location name is not valid, an escape message is sent.

Top

---

## Remote network identifier (RMTNETID)

Specifies the network ID of the network where the remote location resides.

\*LOC Any remote network ID for the remote location may be used.

### \*NETATR

The LCLNETID value specified in the system network attributes is used.

**\*NONE**

The remote location does not support network identifiers.

***APPN-network-identifier***

Specify a remote network ID.

Top

---

## System request program (SRQ10PGM)

Specifies that the system request menu is displayed, or specifies the name of the program that starts on the source system when system request option 10 is selected.

The user program can display a menu that allows you to select the system you want to access, and then transfer to a group job that sends the Start Pass-Through (STRPASTHR) command to the desired system. For more information, see the Remote Work Station Support book.

**\*SRQMNU**

The system request menu on the source system is displayed.

***program-name***

Specify the program and library names of the program started when system request option 10 is selected.

The possible library values are:

**\*LIBL** The library list is used to locate the program.

**\*CURLIB**

The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

***library-name***

Specify the library where the program is located.

Top

---

## User profile (RMTUSER)

Specifies the user profile for automatic sign-on to the target system. If a profile is specified for this parameter and password security is active on the target system, (\*NONE) is not valid for the User password prompt (RMTPWD parameter).

**\*NONE**

No user profile name is sent, and no automatic sign-on occurs.

**\*CURRENT**

The user profile of the job using this command is sent. If the target system allows it, and the user profile exists on the target system, and the password specified in the RMTPWD parameter is valid for the profile, the user is automatically signed on. Otherwise, the user is presented with a sign-on display on the target system or a failure message on the source system, depending on the configuration of the target system.

***profile-name***

Specify a user profile name to use that exists on the target system. If the target system allows it, and if the user profile exists on the target system, the user is automatically signed on. Otherwise, the user is presented with a sign-on display on the target system or a failure message on the

source system, depending on the configuration of the target system. If a profile is specified and password security is active on the target system, a password must be specified, even if the profile specified is the same as the current profile.

Top

---

## User password (RMTPWD)

Specifies the password being sent to the target system.

### \*NONE

The system does not send a password. If a profile is specified on the **User profile** prompt (RMTUSER parameter), and password security is active on the target system, this value is not allowed.

### *password*

Specify a password being sent to the target system to verify the sign-on of the user specified in the RMTUSER parameter. This password is encrypted before being sent across the communication line.

Top

---

## Initial program to call (RMTINLPGM)

Specifies the program that is called immediately after you sign on.

### \*RMTUSRPRF

The initial program specified in the remote user profile is called immediately after automatic sign-on.

### \*NONE

No program is run before the initial menu is shown, even if an initial program is specified in the remote user profile.

### *program*

Specify the name of a program that is called immediately after automatic sign-on.

Top

---

## Initial menu (RMTINLMNU)

Specifies the first menu shown when you are automatically signed on to the target system after the initial program is run.

### \*RMTUSRPRF

The initial menu specified in the remote user profile is shown immediately after the initial program is run.

### \*SIGNOFF

A menu is not shown after the initial program is run, even if an initial menu is specified in the remote user profile. After the initial program ends, the user is signed off, and the pass-through session ends.

*menu* Specify the menu that is shown immediately after the initial program is run.

Top

---

## Current library (RMTCURLIB)

Specifies the name of the library that becomes the current library in the library list of the job after automatic sign-on.

### \*RMTUSRPRF

The current library specified in the remote user profile becomes the current library in the library list after automatic sign-on.

### *library*

Specify the library that becomes the current library in the library list after automatic sign-on.

Top

---

## Display option (PASTHRSCN)

Specifies whether the pass-through display and associated status messages appear before the pass-through session is established.

\*YES The pass-through display and information messages are shown before the pass-through session is established.

\*NO The pass-through display and information messages are not shown before the pass-through session is established.

Top

---

## Examples

### Example 1: Pass-Through to Toronto

```
STRPASTHR RMTLOCNAME(*CNNDEV) CNNDEV(DET CHI TOR) VRTCTL(VWSC)
```

This command specifies starting a pass-through to the Toronto system by going through Detroit and Chicago. More information is in the **Remote Work Station Support** book located in the Information Center.

### Example 2: Pass-Through to Detroit

```
STRPASTHR RMTLOCNAME(DETROIT) VRTCTL(VWSC)
```

This command specifies a pass-through to the Detroit system. APPN establishes the route to Detroit.

### Example 3: Pass-Through to Toronto

```
STRPASTHR RMTLOCNAME(DETROIT) CNNDEV(CHI TOR) VRTCTL(VWSC)
```

This command specifies another way to pass-through to the Toronto system by going through Chicago and Detroit. APPN establishes the route to Detroit.

### Example 4: Pass-Through to Detroit

```
STRPASTHR RMTLOCNAME(DETROIT) RMTUSER(*CURRENT)
```

This command specifies a pass-through to the DETROIT system and an automatic sign-on using the user profile with the same name as the one currently used on the source system. It also specifies that the DETROIT system automatically configures a virtual device for the pass-through session, since a virtual controller or virtual device was not specified.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF2702**

Device description &1 not found.

**CPF2703**

Controller description &1 not found.

**CPF5383**

Mode &7 specified for device &4 not valid.

**CPF5536**

System cannot automatically select an APPC device description for the remote location.

**CPF5546**

Class-of-service for device &4 not valid.

**CPF8901**

Virtual device &1 not varied on.

**CPF8902**

Virtual device &1 not available.

**CPF8903**

Device &1 not valid for pass-through.

**CPF8904**

Pass-through request not accepted.

**CPF8905**

Pass-through not allowed on this system.

**CPF8906**

Error during session initialization. Reason code &1.

**CPF8907**

Communications failure for device &1.

**CPF8908**

Controller &1 not varied on.

**CPF8909**

Old software release. Pass-through ended.

**CPF8910**

Controller &1 not valid for pass-through.

**CPF8911**

Communications failure. Session was not started.

**CPF8912**

Pass-through session ended. Reason code &1.

**CPF8913**

Pass-through ended abnormally.

**CPF8916**

Cannot select virtual device &1 at system &2.

**CPF8917**

Not authorized to &1.

**CPF8918**

Job canceled at system &1.

- CPF8919**  
Device &1 not accessed by system &2.
- CPF8920**  
Pass-through failed. &1 must be varied off and on.
- CPF8921**  
APPC failure. Failure code is &3.
- CPF8922**  
Negative response from device &1 at system &2.
- CPF8923**  
Data stream received at system &1 not valid.
- CPF8924**  
No available virtual controller.
- CPF8925**  
Device &1 not created.
- CPF8928**  
Device &1 could not be changed.
- CPF8929**  
Device &1 could not be varied on.
- CPF8931**  
Location &1 not an APPC location.
- CPF8932**  
Device &1 must be non-networking APPC device.
- CPF8933**  
Route to specified location not found.
- CPF8935**  
Pass-through not allowed to system &1.
- CPF8936**  
Pass-through failed for security reasons.
- CPF8937**  
Automatic sign on not allowed.
- CPF8938**  
Error in QRMTSIGN program. Pass-through failed.
- CPF8939**  
Trying to send too much data.
- CPF8940**  
Cannot automatically select virtual device.
- CPF8941**  
Incorrect internal use of pass-through.
- CPF8943**  
Pass-through not allowed from server TELNET session.

**\*STATUS Messages**

- CPI8901**  
No matching device on remote system. Function limited.

**CPI8902**

Pass-through started at system &1.

**CPI8903**

Virtual device &1 selected at system &2.

**CPI8906**

Automatic sign-on not allowed.

[Top](#)





---

## Start PC Command (STRPCCMD)

### Where allowed to run:

- Interactive job (\*INTERACT)
- Interactive program (\*IPGM)
- Using QCMDXEXEC, QCAEXEC, or QCAPCMD API (\*EXEC)

Threadsafe: No

Parameters  
Examples  
Error messages

The Start PC Command (STRPCCMD) command allows you to run a single application, a DOS command, or an OS/2 command on an attached personal computer.

**Note:** Do not precede an entry with an asterisk unless that entry is a "special value" that is shown (on the display itself or in the help information) with an asterisk.

---

## Error messages for STRPCCMD

None

Top

---

## Parameters

Keyword	Description	Choices	Notes
PCCMD	PC command	<i>Character value</i>	Required, Positional 1
PAUSE	Pause	*YES, *NO	Optional

Top

---

## PC command (PCCMD)

Specifies the valid DOS command, OS/2 command, or personal computer application you want to run. If you are using the STRPCCMD command on an AS/400 command line and the command or application name contains special characters, you must enclose the command in apostrophes. Special characters include blanks, commas, and colons.

If you select the option to start a PC command from the Client Access/400 Organizer menu, you do not need to enclose the command in apostrophes unless it ends in a colon. The ending colon is ignored unless the entire PC command is enclosed in apostrophes.

This is a required parameter.

Top

---

## Pause (PAUSE)

Specifies whether the computer should pause after running a command.

The possible values are:

**\*YES** The computer will pause after running the command before returning to the Client Access Organizer menu.

**\*NO** The computer returns immediately to the Client Access Organizer menu after the command runs.

[Top](#)

---

## Examples

None

[Top](#)

---

## Error messages

None

[Top](#)

---

## Start PC Organizer (STRPCO)

### Where allowed to run:

- Interactive job (\*INTERACT)
- Interactive program (\*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (\*EXEC)

Threadsafe: No

Parameters  
Examples  
Error messages

The Start Client Access Organizer (STRPCO) command starts the Client Access Organizer on the host system.

**Note:** Do not precede an entry with an asterisk unless that entry is a "special value" that is shown (on the display itself or in the help information) with an asterisk.

---

## Error messages for STRPCO

None

Top

---

## Parameters

Keyword	Description	Choices	Notes
PCTA	Text Assist	Character value, <u>*YES</u> , *NO	Optional, Positional 1

Top

---

## Text Assist (PCTA)

Specifies whether you are going to use the Personal Computer Text Assist.

The possible values are:

**\*YES** The Personal Computer Text Assist is going to be used.

**\*NO** The Personal Computer Text Assist is not going to be used.

Top

---

## Examples

None

Top

---

## Error messages

None



---

## Start Performance Explorer (STRPEX)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
Examples  
Error messages

The Start Performance Explorer (STRPEX) command starts a new Performance Explorer session or resumes a suspended Performance Explorer session.

### Restrictions:

1. This command is shipped with public \*EXCLUDE authority.
2. To use this command you must have \*SERVICE special authority, or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
3. The following user profiles have private authorities to use the command:
  - QPGMR
  - QSRV
4. Two threads within the same job will not be allowed to run STRPEX at the same time. The thread that issued STRPEX first will run the command to completion while the second STRPEX waits.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	<i>Name</i>	Required, Key, Positional 1
OPTION	Option	*NEW, *INZONLY, *RESUME	Optional, Positional 2
DFN	Definition	<i>Name</i> , *SELECT	Optional
FTR	Filter	<i>Name</i> , *NONE, *SELECT	Optional

Top

---

## Session ID (SSNID)

Specifies an identifier for this Performance Explorer session. This name must be unique within the active sessions of the Performance Explorer tool.

This is a required parameter.

### *session-identifier*

Specify the session identifier for a new or suspended Performance Explorer session.

Top

---

## Option (OPTION)

Specifies whether a new session should be started or a session which is currently suspended should be resumed.

This is a required parameter.

**\*NEW** A new session of the Performance Explorer tool is started. Performance data collection begins as soon as the session has been established.

**\*INZONLY**

A new Performance Explorer session is started, but once the session is established, it is suspended. This option allows the user to perform the setup for a particular session before the scenario to be monitored is started. To begin data collection, the user must invoke this command again specifying the same session identifier and OPTION(\*RESUME).

**\*RESUME**

A suspended Performance Explorer session is resumed. The session was suspended either by specifying OPTION(\*SUSPEND) on the ENDPEX command or by specifying OPTION(\*INZONLY) on a previous STRPEX command.

Top

---

## Definition (DFN)

Specifies the name of the Performance Explorer definition to use when starting a new Performance Explorer session. The definition identifies the performance data to be collected. This parameter is required in order to start a new Performance Explorer session. This parameter cannot be specified when resuming a suspended session, since the suspended session already has an associated Performance Explorer definition.

*definition-name*

Specify the name of the Performance Explorer definition.

Top

---

## Filter (FTR)

Specifies the name of the Performance Explorer filter to use when starting a new Performance Explorer session. The filter determines which events are collected based on the filter values which are compared to the event's actual data. If a filter is not specified, then all events in the definition are collected. This parameter cannot be specified when resuming a suspended session.

**\*SELECT**

A list of existing filters will be displayed for user selection. This value is valid only if the command is run in an interactive job.

*filter-name*

Specify the name of the Performance Explorer filter.

Top

---

## Examples

### Example 1: Start a New Session

```
STRPEX  SSNID(TESTRUN2)  DFN(NEWDESC)  OPTION(*NEW)
        FTR(MYFILTER)
```

This command starts a new session of the performance explorer using the criteria identified in a definition named NEWDESC and a filter named MYFILTER. The new session name is TESTRUN2.

**Example 2: Resume a Suspended Session**

```
STRPEX  SSNID(TESTRUN1)  OPTION(*RESUME)
```

This command resumes an already existing session of the performance explorer named TESTRUN1.

[Top](#)

---

## Error messages

---

### Error messages for RMVPEXDFN

#### \*ESCAPE Messages

**CPFAF05**

STRPEX command was not successful. Reason code is &1. See details for more information.

**CPFAF11**

Unable to locate program, object, library, or file member as specified in the definition or filter.

[Top](#)





---

## Start Performance Collection (STRPFRCOL)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Performance Collection (STRPFRCOL) command starts the system-level collection of performance data by Collection Services. The properties of the system-level collection are controlled by the Collection Services configuration, which can be changed using the Configure Performance Collection (CFGPFRCOL) command. The data included in the system-level collection is determined by the value specified on the Collection profile parameter.

Performance data collection is conducted by the Collection Services server job (QYPSPFRCOL). If this job is not active, it will be started as a result of this command. If the job is active, any change to the collection profile will take effect immediately and the collection will continue uninterrupted.

QYPSPFRCOL creates a management collection object (\*MGTCOL) to store performance data. Data collected will include both the system-level collection profile categories as well as data collected on behalf of client applications (for example, iSeries Navigator monitors or Performance Collector APIs). All data is stored in the current management collection object. This data may be processed using the Create Performance Data (CRTPFRDTA) command to generate the performance database files.

Cycling the performance collection will cause the Collection Services server job to create a new management collection object prior to implementing the specified Collection profile. Note: cycling may result in the loss of one interval of data and will also impact any client applications that are using Collection Services.

Top

---

### Parameters

Keyword	Description	Choices	Notes
COLPRF	Collection profile	*CFG, *MINIMUM, *STANDARD, *STANDARDP, *ENHCPCPLN, *CUSTOM	Optional, Positional 1
CYCCOL	Cycle collection	*NO, *YES	Optional, Positional 2

Top

---

### Default collection profile (COLPRF)

Specifies the collection profile. This will determine which categories will be included in the system-level collection. Specifying a profile for this parameter will change the default collection profile in the Collection Services configuration. The default collection profile may also be changed using the Configure Performance Collection (CFGPFRCOL) command.

**\*CFG** The collection will use the currently configured value. If the configured value has never been modified, the collection profile will be \*STANDARDP.

**\*MINIMUM**

The minimum data collection recommended. Includes the following categories: \*SYSBUS, \*POOL, \*HDWCFG, \*SYSCPU, \*SYSLVL, \*JOBMI, \*JOBOS, \*DISK, and \*IOPBASE.

## **\*STANDARD**

The standard profile includes all categories which are typically needed by the Performance Tools for iSeries, with the exception of communications data. It includes all categories in the \*MINIMUM profile, as well as the following categories: \*POOLTUNE, \*SUBSYSTEM, \*SNADS, \*LCLRSP, \*APPN, \*SNA, \*TCPBASE, \*USRTNS, and \*LPAR. The category \*DOMINO will be included if the product Domino for iSeries has been installed on the system. The category \*HTTP will be included if the product IBM HTTP Server for iSeries (powered by Apache) has been installed on the system.

## **\*STANDARDP**

The standard plus profile includes all categories which are typically needed by the Performance Tools for iSeries, including communications data. It includes all categories in the \*STANDARD profile, as well as the following categories: \*IPCS, \*CMNBASE, \*CMNSTN, \*CMNSAP, \*TCPIFC, and \*DPS.

## **\*ENHCPCPLN**

The enhanced capacity planner profile includes all categories in the \*STANDARDP profile, with the addition of \*INTPEX.

## **\*CUSTOM**

The custom profile includes categories as defined by the user. This profile must be defined using the Collection Services function in iSeries Navigator.

Top

---

## **Cycle collection (CYCCOL)**

Specifies whether the collection should be cycled. Cycling the collection will cause data to be collected in a new management collection object (\*MGTCOL).

**\*NO** Data will continue to be collected in the current collection object.

**\*YES** Data will be collected in a new collection object.

Top

---

## **Examples**

### **Example 1: Starting the Performance Collection**

```
STRPFRCOL
```

This command will cause the Collection Services server job (QYSPFRCOL) to start the system-level collection of performance data using the currently configured default collection profile.

### **Example 2: Starting with Collection Profile and Cycle Collection**

```
STRPFRCOL COLPRF(*MINIMUM) CYCCOL(*YES)
```

This command will cause the Collection Services server job (QYSPFRCOL) to cycle and begin collecting performance data in a new management collection object (\*MGTCOL) using the \*MINIMUM collection profile. Note: this command will also change the configured default collection profile to \*MINIMUM.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

**CPF3CF2**

Error(s) occurred during running of &1 API.

**CPFB94A**

Collector communications error. Reason code &1.

[Top](#)



---

## Start Performance Trace (STRPFRTTC)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Performance Trace (STRPFRTTC) command is a simplified interface to the TRCINT command for collecting Multiprogramming level (MPL) and Transaction trace data. This command collects the same performance trace data as was collected in previous releases by the Start Performance Monitor (STRPFRTMON) command.

The trace started by this command creates and uses trace table QPM\_STRPFRTTC. If the trace table exists, any existing data will be deleted before this trace begins.

The trace can be stopped and the data can be written to a data base file by using the End Performance Trace (ENDPFRTTC) command.

### Restrictions:

- This command is shipped with public \*EXCLUDE authority.
- The following user profiles have private authorities to use the command:
  - QSRV

Top

---

## Parameters

Keyword	Description	Choices	Notes
SIZE	Trace table size	Single values: *CALC, *MAX Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: Number of units	1-998000	
	Element 2: Unit of measure	*KB, *MB	
OMTTCPNT	Omit trace points	*NONE, *RSCMGT	Optional
JOBTYPE	Job types	Single values: *NONE, *ALL Other values (up to 12 repetitions): *DFT, *ASJ, *BCH, *EVK, *INT, *MRT, *RDR, *SBS, *SYS, *WTR, *PDJ, *PJ, *BCI	Optional
JOBTRCIV	Job trace interval	0.1-9.9, <u>0.5</u>	Optional

Top

---

## Trace table size (SIZE)

Specifies the size of the trace table.

**Note:** The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). Refer to the Trace Internal (TRCINT) command SIZE parameter for additional information regarding the setting of trace table size.

Specifying a size of less than 16 megabytes is not recommended.

## Single values

### \*CALC

The minimum trace table size is determined based on the processor group of your system.

**\*MAX** The trace table is set to the maximum size of 258048 megabytes.

## Element 1: Number of units

Specify the size of the trace table.

### **1-998000**

Specify the size of the trace table in kilobytes or megabytes.

## Element 2: Unit of measure

Specify whether the value specified for the first element of this parameter should be treated as number of kilobytes or number of megabytes.

**\*KB** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

**\*MB** The trace table size is specified in megabytes. The valid range is 1 through 258048.

Top

---

## Omit trace points (OMTTRCPNT)

Specifies trace points whose trace records are to be excluded.

### \*NONE

No trace points are to be excluded.

### **\*RSCMGT**

Resource management trace points (seize/lock conflict data) will be excluded from the trace.

Top

---

## Job types (JOBTYPE)

Specifies the types of jobs for which trace data is to be collected for use in the batch job trace report.

**Note:** The value \*DFT includes the values \*ASJ, \*BCH, \*EVK, \*MRT, \*PDJ, \*PJ and \*BCI. The value \*BCH includes the values \*EVK, \*MRT, \*PDJ, \*PJ, and \*BCI.

## Single values

### **\*NONE**

No jobs are to be traced.

**\*ALL** All job types are to be traced.

## Other values (up to 12 repetitions)

**\*DFT** Batch and autostart jobs are traced.

**\*ASJ** Autostart jobs are traced.

**\*BCH** Batch jobs are traced.

**\*EVK** Jobs started by a procedure start request are traced.

**\*INT** Interactive jobs are traced.

- \*MRT Multiple requester terminal jobs are traced.
- \*RDR Reader jobs are traced.
- \*SBS Subsystem monitor jobs are traced.
- \*SYS System jobs are traced.
- \*WRT Writer jobs are traced.
- \*PDJ Print driver jobs are traced.
- \*PJ Prestart jobs are traced.
- \*BCI Batch immediate jobs are traced.

Top

---

## Job trace interval (JOBTRCITV)

Specifies the time interval (in CPU seconds) to be used between each collection of the job trace data.

0.5 A time slice interval value of 0.5 CPU seconds is used.

*0.1-9.9* Specify the trace interval to be used, in CPU seconds.

Top

---

## Examples

```
STRPFTRC SIZE(*CALC)
```

This command starts the collection of performance trace data. The trace table size may be adjusted to the calculated minimum. This example will result in the same trace table size and data as would STRPFTRMON TRACE(\*ALL) DMPTRC(\*NO).

Top

---

## Error messages

### \*ESCAPE Messages

CPF0A2A

Performance trace already started

Also refer to the TRCINT command for other messages.

Top





---

## Start Program Export List (STRPGMEXP)

Parameters  
Examples  
Error messages

The Start Program Export List (STRPGMEXP) binder definition statement starts a list of exports in a service program export block.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
PGMLVL	Program level	<u>*CURRENT</u> , *PRV	Optional, Positional 1
LVLCHK	Signature level check	* <u>YES</u> , *NO	Optional, Positional 2
SIGNATURE	Signature	<i>Character value</i> , <u>*GEN</u>	Optional, Positional 3

[Top](#)

---

### Program level (PGMLVL)

Specifies the service program level for this export block.

#### \*CURRENT

This service program export block contains exports for the current service program.

**\*PRV** This service program export block contains exports for a previous version of the service program.

[Top](#)

---

### Signature level check (LVLCHK)

Specifies whether a level check is performed on the export block.

**\*YES** The service program export block is level checked by generating a nonzero signature.

**\*NO** The service program export block is not level checked. A zero signature is generated.

[Top](#)

---

### Signature (SIGNATURE)

Specifies the signature for the export block.

**\*GEN** If \*YES is specified for the **Signature level check (LVLCHK)** parameter, the system generates a nonzero signature for the export block. If \*NO is specified for the LVLCHK parameter, this value is required and the system generates a zero signature for the export block.

### *hexadecimal-character-value*

The signature value for the export block is set to the specified string of hexadecimal digits and is not generated by the system. If the specified value is less than 32 hexadecimal digits in length, the system pads it on the left with hexadecimal zeros to 32 digits. If the specified value is greater than 32 hexadecimal digits in length, the system truncates it on the right to 32 digits.

### *character-value*

The signature value for the export block is set to the EBCDIC character codes of the given signature-value and is not generated by the system. If the signature-value is less than 16 characters in length, the system pads the signature-value on the right with spaces to 16 characters. If the signature-value is greater than 16 characters in length, the system truncates it on the right to 16 characters.

[Top](#)

---

## Examples

```
STRPGMEXP  PGMLVL(*CURRENT)  LVLCHK(*YES)  SIGNATURE(*GEN)
```

This binder definition statement marks the beginning of a list of exported variables or procedures for a service program. This service program export block contains exports for the current service program. Level checking will be performed using a signature generated by the operating system.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Start Programmer Menu (STRPGMMNU)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Programmer Menu (STRPGMMNU) command shows the programmer menu. This command can be used instead of the CALL QPGMMENU function, and allows you to pass parameters to specify and control the data which appears in the associated fields on the programmer menu.

### NOTES:

1. A user exit program can be called instead of submitting a job when option 3 is selected.
2. The first four parameters control the defaults that appear when the menu is first displayed.

More information about using the Programmer Menu is in the CL Programming book, SC41-5721.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SRCFILE	Source file	Name, <u>*DFT</u>	Optional, Positional 1
SRCLIB	Source library	Name, <u>*LIBL</u> , *CURLIB	Optional, Positional 2
OBJLIB	Object library	Name, <u>*DFT</u> , *CURLIB	Optional, Positional 3
JOB	Job description	Name, <u>*USRPRF</u>	Optional, Positional 4
ALWUSRCHG	Allow changes	<u>*YES</u> , *NO	Optional
EXITPGM	Option 3 exit program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Option 3 exit program	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
DLTOPT	Delete option	<u>*DLT</u> , *PROMPT, *NODLT	Optional

Top

---

## Source file (SRCFILE)

Specifies an existing source file that contains source file members to be updated or to which new source file members are to be added.

\*DFT This is the default for the type being specified on the menu. This field is blank when shown on the display station.

*name* Specify the name of the source file to be updated.

Top

---

## Source library (SRCLIB)

Specifies the library that is searched for the source file.

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

*name* Specify the name of the library where the source file is located.

Top

---

## Object library (OBJLIB)

Specifies the library that is to contain the object.

**\*DFT** Blanks appear for this field. The library used depends on the menu option you selected.

**\*CURLIB**

The current library for the job is the library that is to contain the object.

*name* Specify the name of the library that is to contain the created object.

Top

---

## Job description (JOBDB)

Specifies the job description used with the job being submitted.

**\*USRPRF**

The job description defined in the user profile of the user running the STRPGMMNU command is used for the job.

*name* Specify the name of the job description used for the job. The job description is found through the library list being used by the job.

Top

---

## Allow changes (ALWUSRCHG)

Specifies whether the menu display fields you specified on the previous parameters in this command can be changed by the user.

**\*YES** Values on the display can be changed.

**\*NO** The display fields cannot be changed.

Top

---

## Option 3 exit program (EXITPGM)

Specifies the user-written program that is called as an exit program in place of submitting a batch job when menu option 3 is selected. When the exit program is called, it receives parameters that are sent by the programmer menu. More information about the EXITPGM parameter is in the CL Programming book, SC41-5721.

### Single values

### \*NONE

No user-written program is called; a batch job is submitted. When \*NONE is specified, \*DLT must be specified for the **Delete option (DLTOPT)** parameter.

### **Qualifier 1: Option 3 exit program**

*name* Specify the name of the program called when option 3 is selected, instead of submitting the create command as a batch job. When a value is specified on this parameter, the text that appears on the menu for option 3 shows the name and library of the exit program.

### **Qualifier 2: Library**

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

### \*CURLIB

The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

*name* Specify the library where the program is located.

Top

---

## **Delete option (DLTOPT)**

Specifies the action to be taken when:

- A program name is specified on the **Option 3 exit program (EXITPGM)** parameter.
- Option 3 of the programmer menu is selected.
- An object of the name and type to be created already exists in the library specified on the menu.

Regardless of the value specified, the system passes a parameter (from among the parameters passed from the Programmer Menu) to the exit program that specifies whether the object exists.

\*DLT This value must be specified if \*NONE is specified on the EXITPGM parameter. If an exit program is specified for the EXITPGM parameter, and the object specified to be created with option 3 exists, and the Enter key is pressed, a message is shown; press the F11 key to proceed. When the F11 key is pressed, the system deletes or replaces the object before calling the program specified by the exit program. This is the normal Programmer Menu function when an exit program is not specified.

### **NOTES:**

1. When the \*DLT value is specified, the object is deleted or replaced before the job is submitted or the user exit program is called.
2. If the source type is one of the following, the object is replaced rather than deleted (an exit program must be used to delete instead of replace):

BAS	C	CBL
CBL36	CLP	DSPF
DSPF36	FTN	ICFF
MNU36	MSGF36	PAS
PLI	PRTF	RPG
RPG36	RPT36	

### \*PROMPT

The system does not delete or replace the object, but you are prompted for approval to delete the object. If the object exists, and the Enter key is pressed, a message is displayed. Press the F11 key to proceed; the system does not delete the object. The user confirms whether the object is deleted or replaced, yet the deletion is still controlled by the exit program.

### \*NODLT

The user exit program is called regardless of the presence of the object.

**Table 1. Figure: Table 1. Actions Taken by the System When an Exit Program Is Called**

DLTOBJ Specified	F11 Required	Object Deleted	Value Passed to Exit Program
If object existed when option 3 was selected			
*DLT	Yes	Yes	1
*PROMPT	Yes	No	0
*NODLT	No	No	0
If object did not exist when option 3 was selected			
*DLT	No	---	2
*PROMPT	No	---	2
*NODLT	No	---	2

Top

## Examples

### Example 1: Displaying Programmer Menu

```
STRPGMMNU
```

This command displays the Programmer Menu with defaults for all parameters. This has the same result as entering CALL QPGMMENU.

### Example 2: Preventing Values from Being Changed

```
STRPGMMNU SRCFILE(YOURFILE) SRCLIB(YOURLIB) OBJLIB(YOURLIB)
JOB(YOURJOB) ALWUSRCHG(*NO)
```

This command prevents the values on the menu from being changed from those specified on the command.

### Example 3: Calling an Exit Program

```
STRPGMMNU EXITPGM(OPT3PGM) DLTOPT(*PROMPT)
```

This command calls user exit program OPT3PGM instead of submitting a batch job when option 3 is specified. If the object already exists, DLTOPT(\*PROMPT) requires the user to press the F11 key; however, the object is not deleted.

### Example 4: Receiving Parameters

The following portion of a CL program is an example of how these parameters would be received by a user exit program. If the specified type is one of those listed, the object is not deleted. The create command with REPLACE(\*YES) specified is passed to the exit program. The value passed to the exit program is 0.

```
PGM  PARM(&OPTION &PARM &TYPE &PARM2 &SRCFILE +
      &SRCLIB &OBJLIB &JOBID &RQSLEN &RQSDTA512 +
      &F4 &F11 &EXIST)
/* The following values are passed in exactly as */
/* they appear on the Programmer Menu.          */
DCL  VAR(&OPTION)  TYPE(*CHAR)  LEN(2)
DCL  VAR(&PARM)    TYPE(*CHAR)  LEN(10)
DCL  VAR(&TYPE)    TYPE(*CHAR)  LEN(10)
DCL  VAR(&PARM2)   TYPE(*CHAR)  LEN(21)
DCL  VAR(&SRCFILE) TYPE(*CHAR)  LEN(10)
DCL  VAR(&SRCLIB) TYPE(*CHAR)  LEN(10)
DCL  VAR(&OBJLIB)  TYPE(*CHAR)  LEN(10)
DCL  VAR(&JOBID)   TYPE(*CHAR)  LEN(10)
/* The following values are derived by QPGMMENU */
/* from the information entered to the above fields */
/* and the F keys.                               */
/* NUMBER OF BYTES OF REQUEST DATA */
DCL  VAR(&RQSLEN) TYPE(*DEC)  LEN(3 0)
/* DATA FOR RRQSDTA PARAMETER OF SBMJOB COMMAND. */
DCL  VAR(&RQSDTA512) TYPE(*CHAR) LEN(512)
/* F4 WAS PRESSED, '1', OTHERWISE '0'. */
DCL  VAR(&F4) TYPE(*CHAR)  LEN(1)
/* F11 WAS PRESSED, '1', OTHERWISE '0'. */
DCL  VAR(&F11) TYPE(*CHAR)  LEN(1)
/* OBJECT EXISTS- '0' OBJECT WAS DELETED- '1'
OR OBJECT DID NOT EXIST -'2'*/
DCL  VAR(&EXIST) TYPE(*CHAR)  LEN(1)
```

Additional information, along with examples of the STRPGMMNU command with the EXITPGM parameter, can be found in the **CL Programming** book in the Information Center.

Top

---

## Error messages

None

Top





---

# Start Program Profiling (STRPGMPRF)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Program Profiling (STRPGMPRF) command starts collection of profiling information of ILE programs or service programs that have been enabled to collect profiling data by specifying PRFDTA(\*COL) on the Change Program (CHGPGM) or Change Service Program (CHGSRVPGM) command, or when the modules were created by a compiler, or by specifying PRFDTA(\*COL) on the Change Module (CHGMOD) CL command. All programs compiled or changed with this option that are active will have profiling information updated until an End Program Profiling (ENDPGMPRF) command is issued.

The profiling information is added to existing profiling information. If this is not desired, the profiling data can be cleared by specifying PRFDTA(\*CLR) on the CHGPGM or CHGSRVPGM command.

## Restrictions:

- This command is shipped with no public (\*EXCLUDE) authority, and QPGMR user profile having use (\*USE) authority to the command.

There are no parameters for this command.

Top

---

## Parameters

None

Top

---

## Examples

STRPGMPRF

This command start the collection of program profiling information.

Top

---

## Error messages

### \*ESCAPE Messages

CPF5CAA

Unexpected error occurred during program profiling.

Top



---

## Start Prestart Jobs (STRPJ)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Start Prestart Jobs (STRPJ) command starts jobs for a prestart job entry in an active subsystem when there are no currently active prestart jobs for the prestart job entry.

This command is valid after an ENDPJ command is complete, or when all prestart jobs have been ended by the system due to an error or were never started during subsystem start up due to STRJOBS (\*NO) on the ADDPJE command. The number of jobs started is determined by the INLJOBS value on the prestart job entry.

### Restrictions:

1. To use this command, you must have:
  - use (\*USE) authority to the subsystem description and execute (\*EXECUTE) authority to the library that contains that subsystem description.
  - use (\*USE) authority to the specified program and execute (\*EXECUTE) authority to the library that contains that program.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	<i>Name</i>	Required, Positional 1
PGM	Program	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

## Subsystem (SBS)

Specifies the name of the active subsystem that contains the prestart job entry.

This is a required parameter.

*name* Specify the name of the active subsystem that contains the prestart job entry.

Top

---

## Program (PGM)

Specifies the name of the program that identifies the prestart job entry. This program name is used to match an incoming request.

This is a required parameter.

### Qualifier 1: Program

*name* Specify the name of the program that identifies the prestart job entry.

### Qualifier 2: Library

**\*LIBL** All libraries in the thread's library list are searched until a match is found.

**\*CURLIB**

The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the name of the program's library.

Top

---

## Examples

```
STRPJ  SBS(SBS1)  PGM(PJLIB/PJPGM)
```

This command starts prestart jobs for prestart job entry PJPGM in subsystem SBS1. Subsystem SBS1 must be active when this command is issued. The number of jobs started is the number specified in the INLJOBS value of prestart job entry PJPGM. The subsystem starts program PJPGM in library PJLIB.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF0921**

Start Prestart Jobs command not allowed now.

**CPF1226**

Start prestart jobs failed.

**CPF1227**

No authority has been granted to use command.

**CPF1317**

No response from subsystem for job &3/&2/&1.

**CPF1351**

Function check occurred in subsystem for job &3/&2/&1.

**CPF1834**

Prestart job entry for program &1 in &2 does not exist.

**CPF1835**

Not authorized to subsystem description.

Top

# Start Printer Emulation (STRPRTEML)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Start Printer Emulation (STRPRTEML) command starts 3270 printer emulation using a binary synchronous communications (BSC) or Systems Network Architecture (SNA) emulation printer device and a printer device file. The STRPRTEML command is used to print host system (System/370 type) information on an iSeries system. It is used when the user is working on an iSeries system and the information is on a System/370 type system.

More information is in the 3270 Device Emulation Support book, SC41-5408.

Top

## Parameters

Keyword	Description	Choices	Notes
EMLCTL	Emulation controller, or	<i>Name</i>	Optional, Positional 1
EMLDEV	Emulation device, or	<i>Name</i>	Optional, Positional 2
EMLLOC	Emulation location	<i>Communications name</i>	Optional, Positional 3
PRTDEV	Print device	<i>Name</i>	Optional
JOB	Job name	<i>Name</i> , <u>*EMLDEV</u>	Optional
ENDBKTEJT	End Bracket eject	<u>*NO</u> , *YES	Optional
PRTFILE	Printer file	<i>Qualified object name</i>	Optional
	Qualifier 1: Printer file	<i>Name</i> , <u>QPEMPRTF</u>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
TIMOUTEJT	Timeout wait eject	Single values: <u>*NONE</u> , *IMMED Other values: <i>Element list</i>	Optional
	Element 1: Minutes	0-99	
	Element 2: Seconds	0-59, <u>0</u>	
DFRPRTOUT	Defer printing spool output	<u>*PRTFILE</u> , *YES, *NO	Optional
SPOOL	Spool output	<u>*PRTFILE</u> , *YES, *NO	Optional
OPNPRTF	Open printer file	<u>*IMMED</u> , *RCVDTA	Optional
NUMCOL	Print positions per line	1-378, <u>*PRTFILE</u>	Optional
NUMLIN	Lines per page	1-255, <u>*PRTFILE</u>	Optional
LPI	Lines per inch	<u>*PRTFILE</u> , 6, 3, 4, 7.5, 7.5, 8, 9	Optional
MSGQ	Message queue	Single values: <u>*DSPDEV</u> , *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	

Keyword	Description	Choices	Notes
CHRSET	Language character set	*SYSVAL, AGB, AGE, AGI, AGM, ALI, ALM, BGB, BGE, BLI, BLM, BRB, BRE, CAB, CAE, CAI, CAM, CLB, CLE, CYB, CSB, CSE, DMB, DMI, DME, DMM, ESB, FAB, FAI, FAE, FAM, FNB, FNI, FQB, FQI, FNE, FNM, ICE, GKB, GNB, GNE, HIB, HNB, HNE, ICB, ICI, ICM, INB, INI, IRB, ITB, ITE, ITI, ITM, JEB, JEI, JKB, JPB, JPE, JUB, KAB, KOB, LTB, LAE, LVB, MKB, MKE, NCB, NCE, NEB, NEI, NEE, NEM, NWB, NWE, NWI, NWM, PKE, PLB, PLE, PRB, PRI, PRE, PRM, RCB, RMB, RME, ROB, ROE, RUB, RUE, SFI, SFM, SGM, SGI, SKB, SKE, SPB, SPE, SPI, SPM, SQB, SQE, SSB, SSL, SWB, SWI, SSE, SSM, SWE, SWM, TAB, THB, THE, TKB, TKE, TRB, TRE, UAE, UKB, UKI, USB, USI, UKE, UKM, USE, USM, VNE, YGI, YGM, *TRNTBL	Optional
SBMJOB	Submit job	*YES, *NO	Optional
JOB	Job description	Qualified object name	Optional
	Qualifier 1: Job description	Name, <u>QBATCH</u>	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
TRNTBLOUT	Outgoing translation table	Single values: *CHRSET Other values: Qualified object name	Optional
	Qualifier 1: Outgoing translation table	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
JOBQ	Job queue	Single values: *JOB Other values: Qualified object name	Optional
	Qualifier 1: Job queue	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
ENDCOND	End emulation conditions	Single values: *NONE Other values (up to 3 repetitions): *DACTLU, *ENDBKT, *UNBIND	Optional
FORMFEED	Acknowledge form feed	*YES, *NO	Optional
EMLCFG	Configuration entry	Name, QEMDFTCFGE, *NONE	Optional

Top

---

## Emulation controller (EMLCTL)

Specifies the name of a BSC controller description or SNA controller description that has attached 3270 printer emulation device descriptions. When this parameter is specified, the printer emulation job uses a 3270 printer emulation device attached to this controller description. The requester must be authorized to the controller and at least one device, and the device must be available.

Either this parameter, the **Emulation device (EMLDEV)** parameter, or the **Emulation location (EMLLOC)** parameter is required.

Top

---

## Emulation device (EMLDEV)

Specifies the name of a BSC or an SNA printer emulation device (EMLDEV(3284, 3286, 3287, 3288, or 3289)) that is used by the printer emulation job to do a type 3270 printer emulation. The user must be authorized to the device, and the device must be available.

Top

---

## Emulation location (EMLLOC)

Specifies the emulation remote location name that describes the location of the 3270 printer emulation devices. This name is defined during device description configuration and refers to the remote location where communication takes place. When this parameter is specified, an available printer emulation device is selected from those referred to by the location. At least one printer emulation device referred to by the location must be available, and the job running emulation must be authorized to use the device.

Either this parameter, the **Emulation controller (EMLCTL)** parameter, or the **Emulation device (EMLDEV)** parameter is required.

Top

---

## Print device (PRTDEV)

Specifies the name of the printer used with this printer device file to print the output. This parameter is ignored if the printer data is spooled. If the output is not spooled and the printer device is being used at the time the job is initiated, the emulation session ends.

**Note:** If a printer name is not specified, the output is sent to the printer device specified on the **Device (DEV)** parameter of the Create Printer File (CRTPRTF) command. This printer device can be displayed by using the Display File Description (DSPFD) command.

Top

---

## Job name (JOB)

Specifies a job name for the printer emulation job. If the EMLDEV parameter is not specified, and a batch job is to be submitted (\*YES is specified for the **Submit job (SBMJOB)** parameter), a job name must be specified. This parameter is ignored when \*NO is specified for the SBMJOB parameter.

### **\*EMLDEV**

The job name is the same as the printer emulation device name.

### *job-name*

Specify a name for this printer emulation job.

Top

---

## End Bracket eject (ENDBKTEJT)

Specifies whether SNA printer emulation should force out the emulation output when an SNA End Bracket (EB) is received from the host system. Emulation output is forced out by closing and then reopening the emulation printer file specified in the PRTFILE parameter. When the emulation printer output is ejected, a page eject is performed. The parameter uses the default value of \*NO for BSC printer emulation.

**\*NO** The emulation output is not forced out when SNA printer emulation receives an End Bracket.

**\*YES** The emulation output is forced out when SNA printer emulation receives an End Bracket. This is done only if the open printer file contains host system data.

Top

---

## Printer file (PRTFILE)

Specifies the printer device file that prints data received from the host system. The printer device file can be spooled or not spooled.

### QPEMPRTF

The standard printer file (which specifies SPOOL(\*YES)) shipped with the emulation program is used as the printer device file.

### *printer-device-file-name*

Specify the name and library of a user-defined printer device file.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the job is used to locate the printer device file. If no library is specified as the current library for the job, QGPL is used.

### *library-name*

Specify the library where the printer device file is located.

Top

---

## Timeout wait eject (TIMOUTEJT)

Specifies whether printer emulation forces out the emulation output when a time-out has occurred while waiting for host system data. The emulation output is forced out only if the open printer file contains host system data. The wait interval can be specified in number of minutes or seconds. If a value is specified for both minutes and seconds, then these values will be added together.

**Note:** When the emulation printer output is ejected, a page eject is also performed.

### \*NONE

The emulation output is not forced out based on a specified timeout period.

### **\*IMMED**

The emulation output is forced out immediately.

### *minutes-seconds*

Specify a time-out wait interval in minutes or seconds or both.

#### **Minutes**

Valid values range from 0 through 99.

#### **Seconds**

Valid values range from 0 through 59.

Top

---

## Defer printing spool output (DFRPRTOUT)

Specifies whether spooled output is printed immediately or is delayed. This parameter is ignored when \*NO is specified for the Spool output (SPOOL) parameter.

### \*PRTFILE

The SCHEDULE value for the printer file controls how the spooled output is printed.



- \*YES Spooled output is printed when the spooled file is closed.
- \*NO Spooled output can be printed before the spooled file is closed. The printed output does not contain all the data sent by the host system until the spooled file is closed. If the printer is not using spooling, this parameter is ignored.

**Note:** Once the printer starts printing output from 3270 device emulation, spooled output from other jobs sharing the printer does not print until the spooled file that is currently printing is complete.

Top

---

## Spool output (SPOOL)

Specifies whether the output data for the printer device file is spooled. If \*NO is specified on this parameter, output is sent to the printer specified on the **Print device (PRTDEV)** parameter. If the output is spooled, it is sent to the output queue specified on the **Spooled output queue (OUTQ)** parameter of the Create Printer File (CRTPRTF) command for the printer file specified on the PRTFILE parameter. The Output Queue value can be displayed by using the Display File Description (DSPFD) command and specifying the printer file name.

**Note:** If \*JOB is specified on the OUTQ parameter, output is sent to the output queue specified on the OUTQ parameter of the job description specified on the JOBID parameter. This value can be displayed by using the Display Job Description (DSPJOBID) command and specifying the job description name.

### \*PRTFILE

The value specified on the **Spool output (SPOOL)** parameter of the Create Printer File (CRTPRTF) command determines whether spooling is performed.

- \*YES The data is spooled.
- \*NO The data is not spooled; it is sent directly to the device and is printed as the output becomes available.

Top

---

## Open printer file (OPNPRTF)

Specifies when the printer file is opened during the SNA 3270 printer emulation session. If the printer data is not spooled, then the printer will be allocated to your job when the printer file is opened. If the printer data is spooled, then the spool writer is allocated to your job after the printer file is opened depending on the value of the DFRPRTOUT parameter.

This parameter is not allowed if specified for BSC 3270 printer emulation.

### \*IMMED

The printer file is opened immediately after starting the 3270 printer emulation session.

### \*RCVDTA

The printer file is opened after first receiving print data from the host system.

Top

---

## Print positions per line (NUMCOL)

Specifies the number of columns in a line when creating the printed output.

### \*PRTFILE

The printer file PAGESIZE(width) value contains the number of columns per line. This value is used if \*IMMED is specified on the open printer file (OPNPRTF) parameter, or \*RCVDTA is specified on the OPNPRTF parameter and the maximum print positions (MPP) value is not sent from the host system. Otherwise, the MPP value sent from the host system is used.

#### *number-of-columns*

Specify the number of columns per line in the printed output. Valid values range from 1 through 378.

Top

---

## Lines per page (NUMLIN)

Specifies the number of lines per page when creating the printed output.

### \*PRTFILE

The printer file PAGESIZE(length) value contains the number of lines per page. This value is used if \*IMMED is specified on the open printer file (OPNPRTF) parameter, or \*RCVDTA is specified on the OPNPRTF parameter and the maximum page length (MPL) value is not sent from the host system. Otherwise, the MPL value sent from the host system is used.

#### *number-of-lines*

Specify the number of lines per page in printed output. Valid values range from 1 through 255.

Top

---

## Lines per inch (LPI)

Specifies the number of lines per inch when creating the printed output.

### \*PRTFILE

The printer file LPI value contains the number of lines per inch. This value is used if \*IMMED is specified on the open printer file (OPNPRTF) parameter, or \*RCVDTA is specified on the OPNPRTF parameter and the set line density (SDL) value is not sent from the host system. Otherwise, the SDL value sent from the host system is used.

#### *lines-per-inch*

Specify the number of lines per inch in the printed output. Valid values are 3, 4, 6, 7.5, 7,5, 8 and 9. Values 3, 7.5 and 7,5 are valid only for double-byte character set (DBCS) printer devices.

Top

---

## Message queue (MSGQ)

Specifies the message queue to which operational messages for this device are sent.

### \*DSPDEV

The current display station message queue is used.

### \*NONE

No messages are sent to message queues other than the job log for the printer emulation job.

The possible library values are:

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the job is used to locate the message queue. If no library is specified as the current library for the job, QGPL is used.

#### *library-name*

Specify the library where the message queue is located.

#### *message-queue-name*

Specify the name and library of the message queue where messages are sent when the printer emulation job is running.

Top

---

## **Language character set (CHRSET)**

Specifies the 3-character country or region keyboard language identifier which represents a specific full character identifier (CHRID - comprised of a character set and code page) that is used by the printer being emulated.

### **\*SYSVAL**

The current QKBDTYPE system value is used.

#### *country-keyboard-identifier*

Specify the country or region keyboard language identifier to be used.

### **\*TRNTBL**

Allows a user-defined translation table to be used. The character translation is defined in the translation table specified by the **Outgoing translation table (TRNTBLOUT)** parameter. The CHRID associated with the QKBDTYPE system value will be used by the printer being emulated.

Top

---

## **Submit job (SBMJOB)**

Specifies whether the printer emulation should be done as a separate job or as part of this job.

**\*YES** A specific job is submitted to do the printer emulation. The job attributes are determined from the job description specified by the **Job description (JOBID)** parameter. The job uses your user profile.

**\*NO** Printer emulation is done in the current job.

Top

---

## **Job description (JOBID)**

Specifies the job description for the job that is being submitted for 3270 printer emulation. This parameter is ignored when \*NO is specified for the **Submit job (SBMJOB)** parameter.

### **QBATCH**

The job description QBATCH is used for the job.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the job is used to locate the job description name. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the library where the job description name is located.

*job-description-name*

Specify the name and library of the job description associated with the job.

Top

---

## Outgoing translation table (TRNTBLOUT)

Specifies the outgoing translation table used to translate characters sent from the host system to 3270 Emulation.

\*CHRSET

Specify that translation is done when data is sent from the host system using the character set specified on the **Language character set (CHRSET)** parameter.

The possible library values are:

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

\*CURLIB

The current library for the job is used to locate the translation table. If no current library entry exists in the library list, QGPL is used.

*library-name*

Specify the library where the translation table is located.

*table-name*

Specifies the name and library of the table which is used for outgoing translation.

Top

---

## Job queue (JOBQ)

Specifies the name of the job queue in which this 3270 printer emulation job is placed. This parameter is ignored when \*NO is specified for the **Submit job (SBMJOB)** parameter.

\*JOBQ

The submitted job is placed in the job queue associated with the job description specified in the (JOBQ) parameter.

*job-queue-name*

Specify the name and library of the job queue to contain the submitted job.

The possible library values are:

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

\*CURLIB

The current library for the job is used to locate the job queue. If no library is specified as the current library for the job, QGPL is used.

*library-name*

Specify the library where the job queue is located.

Top

---

## End emulation conditions (ENDCOND)

Specifies additional ways in which the SNA 3270 printer emulation session can end.

This parameter is not allowed if specified for BSC 3270 printer emulation.

### **\*NONE**

No additional ways to end 3270 printer emulation are requested.

### **\*DACTLU**

The 3270 printer emulation session ends if it receives an SNA DACTLU from the host system.

### **\*ENDBKT**

The 3270 printer emulation session ends if it receives an SNA end bracket from the host system. Please consider the following before selecting this end condition:

- This end condition should be used only when you need to print one host system file for the duration of the session. An end bracket may occur after printing the first file, and the 3270 session ends before a second file can print.

### **\*UNBIND**

The 3270 printer emulation session will end if it receives an SNA UNBIND from the host system. Please consider the following before selecting this end condition:

- This end condition should be used only when you need to print one host system file for the duration of the session. An UNBIND may occur after printing the first file, and the 3270 session will end before a second file can print.

Top

---

## Acknowledge form feed (FORMFEED)

Specifies whether to acknowledge a form-feed instruction located in the first character position of the first print line for a 3270 Information Display System data-stream compatibility (DSC) LU3 printer.

This parameter is ignored for an SNA character string (SCS) LU1 printer.

**\*YES** The form-feed instruction is acknowledged. The print position advances to a new page.

**\*NO** The form-feed instruction is ignored. The print position does not advance to a new page.

Top

---

## Configuration entry (EMLCFGE)

Specifies whether a configuration entry is used for this session. Configuration entries indicate 3270 emulation configuration options. Configuration entries are created with the Add Emulation Configuration Entry (ADDEMLCFGE) command.

### **\*NONE**

No configuration entry is named and the configuration entry defaults are used.

### **QEMDFTCFGE**

The default configuration entry QEMDFTCFGE is used. This entry is shipped with configuration entry defaults, and can be updated with the Change Emulation Configuration Entry (CHGEMLCFGE) command.

### *configuration-entry-name*

Specify the name of the configuration entry to be used. If the configuration entry named does not exist in the configuration file, the configuration entry defaults are used.

---

## Examples

### Example 1: Printing Data to Standard Emulation Printer File

```
STRPRTEML  EMLDEV(HOSTPRT4)
```

This command starts a batch job by accepting data from the HOSTPRT4 device and prints the data to the standard emulation printer file (QPMPRTF). The job is named HOSTPRT4 and runs until the job is canceled. Messages are sent to the current work station message queue.

### Example 2: Emulating a Printer in the Current Job

```
STRPRTEML  EMLDEV(HOSTPRT5) SBMJOB(*NO)
```

This command does printer emulation in the current job by accepting data from the HOSTPRT5 device, and writing the data to the standard emulation printer device file (QPMPRTF). The request is active until it ends through the End Printer Emulation (ENDPRTEML) command, or until the job is canceled.

### Example 3: Printing Output Immediately

```
STRPRTEML  EMLCTL(EMLCTL1) TIMOUTEJT(10)  
           DFRPRTOUT(*NO)  NUMLIN(96)
```

This command starts a batch job by accepting data from the device and printing the data in printer file QPMPRTF. If a timeout of 10 minutes occurs, printer emulation forces out the emulation output. The output prints immediately; the maximum number of lines per page is 96.

---

## Error messages

### \*ESCAPE Messages

#### CPF85EB

3270 device emulation session ended.

#### CPF85ED

Values other than ENDCOND(\*NONE) are not supported.

#### CPF85EE

\*RCVDTA on the OPNPRTF parameter is not supported.

#### CPF8510

Internal error occurred on device &1.

#### CPF8511

Emulation ended by errors on device &2.

#### CPF8512

Emulation ended because device &2 was held.

#### CPF8513

Emulation ended by errors on device &2.

#### CPF8514

Error recovery stopped on device &1.

#### CPF8515

3270 emulation session ended by host.

- CPF8516**  
No match between host and device &2.
- CPF8517**  
Received more than maximum number of fields allowed.
- CPF8518**  
Emulation ended because of internal failure in system.
- CPF8519**  
Function check in 3270 emulation.
- CPF8561**  
Printer emulation not started.
- CPF8564**  
Printer emulation job &3/&2/&1 ended.
- CPF8570**  
Translate of 3270 printer data stream failed.
- CPF8579**  
Cannot open printer file &1 in library &2.
- CPF8580**  
File &1 is not a printer file or has been overridden.
- CPF8582**  
Cannot open printer file &1 again in library &2.
- CPF8583**  
Printer emulation cannot open required file.

Top





# Start Printer Writer (STRPRTWTR)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Start Printer Writer (STRPRTWTR) command starts a spooling writer to the specified printer. The writer, which is a system job, takes spooled files from an output queue and produces (writes) the output on the printer device. This command specifies the name of the printer, the names of the output and message queues used, and the name of the writer.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, its own device, and only one type of writer (print, remote, or diskette) can be active to a single output queue. A writer that has been started can be actively writing output or waiting for a file entry to be put on the output queue. The writer can be changed, held, or canceled if the Change Writer (CHGWTR), Hold Writer (HLDWTR), or End Writer (ENDWTR) command is used. Because each writer runs independently of the job that started it, you can continue doing other work on the system after you start a writer.

Top

## Parameters

Keyword	Description	Choices	Notes
DEV	Printer	Name, *ALL, *SYSVAL	Required, Positional 1
OUTQ	Output queue	Single values: *DEV Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Output queue	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
MSGQ	Queue for writer messages	Single values: *DEV, *REQUESTER Other values: <i>Qualified object name</i>	Optional, Positional 4
	Qualifier 1: Queue for writer messages	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
FORMTYPE	Form type options	Element list	Optional
	Element 1: Form type	Character value, *ALL, *STD, *FORMS	
	Element 2: Message option	*INQMSG, *MSG, *NOMSG, *INFMSG	
FILESEP	File separators	0-9, *FILE	Optional
SEPDRAWER	Drawer for separators	1-255, *DEV, *FILE	Optional
WTR	Writer	Name, *DEV	Optional, Positional 3
AUTOEND	Auto-end options	Element list	Optional, Positional 5
	Element 1: Automatically end writer	*NO, *YES	
	Element 2: If yes, when to end	*NORDYF, *FILEEND	
ALWDRTPRT	Allow direct print	*NO, *YES	Optional
ALIGN	Align page	*FILE, *WTR, *FIRST	Optional

Keyword	Description	Choices	Notes
INIT	Initialize printer	<b>*WTR</b> , <b>*FIRST</b> , <b>*ALL</b>	Optional
FILE	Spooled file	<i>Name</i> , <b>*NONE</b> , <b>*LAST</b>	Optional
JOB	Job name	Single values: * Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, <b>*ONLY</b> , <b>*LAST</b> , <b>*ANY</b>	Optional
JOBSYSNAME	Job system name	<i>Name</i> , <b>*ONLY</b> , <b>*CURRENT</b> , <b>*ANY</b>	Optional
CRTDATE	Spooled file created	Single values: <b>*ONLY</b> , <b>*LAST</b> Other values: <i>Element list</i>	Optional
	Element 1: Creation date	<i>Date</i>	
	Element 2: Creation time	<i>Time</i> , <b>*ONLY</b> , <b>*LAST</b>	
PAGE	Starting page	<i>Integer</i> , <b>*BEGIN</b>	Optional

Top

---

## Printer (DEV)

Specifies the name of the printer device used to print the spooled file. The device must be available before the writer can be started.

The possible values are:

**\*ALL** Start a printer writer for every printer configured on the system.

**\*SYSVAL**

Start a printer writer for the system default printer.

*printer-device-name*

Specify the name that the printer device being started is identified by.

Top

---

## Output queue (OUTQ)

This is a required parameter.

Specifies the name of the output queue from which the writer processes spooled files.

The possible values are:

**\*DEV** Use the default output queue associated with the printer specified on the **Printer** prompt (DEV parameter).

*output-queue-name*

Specify the name of the output queue that the writer processes spooled files from.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the output queue. If no current library entry exists in the library list, QGPL is used.

*library-name*

Specify the library where the output queue is located.

Top

---

## Queue for writer messages (MSGQ)

Specifies the message queue that is used by this user.

The possible values are:

**\*DEV**

The messages are sent to the message queue specified in the printer's device description.

**\*REQUESTER**

The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, \*DEV is used.

*message-queue-name*

Specify the name of the message queue where writer messages should be sent.

The possible library values are:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the message queue. If no current library entry exists in the library list, QGPL is used.

*library-name*

Specify the library where the message queue is located.

Top

---

## Form type options (FORMTYPE)

Specifies which form type should be selected to print. This parameter specifies that only the files with this form type are processed now. All other files are left on the output queue as available. If you wish to change the type of form to be printed after the writer is started, you should use the Change Writer (CHGWTR) command.

**Note:** The form load message is issued when the spooled file to be printed has a form type different from the form type of the last spooled file that was printed on the device. The last form type printed is kept from the last STRPRTWTR, CHGWTR, or VRYCFG command issued.

Consider the following example:

1. The last spooled file printed on printer PRT01 had the form type \*STD.
2. The user changes the form type on PRT01 to XYZ using the following command:  
CHGWTR PRT01 FORMTYPE(XYZ)
3. No spooled file with the form type XYZ is printed on PRT01.
4. The user then sends a spooled file with the form type \*STD to PRT01. The form load message is not issued, despite the intervening CHGWTR command, because the last spooled file printed on PRT01 had the same form type as the spooled file being printed.

The form load message would be issued if a spooled file with the form type XYZ were actually printed on PRT01.

### Element 1: Type of Form Designation

**\*ALL** All available files on the output queue will be processed regardless of their form type.

**\*FORMS**

All available files on the output queue with the same form type are processed as a group before the writer moves on to the next form type. The writer first chooses the first available file on the queue. After the first file is complete, all files with the same form type will be processed. The writer again chooses the first available file on the queue and repeats the process for that form type.

**\*STD** The writer processes spooled files with a form type of \*STD.

*form-type*

Specify the type of form for which you want spooled files processed.

**Element 2: Message Sending Options**

**\*INQMSG**

An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer.

**\*INFOMSG**

An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

**\*MSG** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer and an informational message is sent when no spooled files requiring this form type remain in the output queue.

**\*NOMSG**

Neither an inquiry message nor an informational message is sent to the message queue.

Top

---

## File separators (FILESEP)

Specifies how to control the number of file separator pages that are printed before each file.

The possible values are:

**\*FILE** The number of separators that was specified for each individual file is used.

*number-of-separators*

Specify the number (range of 0 to 9) of separator pages to print. Whenever you respond to the change form type message indicating that a new form type has been put on the printer, the writer issues a message inquiring how many file separator pages should be printed with the new form type.

Top

---

## Drawer for separators (SEPDRAWER)

Specifies which paper drawer is selected for printing job and file separators.

The possible values are:

**\*DEV**

The value stored in the device description for the printer is used.

**\*FILE** The separator pages are printed from the same drawer as the spooled file.

**1** The separator pages are printed from drawer 1.

2 The separator pages are printed from drawer 2.

3 The separator pages are printed from drawer 3.

*separator-drawer*

Specify a value ranging from 1 through 255 to indicate the drawer from which the separator pages are printed.

**Note:** For some printers, SEPDRAWER(3) implies an envelope drawer.

Top

---

## Writer (WTR)

Specifies the name of the spooling writer being started. Each writer name must be unique.

The possible values are:

**\*DEV** The name of the writer is the same as the name of the printer device specified on the **Printer** prompt (DEV parameter).

*writer-name*

Specify the name by which the writer being started is identified.

Top

---

## Auto-end options (AUTOEND)

Specifies whether the writer ends automatically.

The possible values for it to end are:

**\*NO** The writer does not end when the last available file has been removed from the output queue; it waits for another spooled file entry to be put on the queue.

**\*YES** The writer automatically ends after it has reached the state specified by the second part of this parameter.

The possible values for when to end are:

**\*NORDYF**

The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

**\*FILEEND**

The writer stops after it has finished processing one spooled file.

Top

---

## Allow direct print (ALWDRTprt)

Specifies whether the printer writer allows files to be printed directly to the printer. A file printed directly to the printer is created by specifying SPOOL(\*NO) for a printer file. When direct printing is allowed, the non-spooled printer file is printed immediately if the printer is available or, if the printer is busy, the non-spooled printer file waits until the printer is available. The maximum wait is the length of time specified on the WAITFILE parameter on the printer file, after which the job is automatically canceled. The user can cancel a non-spooled printer file only with an End Job (ENDJOB) command.

The possible values are:

- \*NO** The printer does not allow non-spooled printer files to be printed to the device.
- \*YES** The printer can be used to print spooled and non-spooled output. See the Create, Change, or Override Printer File (CRTPRTE, CHGPRTE, or OVRPRTE) command to set the value of the WAITFILE parameter.
- Note:** Nonspoiled files wait up to 30 seconds regardless of whether the value specified on the WAITFILE parameter is less than 30 seconds.

Top

---

## Align page (ALIGN)

Specifies how to control the forms alignment.

The possible values are:

- \*WTR** The writer keeps track of the output that is printed and issues a forms alignment message whenever it determines that forms may need to be aligned.
- \*FILE** The forms alignment message is issued for every file that has \*YES specified for the **Align page** prompt (ALIGN parameter). This option should be taken whenever the automatic forms alignment control provided by the writer does not provide the desired results.
- \*FIRST** The forms alignment message is issued only for the first file printed. No alignment messages are issued when subsequent errors occur on the printer.

Top

---

## Initialize printer (INIT)

Specifies how often to initialize the printer device.

**Note:** This parameter is ignored if TRANSFORM(\*YES) or a user data transform program was specified on the printer device description.

The possible values are:

- \*WTR** The writer initializes the printer device when necessary.
- \*FIRST** The writer initializes the printer device only before the first file is printed, or after a device error occurs.
- \*ALL** The writer initializes the printer device before each file and each copy of the file is printed.

Top

---

## Spoiled file (FILE)

Specifies the name of the first (or only) spoiled file to print on the printer. If several files are available on the output queue, the next file produced is the first one available with the highest priority.

The possible values are:

- \*NONE** No spoiled file name is specified; the first spoiled file that becomes available on the output queue is processed first.

*spooled-file-name*

Specify the name of the spooled file that is the first (or only) file to be written to the printer.

**\*LAST**

The spooled file which was being printed when the writer ended will start to print when the writer is restarted.

Top

---

## Job name (JOB)

Specifies the name of the job that created the spooled file. This parameter is valid only if a spooled file name is specified on the **Spooled file** prompt (FILE parameter).

\*  
- The job from which this Start Printer Writer (STRPRTWTR) command was issued is the job that created the spooled file.

*user-name*

Specify the user name that identifies the user profile under which the job is run.

*job-number*

Specify the system-assigned job number.

*job-name*

Specify the name of the job that created the spooled file. If no job qualifier is given, all of the jobs currently in the system are searched for the simple name of the job.

Top

---

## Spooled file number (SPLNBR)

Specifies the number of the spooled file that is processed first. This parameter is valid only if a spooled file name is specified on the **Spooled file** prompt (FILE parameter).

The possible values are:

**\*ONLY**

Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

**\*LAST**

The spooled file with the highest number and the specified file name is used.

**\*ANY** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

*spooled-file-number*

Specify the number of the specified file from the job on the specified output queue that is to be processed first.

Top

---

## Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

**\*ONLY**

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file create date and time.

**\*CURRENT**

The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

**\*ANY** The job system name is not used to determine which spooled file is used. Use this value when the spooled file create date and time parameter is to take precedence over the job system name when selecting a spooled file.

*job-system-name*

Specify the name of the system where the job that created the spooled file ran.

Top

---

## Spooled file created (CRTDATE)

Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met.

The possible single values are:

**\*ONLY**

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

**\*LAST**

The spooled file with the latest create date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

The possible create date value is:

*spooled-file-create-date*

Specify the date the spooled file was created.

The possible create time values are:

**\*ONLY**

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.

**\*LAST**

The spooled file with the latest create time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.

*spooled-file-create-time*

Specify the time the spooled file was created.

Top

---

## Starting page (PAGE)

Specifies the page number of the first page to print from the first file. This parameter is valid only if a spooled file name is specified on the **Spooled file** prompt (FILE parameter).

The possible values are:



## **\*BEGIN**

The restart page of the spooled file is the first page to print.

**Note:** If this value has not been changed using the Change Spooled File Attribute (CHGSPLFA) command, the file will print in the same manner as it would if the value were \*BEGIN.

## *page-number*

Specify the number of the first page to print. This number must be within the page range of the file to be valid.

Top

---

## **Examples**

```
STRPRTWTR  DEV(QSYSVRT)  OUTQ(QPRINTS)  WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named QPRINTS and prints the output on the printer named QSYSVRT. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

Top

---

## **Error messages**

### \*ESCAPE Messages

#### **CPF0906**

A duplicate job named &3/&2/&1 was found.

#### **CPF1338**

Errors occurred on SBMJOB command.

#### **CPF1764**

Writer already started for device &1.

#### **CPF1842**

Cannot access system value &1.

#### **CPF2115**

Object &1 in &2 type \*&3 damaged.

#### **CPF2207**

Not authorized to use object &1 in library &3 type \*&2.

#### **CPF3303**

File &1 not found in job &5/&4/&3.

#### **CPF3305**

Output queue &1 in library &2 assigned to another writer.

#### **CPF3309**

No files named &1 are active.

#### **CPF3310**

Writer &1 already started.

#### **CPF3330**

Necessary resource not available.

#### **CPF3340**

More than one file with specified name found in job &5/&4/&3.

- CPF3342**  
Job &5/&4/&3 not found.
- CPF3343**  
Duplicate job names found.
- CPF3347**  
Device &1 not found.
- CPF3357**  
Output queue &1 in library &2 not found.
- CPF3362**  
Objects in QTEMP not valid for parameter values.
- CPF3363**  
Message queue &1 in library &2 not found.
- CPF3369**  
Device &1 not printer device.
- CPF336B**  
Not found or not authorized to driver exit program &1 in library &2.
- CPF3418**  
Duplicate file &1 number &2 found in job.
- CPF346A**  
Transform exit program &1 in library &2 not found or user is not authorized.
- CPF346B**  
Driver exit program &1 in library &2 not found or user is not authorized.
- CPF3463**  
Output queue for device &1 not found.
- CPF3464**  
Not authorized to output queue &1 in library &2.
- CPF3478**  
File &1 not found in job &5/&4/&3 on output queue &6 in library &7.

Top

---

## Start Query Management Proc (STRQMPCR)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Query Management Procedure (STRQMPCR) command allows you to run a query management procedure that was saved as a member in a source file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SRCMBR	Source member	<i>Name</i>	Required, Positional 1
SRCFILE	Source file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Source file	<i>Name</i> , <u>QQMPCRSRC</u>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
RDB	Relational database	<i>Simple name</i> , *NONE, *CURRENT	Optional
RDBCNNMTH	Connection Method	*DUW, *RUW	Optional
USER	User	<i>Name</i> , *CURRENT	Optional
PASSWORD	Password	<i>Character value</i> , *NONE	Optional
NAMING	Naming convention	*SYS, *SQL, *SAA	Optional
ALWQRYDFN	Allow information from QRYDFN	*NO, *YES, *ONLY	Optional
CMDSRCFILE	Command source file	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Command source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
CMDSRCMBR	Source member	<i>Name</i> , *FIRST	Optional
ALWDSPLAY	Display screens	*YES, *NO	Optional

Top

---

### Source member (SRCMBR)

Specifies the name of the member in the source that contains the query management procedure to be run. Specify the name of the member.

This is a required parameter.

Top

---

## Source file (SRCFILE)

Specifies the qualified name of the source file that contains the query management procedure to be run.

This is a required parameter.

The possible **source file** values are:

### QQMPRCSRC

The file having the IBM-supplied source file name, QQMPRCSRC, is used.

### *source-file-name*

Specify the name of the source file.

The name of the source file can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

### *library-name*

Specify the name of the library to be searched.

Top

---

## Relational database (RDB)

Specifies the name of the relation database that is accessed during the processing of this command.

### \*NONE

The local database is accessed. If the user is connected to a remote database, the connection is reset to local and remains local until completion of this command.

### **\*CURRENT**

The relational database to which the user is currently connected is accessed.

### *relational-database-name*

Specify the name of the relational database that is accessed. The database must have an entry in the relation database directory.

Top

---

## Connection Method (RDBCNNMTH)

Specifies the connection method to use.

The possible values are:

### \*DUW

Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

**\*RUW** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

Top

---

## User (USER)

Specifies the user name sent to the remote system when starting the conversation.

The possible values are:

### \*CURRENT

The user name associated with the current job is used.

### *user-name*

Specify the user name being used for the application requester job.

Top

---

## Password (PASSWORD)

Specifies the password to be used on the remote system.

The possible values are:

### \*NONE

No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

### *password*

Specify the password of the user name specified on the USER parameter.

Top

---

## Naming convention (NAMING)

Specifies the naming convention used for naming objects.

The possible values are:

\*SYS The system naming convention is used (library-name/object-name).

\*SQL The SQL naming convention is used (database-name.object-name). If NAMING(\*SQL) is specified, CMDSRCFILE(\*LIBL) cannot be specified or allowed as a default value for locating any of the objects specified on other parameters on this command.

\*SAA The SQL naming convention is used (database-name.object-name). If NAMING(\*SAA) is specified, CMDSRCFILE(\*LIBL) cannot be specified or allowed as a default value for locating any of the objects specified on other parameters on this command.

Top

---

## Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query or form information is taken from a query definition (QRYDFN) object when no query management query (QMQRy) or query management form (QMFORM) object can be found using the specified object name. Any information that has to be derived in this way is discarded when the common programming interface (CPI) command in the procedure is completed. No query management objects are created.

The possible values are:

\*NO The information is not taken from a QRYDFN object.

**\*YES** The information is taken from a QRYDFN object when the specified QMQRV or QMFORM object is not found.

**\*ONLY**

Information can be derived only from a QRYDFN object. Query management objects are ignored.

Top

---

## Command source file (CMDSRCFILE)

Specifies the name and library of the command source file that query management uses to run a command procedure. A command procedure can only contain query management set commands which can set application variables as well as query management variables that start with the 'DSQ' value.

The supported DSQ variables are:

- DSQCMTLV
- DSQCONFIRM
- DSQOAUTH
- DSQSDBNM
- DSQSRUN

The possible **command source file** values are:

**\*NONE**

A command source file is not used. The CMDSRCMBR parameter is ignored.

*command-source-file-name*

Specify the name of the command source file.

The name of the command source file can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library to be searched.

Top

---

## Source member (CMDSRCMBR)

Specifies the name of the command source member that query management uses to run a command procedure. A command procedure can only contain query management set commands which set variables that start with the 'DSQ' value.

The possible values are:

**\*FIRST**

The first member is used.

*command-source-member-name*

Specify the name of the command source member.

Top

---

## Display screens (ALWDSPLAY)

Specifies the display mode used. The query management session is set to interactive mode if ALWDSPLAY(\*YES) is specified. If ALWDSPLAY(\*NO) is specified, then the session is set to batch mode. The mode is automatically set to batch when running this command in a batch environment.

The possible values are:

**\*YES** Displays are shown when used in an interactive session. This mode allows you to interact with the query management commands in the procedure.

**\*NO** No displays are shown.

Top

---

## Examples

### Example 1: Running a Query Management Procedure

```
STRQMPC SRCMBR(MYPROC) SRCFILE(RPTLIB/PROCFIL)
```

This command starts the query management procedure stored as the member named MYPROC in the source file named PROCFILE in the RPTLIB library.

### Example 2: Taking Information From QRYDFN Objects

```
STRQMPC SRCMBR(MYPROC) SRCFILE(PROCFIL)  
        ALWQRYDFN(*YES) ALWDSPLAY(*NO)
```

This command starts the query management procedure stored as the member named MYPROC in the first file named PROCFILE in the library list for the job. Query and form information is allowed to be taken, as needed, from QRYDFN objects when the procedure statements are processed. No reports are shown but they can be printed if the user specifies a print request. Objects are replaced without confirmation if confirmation is not requested by the user. The procedure ends with some errors if processing locates a global variable that is not set or if confirmation was requested before replacing objects that already exist.

Top

---

## Error messages

### \*ESCAPE Messages

#### QWM2701

&1 command failed.

#### QWM2703

&1 command ended.

#### QWM2707

\*LIBL not allowed when SQL naming applied.

#### QWM2709

User or password not valid with relational database value.

#### QWM2710

Password value \*NONE only valid with user value \*CURRENT.

#### QWM2712

Character in user name not valid.





## Start Query Management Query (STRQMQR)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Start Query Management Query (STRQMQR) command is used to run a query.

To use this command, you must first identify the query that is to be processed. The query is any single Structured Query Language (SQL) statement in a QMQR object. The SQL statement can also be taken from a query definition (QRYDFN) object when a QMQR object does not exist.

You can show the output on the display, print it, or store it in a database file.

If the SQL statement inside the query does not create an answer-set, then no report or output file is created. This happens if the SQL statement inside a query is not valid or the SQL statement is not a SELECT clause.

If the query contains substitution variables, the SETVAR parameter can be used to set the variables for the query. If prompting is enabled, query management asks the user to provide a value for each variable that was not set.

Top

### Parameters

Keyword	Description	Choices	Notes
QMQR	Query management query	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Query management query	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTPUT	Output	<i>*, *PRINT, *OUTFILE</i>	Optional, Positional 2
QMFORM	Query management report form	Single values: <i>*SYSDFT, *QMQR</i> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Query management report form	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB, *LIBL</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member	<i>*FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
DATETIME	Date and time	<i>*YES, *NO</i>	Optional
PAGNBR	Page numbers	<i>*YES, *NO</i>	Optional
RDB	Relational database	<i>Simple name, *NONE, *CURRENT</i>	Optional
RDBCNMTH	Connection Method	<i>*DUW, *RUW</i>	Optional

Keyword	Description	Choices	Notes
USER	User	Name, *CURRENT	Optional
PASSWORD	Password	Character value, *NONE	Optional
NAMING	Naming convention	*SYS, *SQL, *SAA	Optional
ALWQRYDFN	Allow information from QRYDFN	*NO, *YES, *ONLY	Optional
SETVAR	Set variables	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Variable name	Character value	
	Element 2: Variable value	Character value	

Top

---

## Query management query (QMQRV)

Specifies the name of the query management query (QMQRV) to be run.

This is a required parameter.

### *query-name*

Specify the name of the query to run.

The name of the query can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

### *library-name*

Specify the name of the library where the query is located.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station, printed with the job's spooled output, or directed to a database file.

The possible values are:

**\*** The output produced by the query is formatted with the specified report form and, in interactive mode, sent to the work station that runs the command. If the command is run in batch mode, the output is sent to the default printer used by query management.

### **\*PRINT**

The output produced by the query is formatted with the specified query management form, then sent to the default printer used by query management.

### **\*OUTFILE**

The output produced by the query is written to a database file (table), which is inserted into a collection.

Top

---

## Query management report form (QMFORM)

Specifies which query management report form is to be applied to the answer-set to format the printed or displayed output.

The possible **report form** values are:

### \*SYSDFT

A default report form is created and used for the report that is printed or displayed.

### \*QMORY

The value specified on the **Query management query** prompt (QMORY parameter) is used to locate the report form.

### *report-form-name*

Specify the name of the report form.

The name of the report form can be qualified by one of the following library values:

\*LIBL The library list is used to locate the report form.

### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

### *library-name*

Specify the name of the library to be searched.

Top

---

## File to receive output (OUTFILE)

Specifies the database file that receives the query output. If the file specified does not exist, the system creates it in the specified library as a table in a collection. If the file is created by this function, the authority for users without specific authority is \*EXCLUDE.

### *database-file-name*

Specify the name of the database file that receives the output of the command.

The name of the database file can be qualified by one of the following library values:

\*LIBL All libraries in the job's library list are searched until the first match is found.

### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

### *library-name*

Specify the name of the library to be searched.

Top

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed.

**\*FIRST**

The first member in the file receives the output.

The possible action to take values are:

**\*REPLACE**

The file is cleared before new records are inserted.

**\*ADD** New records are added after any existing record.

Top

---

## Date and time (DATETIME)

Specifies whether the system date and time are printed on the bottom of each page.

The possible values are:

**\*YES** The system date and time are printed on the bottom of each page.

**\*NO** The system date and time are not printed on the bottom of each page.

Top

---

## Page numbers (PAGNBR)

Specifies whether page numbers are printed on the bottom of each page.

The possible values are:

**\*YES** The page numbers are printed on the bottom of each page.

**\*NO** The page numbers are not printed.

Top

---

## Relational database (RDB)

Specifies the name of the relational database that is accessed during the processing of this command.

The possible values are:

**\*NONE**

The local database is accessed. If you are connected to a remote database, the connection is reset to local and remains local until completion of this command. If the connection management method is \*DUW the remote connection is left in a dormant state.

**\*CURRENT**

The relational database to which you are currently connected is accessed.

With \*RUW connection management, if the user is connected to a remote database, \*OUTFILE cannot be specified on the OUPUT parameter.

With \*DUW connection management, if the user is connected to a remote database and OUTPUT(\*OUTFILE) is specified, the connection is set to local for the \*OUTFILE processing and then the remote connection is restored when the STRQMQRV command is completed.

***relational-database-name***

Specify the name of the relational database that is accessed. The database you specify must have an entry in the relational database directory.

With \*RUW connection management, if the relational database specified is a remote database and OUTPUT(\*OUTFILE) is specified, the connection is reset to local for the \*OUTFILE processing and remains local when the STRQMQRy command is completed.

With \*DUW connection management, if the relational database and OUTPUT(\*OUTFILE) is specified, the connection is set to local for the \*OUTFILE processing and then the remote connection is restored upon completion of the STRQMQRy command.

Top

---

## Connection Method (RDBCNNMTH)

Specifies the connection method to use.

The possible values are:

### \*DUW

Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in the disconnection of previous connections.

**\*RUW** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

Top

---

## User (USER)

Specifies the user name sent to the remote system when starting the conversation.

The possible values are:

### \*CURRENT

The user name associated with the current job is used.

### *user-name*

Specify the user name being used for the application requester job.

Top

---

## Password (PASSWORD)

Specifies the password to be used on the remote system.

The possible values are:

### \*NONE

No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

### *password*

Specify the password of the user name specified on the USER parameter.

Top

---

## Naming convention (NAMING)

Specifies the naming convention used for naming objects.

The possible values are:

- \*SYS** The system naming convention is used (database-name/object-name).
- \*SQL** The SQL naming convention is used (database-name.object-name). If NAMING(\*SQL) is specified, the \*LIBL value cannot be specified or allowed to be a default value for locating any of the objects specified on other parameters on this command.
- \*SAA** The SQL naming convention is used (database-name.object-name). If NAMING(\*SAA) is specified, the \*LIBL value cannot be specified or allowed to be a default value for locating any of the objects specified on other parameters on this command.

Top

---

## Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query or form information is taken from a query definition (QRYDFN) object when no query management query (QMQRy) or query management form (QMFORM) object can be found using the specified object name. Any information that has to be derived in this way is discarded when the command completes. No query management object is created.

The possible values are:

- \*NO** Information is not taken from a QRYDFN object.
- \*YES** Information is taken from a QRYDFN object when the specified QMQRy or QMFORM object name is not found.
- \*ONLY** Information is taken only from a QRYDFN object. Query management objects are ignored.

Top

---

## Set variables (SETVAR)

Specifies the variables that are set by query management before the query is run. Up to 50 variables can be set.

### *variable-name*

Specify a variable name. Valid values range from 1 to 30 characters. Because lower-case characters in variable names are changed to upper-case characters when passed to the command processing program, you cannot use this parameter to set values for variables with mixed case names.

### *variable-value*

Specify a variable value. Valid values range from 0 to 55 characters. If you enclose a value in apostrophes, the apostrophes are removed and double apostrophes within the value are condensed to single apostrophes when the value is passed to the command processing program.

Top

---

## Examples

### Example 1: Displaying Query Output

```
STRQMQRy  QMQRy(MYLIB/MYQRy)  QMFORM(FORM1)
```

This command runs query management query MYQRY located in library MYLIB. The library list is searched for form FORM1, which is used for the output sent to the display.

#### Example 2: Taking Information From Either QMQRY or QRYDFN

```
STRQMQR  QMQRY(MYLIB/MYQRY) QMFORM(FORM1) ALWQRYDFN(*YES)
```

This command runs query management query (QMQRY) MYQRY located in library MYLIB. If QMQRY object MYQRY is not found in library MYLIB, the information is taken from query definition (QRYDFN) MYQRY located in library MYLIB. The library list is searched for query management form FORM1 whose information is used to format the output. If QMFORM object FORM1 is not found in the library list, the library list is searched for QRYDFN FORM1, and that information is used to format the output shown on the display.

#### Example 3: Printing Query Output

```
STRQMQR  QMQRY(MYLIB/QUERY1) OUTTYPE(*PRINTER)
```

This command runs query QUERY1 located in library MYLIB. The report is formatted and printed on the printer specified in the printer file associated with the query session.

#### Example 4: Sending Output to an Existing File

```
STRQMQR  QMQRY(*CURLIB/MYQRY)
          OUTPUT(*OUTFILE)  OUTFILE(MYTAB)  OUTMBR(*FIRST *ADD)
```

This command runs the query named MYQRY located in the current library for the user's job. The selected data records are added to the previously created table named MYTAB in collection MYCOL.

#### Example 5: Running a Query Containing Substitution Variables

```
STRQMQR  QMQRY(MYQUERY)
          SETVAR((VAR1 'select * from mytable')
                (VAR2 'where salary > 15000'))
```

This command runs query MYQUERY, which contains only substitution variables, &VAR1 and &VAR2. These two variables contain the entire structured query language (SQL) statement.

#### Example 6: Changing a Variable

```
STRQMQR  QMQRY(QRYNAME) SETVAR((LASTNAME '''Smith'''))
```

This command runs query QRYNAME, setting the variable LASTNAME to the value, 'Smith'.

Top

---

## Error messages

### \*ESCAPE Messages

#### QWM2701

&1 command failed.

#### QWM2703

&1 command ended.

#### QWM2707

\*LIBL not allowed when SQL naming applied.

#### QWM2709

User or password not valid with relational database value.

**QWM2710**

Password value \*NONE only valid with user value \*CURRENT.

**QWM2712**

Character in user name not valid.

[Top](#)



---

## Start Query (STRQRY)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Query (STRQRY) command displays the Query for iSeries main menu.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

STRQRY

This command shows the main Query Utilities menu.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Start QSH (STRQSH)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start QSH (STRQSH) command, as known as QSH, starts the **qsh** shell interpreter.

If run in an interactive job, STRQSH starts an interactive shell session. If a shell session is not already active in the job, then:

1. A new shell session is started and a terminal window is displayed.
2. **qsh** runs the commands from the file `/etc/profile` if it exists.
3. **qsh** runs the commands from the file `.profile` in the user's home directory if it exists.
4. **qsh** runs the commands from the file specified by the expansion of the ENV variable if it exists.

If a shell session is already active in an interactive job, you are reconnected to the existing session. From the terminal window, you can enter shell commands and view output from the commands.

### Using the Terminal Window

The terminal window has two parts:

- an input line for entering commands, and
- an output area that contains an echo of the commands you entered and any output generated by the commands.

The terminal window supports the following function keys:

#### **F3 (Exit)**

Close the terminal window and end the qsh session.

#### **F5 (Refresh)**

Refresh the output area.

#### **F6 (Print)**

Print the output area to a spool file.

#### **F7 (Up)**

Roll output area up one page.

#### **F8 (Down)**

Roll output area down one page.

#### **F9 (Retrieve)**

Retrieve a previous command. You can press this key multiple times to retrieve any previous command. For example, to retrieve the second to last command you entered, press this key two times. You can also select a specific command to be run again by placing the cursor on that command and pressing this key. When the interactive job is running in a double-byte CCSID, this key is not available.

#### **F11 (Toggle line wrap)**

Toggle the line wrap/truncate mode in the output area. In line wrap mode, lines longer than the width of the terminal window are wrapped to the next line. In truncate mode, the portion of a line beyond the width of the terminal window is not shown.

**F12 (Disconnect)**

Disconnect from the **qsh** session. This key only closes the terminal window and does not end the **qsh** session. You can redisplay the disconnected **qsh** session by running STRQSH again.

**F13 (Clear)**

Clear the output area.

**F17 (Top)**

Display top of output area.

**F18 (Bottom)**

Display bottom of output area.

**F19 (Left)**

Shift output area to the left.

**F20 (Right)**

Shift output area to the right.

**F21 (CL command entry)**

Display a command entry window where you can enter CL commands.

Also, you can use SysReq 2 to interrupt the currently running command.

---

## Error messages for STRQSH

**\*ESCAPE Messages****QSH0002**

Error found with QSH session, reason code &1, errno &2.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CMD	Command	Character value, <u>*NONE</u>	Optional

Top

---

## Command (CMD)

Specifies the shell command to be run.

The possible values are:

**\*NONE:**

No command is provided and an interactive session is started.

**command**

A shell command to run. The command can be a maximum of 5000 bytes in length. If a blank or other special characters are used, the command must be enclosed in apostrophes ('). If an apostrophe is intended, two apostrophes must be used (").

**Note:** The case is preserved when lowercase characters are specified.

Top

---

## Examples

None

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### QSH0002

Error found with QSH session, reason code &1, errno &2.

[Top](#)



---

## Start Question and Answer (STRQST)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Question and Answer (STRQST) command shows the main Question & Answer (Q & A) menu. More information is available in the Basic System Operation information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

---

### Parameters

Keyword	Description	Choices	Notes
QSTDB	Q/A database	Name, <u>*SELECT</u>	Optional, Positional 1
LIB	Lib containing Q/A database	Name, <u>*QSTLIB</u>	Optional, Positional 2

Top

---

### Q/A database (QSTDB)

Specifies the Question and Answer (Q & A) database with which to work.

The possible values are:

#### \*SELECT

You are asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

#### *question-database*

Specify the name of the Q & A database with which to work.

Top

---

### Lib containing Q/A database (LIB)

Specifies the name of the library that contains the Q & A database.

The name of the Q & A database can be qualified by one of the following library values:

#### \*QSTLIB

The library containing the specified Q & A database is searched. If \*SELECT is specified on the QSTDB parameter, any Q & A database in any library to which you are authorized can be selected.

#### *library-name*

Specify the name of the library to be searched. If \*SELECT is specified on the QSTDB parameter, any Q & A database in the library to which you are authorized can be selected.

[Top](#)

---

## Examples

STRQST

This command shows the Question and Answer (Q & A) main menu.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Start REXX Procedure (STRREXPRC)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start REXX Procedure (STRREXPRC) command calls the interpreter, explicitly specifying the library, file, and source member to interpret.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SRCMBR	Source member	<i>Name</i>	Required, Positional 1
SRCFILE	Source file	<i>Qualified object name</i>	Optional
	Qualifier 1: Source file	<i>Name</i> , <u>QREXSRC</u>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
PARM	Parameters	<i>Character value</i> , *NONE	Optional
CMDENV	Command environment	Single values: *COMMAND, *CPICOMM, *EXECSQL Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Command environment	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
EXITPGM	Exit program	Single values: *NONE Other values (up to 8 repetitions): <i>Element list</i>	Optional
	Element 1: Program	<i>Qualified object name</i>	
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 2: Exit code	2, 3, 4, 5, 7, 8, 9, 10	

Top

---

### Source member (SRCMBR)

Specifies the name of the source file member that contains the REXX procedure to be started.

This is a required parameter.

Top

---

### Source file (SRCFILE)

Specifies the source file that contains the REXX procedure to be started.

The possible **source file** values are:

## QREXSRC

The IBM-supplied source file, QREXSRC, contains the REXX procedure.

### *source-file-name*

Specify the name of the source file that contains the REXX procedure to be run.

The possible library values are:

\*LIBL The library list is used to locate the program.

### \*CURLIB

The current library is used to locate the program. If no library is specified as the current library for the job, QGPL is used.

### *library-name*

Specify the library name.

Top

---

## Parameters (PARM)

Specifies procedure parameter values passed to the REXX procedure when it is started. These values are accessed through the argument (ARG) instruction within the REXX procedure.

The possible values are:

### \*NONE

There are no procedure parameters for the REXX procedure. The ARG instruction returns a null string.

### *procedure-parameters*

Specify the procedure parameter value to pass to the REXX procedure. A maximum of 3000 characters is allowed.

Top

---

## Command environment (CMDENV)

Specifies the initial command environment program to process commands embedded in the REXX procedure. The REXX interpreter will call this environment whenever a command is encountered within the REXX procedure.

The possible values are:

### \*COMMAND

The AS/400 control language (CL) command environment is used.

### \*CPICOMM

The Common Programming Interface (CPI) for Communications command environment is used.

### \*EXECSQL

The Structured Query Language (SQL) Command environment is used. EXECSQL is the command environment used for CL commands that are imbedded within a REXX procedure.

### *program-name*

Specify the name of the program to process commands embedded in the REXX procedure.

The possible library values are:

\*LIBL The library list is used to locate the program.

### **\*CURLIB**

The current library for the job is used to locate the program. If no library is specified as the current library for the job, QGPL is used.

### *library-name*

Specify the name of the library where the program is located.

Top

---

## **Exit program (EXITPGM)**

Specifies exit programs to be used when the interpreter is called. A maximum of 8 program and exit code pairs can be specified.

**Note:** If multiple system exit codes are specified, the last one specified is the one taken.

The possible values are:

### **\*NONE**

There are no exit programs for this call.

### *program-name*

Specify a program name. A maximum of 8 program names can be specified.

The possible library values are:

**\*LIBL** The library list is used to locate the programs.

### **\*CURLIB**

The current library for the job is used to locate the programs. If no library is specified as the current library for the job, QGPL is used.

### *library-name*

Specify the name of the library where the programs are located.

### *exit-code*

Specify an exit code. A maximum of 8 exit codes can be specified.

### **Exit-code**

#### **Description**

- |   |   |
|---|---|
| 2 | The associated program is called whenever an external function or subroutine has been called by the REXX program. The exit program is then responsible for locating and calling the requested routine.                |
| 3 | The associated program is called whenever the interpreter is going to call a command. The exit program is responsible for locating and calling the command given the command string and the current environment name. |
| 4 | The associated program is called whenever a REXX instruction or function attempts an operation on the REXX external data queue.   |
| 5 | The associated program is called when session input or output operations are attempted.   |
| 7 | The associated program is called after running each clause of the REXX procedure to determine whether it should be stopped.   |
| 8 | The associated program is called after running each clause of the REXX program to check if tracing should be turned on or off.  |

- 9 The associated program is called before interpretation of the first instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).
- 10 The associated program is called after interpretation of the last instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

Top

---

## Examples

```
STRREXPRC SRCMBR(ABC)
```

This command calls the REXX interpreter instructing it to run the source member named ABC in the first QREXSRC source file in the library list (\*LIBL).

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF7CFB**

Error occurred while processing REXX exit programs.

#### **CPF7CFD**

Error occurred running REXX procedure &1.

#### **CPF7CFF**

REXX procedure &1 ended; return code &4.

#### **CPF7CF2**

REXX procedure &1 not found.

#### **CPF7CF3**

Not authorized to source file &2.

#### **CPF7CF4**

Cannot allocate REXX procedure &1.

#### **CPF7CF6**

Cannot allocate REXX source file &2.

#### **CPF7CF7**

REXX external data queue is damaged.

Top

---

## Start Remote Support (STRRMTSPT)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Remote Support (STRRMTSPT) command creates and varies on all configuration objects needed for remote support. Remote support allows the IBM service organization to access your system. Remote support options that are available include:

- remote work station
- virtual device
- virtual device over a systems network architecture (SNA) connection
- virtual device over a point-to-point protocol (PPP) connection using internet protocol (IP)
- virtual device over a virtual private network (VPN) connection using internet protocol (IP)

Each remote support option has different command parameters that are required for connectivity. If any existing remote support configuration objects are found, they are deleted and then re-created. After the configuration objects have been created, they are varied on. You must provide a user identifier and password before the support person can sign-on your system.

### Restrictions:

1. When using DEVCLS(\*RMT), the remote work station used by a support organization must be one of those listed in the DSPTYPE and DSPMODEL parameters. If your support organization has a configuration that does not match, you must work with the support person to create the correct configuration objects on your system.
2. When using DEVCLS(\*VRT) or DEVCLS(\*IPS), the QUSER user profile must not be disabled.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DEVCLS	Device class	*RMT, *VRT, *IPS, *PPP, *VPN	Required, Positional 1
DSPTYPE	Display type	3179, 3180, 3196, 3197, 5251, 5291, 5292	Optional, Positional 2
DSPMODEL	Display model	1, 2, 11, A1, A2, B1, B2, C1, C2, D1, D2, W1, W2, 0001, 0002, 0011	Optional, Positional 3
STNADR	Station address	X'01'-X'FE', <u>FE</u>	Optional, Positional 4
USRPRF	User profile	Character value, <u>QPGMR</u>	Optional, Positional 5
RSRCNAME	Resource name	Character value, <u>*DFT</u>	Optional, Positional 6
RMTLOCNAME	Remote location	Communications name, <u>QREMOTE</u>	Optional, Positional 7
LCLLOCNAME	Local location	Communications name, <u>QLOCAL</u> , *NETATR	Optional, Positional 8
RMTNETID	Remote network identifier	Communications name, *NETATR, <u>*NONE</u>	Optional

Keyword	Description	Choices	Notes
MAXLENRU	Maximum length of request unit	241-32767, <u>32767</u> , *SAME, *CALC	Optional
DTACPR	Data compression	*SAME, *NETATR, *NONE, *ALLOW, <u>*REQUEST</u> , *REQUIRE	Optional
INDTACPR	Inbound data compression	*SAME, *RLE, *LZ9, *LZ10, <u>*LZ12</u> , *NONE	Optional
OUTDTACPR	Outbound data compression	*SAME, *RLE, *LZ9, *LZ10, <u>*LZ12</u> , *NONE	Optional
MODEM	Modem	<i>Character value</i> , <u>*RSRCNAME</u> , *RMTPPP, *SELECT	Optional
MDMRMTSYS	Modem remote system	<i>Character value</i>	Optional

Top

---

## Device class (DEVCLS)

Specifies the device class for this display station. Different configuration objects are created, depending on the value specified.

This is a required parameter.

- \*RMT** This device class is for a device connected to a remote work station. The configuration objects created for this option include a line description, a controller description, a display device description, and a printer device description.
- \*VRT** This device class is for a virtual device. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.
- \*IPS** This device class is for a virtual device and also supports internet protocol (IP) over SNA sessions on this connection. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.
- \*PPP** This device class supports IP over a point-to-point protocol (PPP) connection. The configuration objects created for this option include a line description, a controller description, a device description, and a PPP profile. DEVCLS(\*PPP) requires that the QRETSVRSEC system value be set to '1' to retain server security data. This allows for additional authentication to be performed when the service organization attempts to connect.
- \*VPN** This device class supports IP over a virtual private network (VPN) connection. The Universal Connection wizard must be run before running the STRRMTSPT command in order to set up a VPN configuration on the system. Once the STRRMTSPT command is entered, eleven alphanumeric characters are displayed on your screen. These characters need to be told to the Customer Support Representative for a complete connection to be established. No configuration objects are created for this option.

Top

---

## Display type (DSPTYPE)

Specifies the workstation display device type which will be used for remote support. A value must be specified for this parameter when DEVCLS has a value of \*RMT, \*VRT, or \*IPS.

Valid display station type values are:

- 3179
- 3180

- 3196
- 3197
- 5251
- 5291
- 5292

Top

---

## Display model (DSPMODEL)

Specifies the model number of the device for this description. A value must be specified for this parameter when DEVCLS has a value of \*RMT, \*VRT, or \*IPS.

The possible values for the device model for each device type are:

**TYPE**    **MODEL**

**3179**    2

**3180**    2

**3196**    A1, A2, B1, or B2

**3197**    C1, C2, D1, D2, W1, or W2

**5251**    11

**5291**    1, or 2

**5292**    1, or 2

Top

---

## Address (STNADR)

Depending on the value specified for the DEVCLS parameter, this parameter specifies the address that will be used for either the station address or the local IP address qualifier. If the DEVCLS is not \*PPP or \*VPN, then this parameter specifies the station address. The station address is the hexadecimal address by which the local system is known to the remote system. The hexadecimal address is the polling address assigned to this system.

If the DEVCLS is \*IPS, \*PPP, or \*VPN, this parameter is used as the local IP address qualifier. This specifies the value of the last byte of the local internet address that will be used.

**FE**        The hexadecimal value FE is the local system address.

**address**

Specify a hexadecimal value from 01 to FE.

Top

---

## User profile (USRPRF)

Specifies the name of the user profile that the IBM service personnel will use to sign on to your system. This profile is made the owner of the objects created by the system when remote support is started. The user profile must already exist on your system.

**QPGMR**

The default system-supplied user profile, QPGMR, is used to sign on to your system.

*user-profile-name*

Specify the name of the existing user profile that will be used to sign on to the remote system.

Top

---

## Resource name (RSRCNAME)

Specifies the name of the resource used to access iSeries electronic customer support. This parameter is not used for DEVCLS(\*VPN).

### \*DFT

- For DEVCLS(\*PPP):
  - Look for resources being used by the 2771 or 2793 integrated modem. If only one 2771 or 2793 is defined, that resource is used for this PPP line. Note that more than one 2771 or 2793 could be defined, but a 2771 or 2793 resource can only be calculated if only one is defined.
  - If a 2771 or 2793 modem resource cannot be used, determine if any resources are defined for use by electronic customer support (ECS). If an ECS resource is available, that resource is used for this PPP line.
  - If neither a single 2771 or 2793 integrated modem or ECS resource is available, the resource cannot be calculated and it will have to be explicitly specified.
- For all other DEVCLS values (except \*VPN): The resource name associated with the shipped default port for accessing electronic customer support is used. This is the first port on the I/O (input/output) adapter in card position B of the first multifunction IOP (input/output processor) on the bus. If this port does not exist on the system, resource name CMN01 is used.

*resource-name*

Specify the name of the resource used to access electronic customer support.

Top

---

## Remote location (RMTLOCNAME)

Specifies the remote location name of the system with which this object communicates. This parameter is used when DEVCLS is \*VRT or \*IPS.

### QREMOTE

The default system-supplied remote support location name.

*remote-location-name*

Specify the name of the remote support location.

Top

---

## Local location (LCLLOCNAME)

Specifies the unique location name that identifies the local system to remote devices. The name cannot be the same as that specified for the **Remote location** prompt (RMTLOCNAME parameter). The combination of the names specified for the **Local location** prompt (LCLLOCNAME parameter) and the **Remote location** prompt (RMTLOCNAME parameter) must be unique for each device attached to the same controller. This parameter is used when the DEVCLS parameter has a value of \*VRT or \*IPS.

### QLOCAL

The default system-supplied local location name is used.

### \*NETATR

The LCLLOCNAME value specified in the system network attributes is used.



### *local-location-name*

Specify the location name to be used to identify the local system to remote devices.

Top

---

## Remote network identifier (RMTNETID)

Specifies the name of the remote network in which the remote system resides. This parameter is used when the DEVCLS parameter has a value of \*VRT or \*IPS.

### \*NONE

No remote network name is used.

### *remote-network-name*

Specify the remote network name.

Top

---

## Maximum length of request unit (MAXLENRU)

Specifies the maximum request unit (RU) length allowed. This parameter is used when the DEVCLS parameter has a value of \*VRT or \*IPS.

32767 The maximum request unit length is 32767 bytes.

### \*CALC

The system calculates the maximum value to use.

### *maximum-request-unit-length*

Specify a value, ranging from 241 through 32767 bytes, for the maximum length of incoming request units.

Some common values, based on line type, are:

- SDLC lines: 256, 512, 1024, 2048
- Token-Ring Network lines: 256, 512, 1024, 1985
- X.25 (QLLC) lines: 247, 503, 1015
- X.25 (ELLC) lines: 241, 497, 1009

More information on setting the maximum RU length is in the Communications Configuration book, SC41-5401 book.

Top

---

## Data compression (DTACPR)

Specifies whether data compression is used.

### \*REQUEST

Data compression is requested on the session by the local system. However, the request can be refused or changed to lower compression levels by the remote system. Data compression is allowed on the session if requested by the remote system. The requested compression levels for inbound and outbound data are the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

**\*NONE**

Compression is not allowed on the session.

**\*ALLOW**

Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

**\*REQUIRE**

Data compression is required on the session. If the remote system does not accept the local system's exact required levels of compression, the session is not established.

The data compression levels that the local system require are the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

Top

---

## Inbound data compression (INDTACPR)

Specifies the desired level of compression for inbound data. No data compression occurs if \*NONE is specified on the **Data compression** prompt (DTACPR parameter). Adaptive dictionary-based compression is a dynamic compression algorithm, similar to Lempel-Ziv, that compresses previously seen strings to 9-, 10-, and 12-bit codes. This algorithm is referred to as LZ in the following parameters.

- \*LZ12** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.
- \*RLE** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.
- \*LZ9** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.
- \*LZ10** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

Top

---

## Outbound data compression (OUTDTACPR)

Specifies the desired level of compression for outbound data. No data compression occurs if \*NONE is specified on the **Data compression** prompt (DTACPR parameter).

- \*LZ12** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.
- \*RLE** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.
- \*LZ9** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.
- \*LZ10** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

Top

---

## Modem (MODEM)

Specifies the modem description to use for the point-to-point protocol (PPP) profile.

### \*RSRCNAME

The modem description will be determined based on the value defined for the **RSRCNAME** parameter.

- If the resource is defined to use the 2771 integrated modem, the '2771 Internal Modem' description is used.
- If the resource is defined to use the 2772 integrated modem, the '2772 Internal Modem' description is used.
- If the resource is defined to use the 2793 integrated modem, the '2793 Internal Modem' description is used.
- If the resource is defined to use the 2805 integrated modem, the '2805 Internal Modem' description is used.
- If the ECS resource was chosen, the 'IBM 7852-400' modem description is used.
- Otherwise if the resource does not have a pre-defined modem description, **MODEM(\*RSRCNAME)** cannot be used and the modem description will have to be explicitly defined.

### **\*RMTPPP**

Indicates that a modem on a different partition or system is to be used. The internet address or host name where the modem is located must be specified for the **Modem remote system (MDMRMTSYS)** parameter.

### **\*SELECT**

A list of modem descriptions is shown from which you can select the modem to use. This option is only valid when running the **STRRMTSPT CL** command in interactive mode, otherwise an

error will occur. If you are running interactively, it is recommended that you use the \*SELECT value to help ensure that you properly select the modem to use.

**'modem-identification'**

Specify the name of the modem to use. Note that modem names are case sensitive and must match exactly to the modems defined for the system.

**'generic\*-modem-identification'**

Specify the generic name of the modem you wish to use. A generic modem name is a character string of one or more characters followed by an asterisk (\*); for example, 'abc\*'. If a generic name is specified, then the FIRST modem name that matches with the generic name will be used. It is recommended that you include as many characters in the modem name string as possible to avoid any ambiguity. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete modem name. The actual modem name chosen will be posted in a message in the joblog.

Top

---

## Modem remote system (MDMRMTSYS)

Specifies the internet protocol (IPv4) address or the host name for the system or partition where the modem to be used for remote support is located.

**Note:** This parameter is required if \*RMTPPP is specified for the **Modem (MODEM)** parameter. If the MODEM parameter has any value other than \*RMTPPP, this parameter cannot be specified.

**character-value**

Specify the IP address or host name of the remote system or partition where the modem is located.

Top

---

## Examples

### Example 1: Start Remote Support using PPP

```
STRRMTSPT  DEVCLS(*PPP)  STNADR(FA)
```

This command creates and starts a PPP answer profile and associated configuration objects. The local internet address assigned is 169.254.2.250.

### Example 2: Start Remote Support for Virtual Device

```
STRRMTSPT  DEVCLS(*VRT)  DSPTYPE(5251)
           DSPMODEL(11)  STNADR(FE)
```

This command creates and varies on the 5251 Model 11 Display Station located at station address FE.

### Example 3: Start Remote Support using VPN

```
STRRMTSPT  DEVCLS(*VPN)
```

This command starts remote support over a virtual private network (VPN) connection. Before running this command, the Universal Connection wizard needs to be run, specifying a VPN or multihop type of connection (local or remote) for Electronic Customer Support (ECS).

### Example 4: Start Remote Support using a Remote Modem and PPP

```
STRRMTSPT  DEVCLS(*PPP)  STNADR(FE)  MODEM(*RMTPPP)
           MDMRMTSYS('10.1.1.2')
```

This command creates and starts an L2TP remote answer profile and associated configuration objects. Before running this command, the Universal Connection wizard needs to be run on the system or partition with internet address 10.1.1.2 where the modem to be used is located. When running the Universal Connection wizard, you need to specify that the modem should provide connectivity to others and that the connection type is AT&T. An alternative would be to create an L2TP terminator profile using the Remote Access Service New Profile GUI on the system or partition where the modem is located, specifying that outgoing calls are supported.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF1394**

CPF1394 User profile &1 cannot sign on.

#### **CPF9801**

Object &2 in library &3 not found.

#### **CPF9899**

Error occurred during processing of command.

#### **TCP83A7**

QRETSVRSEC=0. Unable to save remote service password for &2.

Top



# Start Remote Writer (STRRMTWTR)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Start Remote Writer (STRRMTWTR) command starts a spooling writer that sends spooled files from an output queue to a remote system. The writer, which is a system job, takes spooled files from an output queue and sends them to a remote system using SNADS or TCP/IP.

After the spooled file is successfully sent to a remote system, the spooled file will be deleted or saved, as specified by the SAVE spooled file attribute.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, and only one type (print, remote, or diskette) of writer can be active to a single output queue. A writer that has been started can be actively sending output or waiting for a file to be put on the output queue. Optionally, the writer can end automatically when it has processed all the files on the output queue. You can also change, hold, or cancel the writer.

You can continue with other work after starting a writer because each job runs independently. The writer is owned by the user who issues the STRRMTWTR command.

Top

## Parameters

Keyword	Description	Choices	Notes
OUTQ	Output queue	Single values: *ALL Other values: <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Output queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MSGQ	Queue for writer messages	Single values: *OUTQ, *REQUESTER Other values: <i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Queue for writer messages	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
FORMTYPE	Form type options	<i>Element list</i>	Optional
	Element 1: Form type	<i>Character value</i> , *ALL, *STD, *FORMS	
	Element 2: Message option	*NOMSG, *INQMSG, *MSG, *INFOMSG	
WTR	Writer	<i>Name</i> , *OUTQ, *SYSGEN	Optional, Positional 2
AUTOEND	Auto-end options	<i>Element list</i>	Optional, Positional 4
	Element 1: Automatically end writer	*NO, *YES	
	Element 2: If yes, when to end	*NORDYF, *FILEEND	

Top

---

## Output queue (OUTQ)

Specifies the qualified name of the output queue.

The possible values are:

**\*ALL** Starts remote writers for every output queue on the system which have a remote system specified. No writer starts if an output queue has a different type of writer already started. The number of "writers to autostart" attribute in the output queue determines the number of writers started to a particular queue. This will determine how many remote writer jobs will be sending spooled output to a remote system, from a single output queue.

The user-defined value can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*library-name*

Specify the name of the library to be searched.

The possible user-defined value is:

*output-queue-name*

Specify the name of the output queue from which the writer processes output files.

Top

---

## Queue for writer messages (MSGQ)

Specifies the qualified name of the message queue to which messages created by the writer are sent.

**\*OUTQ**

Messages are sent to the message queue of the output queue specified on the OUTQ parameter.

**\*REQUESTER**

The messages are sent to the workstation message queue of the workstation of the user who started the writer. If this value is specified for a batch job, \*OUTQ is used.

The user-defined value can be qualified by one of the following library values:

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*library-name*

Specify the name of the library to be searched.

The possible user-defined value is:

*message-queue-name*

Specify the name of the message queue to which messages created by the writer are sent.

Top



---

## Form type options (FORMTYPE)

Specifies the name of the form type. The writer uses this value to select spooled files from the specified output queue for sending to a remote system.

**Note:** A spooled file's form type is specified in the device file that produced the spooled file.

The possible Type of Form Designation values are:

**\*ALL** All form types are processed by the writer.

**\*FORMS**

The writer first chooses the first available spooled file on the output queue. After the first spooled file is complete, all spooled files with the same form type are processed. The writer then chooses the first available spooled file on the output queue and repeats the process for that form type.

**\*STD** Only spooled files that specify the standard form type are selected.

*form-type*

The form type of the spooled files being produced.

The possible Message Sending Options values are:

**\*NOMSG**

Neither an inquiry message nor an informational message is sent to the message queue.

**\*INQMSG**

An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type last sent.

**\*INFOMSG**

An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

**\*MSG** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the remote and an informational message is sent when no spooled files requiring this form type remain in the output queue.

Top

---

## Writer (WTR)

Specifies the name of the spooling writer being started. Each writer name must be unique.

The possible values are:

**\*OUTQ**

The name of the writer is the same as that of the output queue specified on the OUTQ parameter. If OUTQ(\*ALL) is specified, and you have more than one writer to start, the name for each additional writer is taken from the first nine characters of the output queue followed by a digit. This last digit is sequentially assigned, starting with 2 for the second writer, 3 for the third writer, and continuing through 0 for the tenth writer.

**\*SYSGEN**

The writer name is generated by the system, starting with 'RMTW000001' and incrementing the numeric part for each successive writer.

*writer-name*

Specify the name by which the writer being started is identified.

Top

---

## Auto-end options (AUTOEND)

Specifies whether the writer ends automatically.

Element 1: Stop Writer Option

**\*NO** The writer does not end when the last available file has been removed from the output queue. It waits for another spooled file entry to be put on the queue. This is a single value (Element 2 is not specified).

**\*YES** The writer automatically ends after it has reached the condition specified on the second element of this parameter.

Element 2: Conditions for Stopping Writer

**\*NORDYF**

The writer automatically ends when there are no ready files (all the available files have been removed from the output queue).

**\*FILEEND**

The writer ends after it finishes processing one spooled file.

Top

---

## Examples

```
STRRTWTR  OUTQ(RMTOUTQ)  WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named RMTOUTQ and sends the output to the remote system and printer queue specified in the output queue RMTOUTQ. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF330A**

Output queue &1 has RMTSYS specified as \*NONE.

**CPF3305**

Output queue &1 in library &2 assigned to another writer.

**CPF3310**

Writer &1 already started.

**CPF3357**

Output queue &1 in library &2 not found.

**CPF3362**

Objects in QTEMP not valid for parameter values.

**CPF3363**

Message queue &1 in library &2 not found.

Top

---

## Start S/36 Session (STRS36)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start System/36 (STRS36) command starts a System/36 environment session (if one is not active already). Even if the System/36 environment is active, this command allows the user to show a menu or run a program or procedure before showing a menu.

When the command ends, the System/36 environment returns to the active or inactive state from which this command is issued.

**Restrictions:** This command cannot be used if a System/36 procedure is already in process. This command cannot be placed in a procedure or in a program that is started by a procedure.

Top

---

### Parameters

Keyword	Description	Choices	Notes
MENU	Menu	Name, <u>*SAME</u>	Optional, Positional 1
CURLIB	Current library	Name, <u>*SAME</u>	Optional, Positional 2
PRC	Procedure or program	Name, <u>*NONE</u>	Optional, Positional 3
FRCMNU	Force menu	<u>*NO</u> , *YES	Optional

Top

---

### Menu (MENU)

Specifies the first menu that is shown when the System/36 environment is started.

The possible values are:

\*SAME

The menu specified in the job does not change. If no menu is specified in the job, the initial menu specified in the user profile is shown.

*menu-name*

Specify the name of the first menu shown when the System/36 environment is started.

Top

---

### Current library (CURLIB)

Specifies the current library to use in the System/36 environment.

The possible values are:

### \*SAME

The current library does not change. If the current library is \*CRTDFT and \*SAME is specified, the current library is set to #LIBRARY.

### *library-name*

Specify the name of the library you want to use as the current library in the System/36 environment.

Top

---

## Procedure or program (PRC)

Specifies the name of the procedure or program to run before the menu is shown.

The possible values are:

### \*NONE

No procedure or program runs.

### *procedure-name*

Specify the name of the procedure or program to run.

Top

---

## Force menu (FRCMNU)

Specifies whether a menu is shown and what this command does in an active System/36 environment.

The possible values are:

### \*NO

The specified menu is not shown if the System/36 environment is active when this command is issued. This command does nothing, and the user is returned to the point at which the command was issued.

### \*YES

The specified menu is shown even if the System/36 environment is active when this command is issued. The current library is set and the program or procedure is run as specified in this command.

Top

---

## Examples

```
STRS36 MENU(USER) CURLIB(MYLIB) PRC(INITPROC)
```

This command starts a System/36 environment session. This command:

- Displays the USER menu
- Changes the current library to MYLIB
- Runs procedure INITPROC before showing the USER menu

Top

---

## Error messages

### \*ESCAPE Messages

**CPF3709**

Tape devices do not support same densities.

**CPF3738**

Device &1 used for save or restore is damaged.

**CPF3767**

Device &1 not found.

[Top](#)



---

## Start S/36 Procedure (STRS36PRC)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start System/36 Procedure (STRS36PRC) command starts a System/36 procedure. It is valid whether or not the System/36 Environment is active, but it is not valid if a System/36 procedure is already running. It cannot be placed in a procedure or in a program that is called by a procedure.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PRC	Procedure	<i>Name</i>	Required, Positional 1
CURLIB	Current library	<i>Name</i> , <b>*SAME</b>	Optional, Positional 2
PARM	Procedure parameters	<i>Character value</i>	Optional, Positional 3

Top

---

### Procedure (PRC)

Specifies the name of the System/36 procedure to run. The procedure is a member of source physical file QS36PRC. The library search order for locating QS36PRC is:

1. current library (\*CURLIB)
2. #LIBRARY
3. job library list (\*LIBL)

This is a required parameter.

Top

---

### Current library (CURLIB)

Specifies the current library to use to run the System/36 procedure.

#### \*SAME

The current library does not change. If the current library is \*CRTDFT and \*SAME is specified, the current library is set to #LIBRARY.

#### *library-name*

Specify the name of the library you want to use for the current library while running the System/36 procedure.

Top

---

## Procedure parameters (PARM)

Specifies procedure parameters for the procedure. Procedure parameters allow information to be passed to the procedure. If no parameters are specified, no parameters are passed to the procedure.

[Top](#)

---

## Examples

### Example 1: Changing the Current Library

```
STRS36PRC  PRC(PROC1)  CURLIB(MYLIB)
```

This command changes the current library to MYLIB and runs procedure PROC1.

### Example 2: Listing Files Used by the System

```
STRS36PRC  PRC(CATALOG)  PARM('ALL,F1')
```

This command lists all files used by the System/36 environment.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### SSP0010

System/36 job ended abnormally.

[Top](#)



---

## Start Subsystem (STRSBS)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Subsystem (STRSBS) command starts a subsystem using the subsystem description specified in the command. When the subsystem is started, the system allocates the necessary and available resources (storage, work stations, and job queues) that are specified in the subsystem description.

**Allocating Storage:** Storage is allocated to the subsystem according to the storage pool definitions specified in the subsystem description, starting with the lower numbered storage pool definitions. If all the pool definitions cannot be allocated, because the maximum number of storage pools on the system is reached or because insufficient storage is available, messages indicating which pools could not be allocated are sent to the system operator. If storage becomes available later, or if the number of active storage pools is reduced, the available resources are automatically allocated to the subsystem to satisfy its unfulfilled requirements. Any jobs that would normally run in a storage pool that is not allocated are run in the shared storage pool \*BASE.

**Allocating Work Stations:** Work stations are allocated to the subsystem according to the work station entries in the subsystem description. Each work station whose name (or type, if not specified by name) is contained in one of the subsystem description's work station entries, and whose entry specifies AT(\*SIGNON), is allocated to this subsystem unless it is currently signed on to another subsystem. The sign-on prompt is displayed on each work station that is allocated. Work stations that are already signed on in another subsystem remain allocated to that subsystem until the subsystem that allowed the sign-on is ended, or until the user transfers the job to this subsystem.

If multiple subsystems specify the same work station in their work station entries, each subsystem, as it is started, attempts to allocate that work station. Each successive subsystem allocates that work station unless a user signs on while the work station is allocated to one of the previously started subsystems. When a signed-on work station is signed off, it still remains allocated to the same subsystem until another subsystem is started that specifies that work station. However, if a work station is varied offline and several active subsystems specify that work station, the subsystem to which the work station is allocated when it is varied online is unpredictable.

**Allocating Job Queues:** If a job queue is specified in the work entries of the subsystem description, the job queue is allocated to the subsystem. If the job queue does not exist or if it is already allocated to an active subsystem, no job queue is allocated to the subsystem and a message is sent to the system operator. If the job queue later becomes available, it is automatically allocated to the subsystem.

### Restrictions:

1. To use this command, you must have:
  - use (\*USE) authority to the subsystem description and execute (\*EXECUTE) authority to the library that contains that subsystem description.
  - job control (\*JOBCTL) special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Subsystem description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

---

### Subsystem description (SBSD)

Specifies the name and library of the subsystem description that defines the operational environment (subsystem) being started.

The name of the subsystem description cannot be the same as the name of a subsystem that is currently active, even though the subsystem descriptions are in different libraries.

This is a required parameter.

#### Qualifier 1: Subsystem description

*name* Specify the name of the subsystem description that defines the subsystem being started.

**Note:** The IBM-supplied object named QLPINSTALL is not allowed for the subsystem description name.

#### Qualifier 2: Library

**\*LIBL** All libraries in the thread's library list are searched until a match is found.

#### **\*CURLIB**

The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the library where the subsystem description is located.

**Note:** The library QTEMP is not allowed for the library name.

Top

---

## Examples

### Example 1: Starting the Batch Subsystem

```
STRSBS  SBSB(QBATCH)
```

This command starts the batch subsystem named QBATCH.

### Example 2: Starting a User Subsystem

```
STRSBS  SBSB(QGPL/TELLER)
```

This command starts the subsystem that is associated with the TELLER subsystem description in the QGPL library. The subsystem name is TELLER.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF1001**

Wait time expired for system response.

**CPF1004**

Function check occurred during start subsystem.

**CPF1010**

Subsystem name &1 active.

**CPF1011**

Start subsystem failed for SBSID &1 in library &2.

**CPF1012**

No authority to start subsystem.

**CPF1013**

Subsystem &1 in library &2 not found.

**CPF1014**

Subsystem &1 not started.

**CPF1031**

Not authorized to library &1.

**CPF1038**

No authority to use command.

**CPF1049**

Cannot allocate subsystem &1 in library &2.

**CPF1050**

Not enough storage to start subsystem.

**CPF1057**

Subsystem &1 in library &2 damaged.

**CPF1067**

Cannot allocate library &1.

**CPF1080**

Library &1 not found.

**CPF1086**

Subsystem &1 in &2 allocated to your job.

**CPF1099**

Subsystem not started because system ending.

**CPF3D87**

Attempted to use system program QLPCTLIN in QSYS.

Top



---

## Start Search Index (STRSCHIDX)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Start Search Index (STRSCHIDX) command allows you to access a search index without using the Help key and the F11 key.

### Restrictions:

- You must have use (\*USE) authority for the search index.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SCHIDX	Search index	Single values: <u>*USER</u> Other values: <i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Search index	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u>	

Top

---

## Search index (SCHIDX)

Specifies the search index from which the index entries are to be displayed.

This is a required parameter.

### Single values

#### \*USER

The search index names that the user has saved are used. If no names were saved, the search index display is shown with an **empty list** message.

### Qualifier 1: Search index

*name* Specify the name of the search index.

### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

#### \*CURLIB

The current library for the job is used to locate the search index. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library where the search index is located.

Top

---

## Examples

STRSCHIDX

This command accesses the search index names that the user has saved.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

CPF6E66

Requested help information not available.

[Top](#)

---

## Start Support Network (STRSPTN)

### Where allowed to run:

- Batch job (\*BATCH)
- Batch program (\*BPGM)
- Interactive program (\*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (\*EXEC)

**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Support Network (STRSPTN) command allows you to establish an application session through a remote support network to the specified destination application.

This command is provided for customers who want to write their own programs to interface with one of the remote support systems.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
ACCOUNT	Account	<i>Character value</i>	Required, Positional 1
SPTUSRID	User ID	<i>Character value</i>	Required, Positional 2
SPTPWD	Password	<i>Character value</i>	Required, Positional 3
FEDEV	Device description	<i>Name</i>	Required, Positional 4
DESTAPP	Destination application	<i>Character value</i>	Required, Positional 5

[Top](#)

---

## Account (ACCOUNT)

Specifies your account number registered with the support network. The organization account ID must be registered with the support network.

This is a required parameter.

[Top](#)

---

## User ID (SPTUSRID)

Specifies the network user identifier within the specified account. The user identifier must be registered with the support network.

This is a required parameter.

---

## Password (SPTPWD)

Specifies the network password for the specified user identifier. The password must be registered with the support network.

**Note:** A support network password can expire. The user must then change the password interactively by using the Work with Product Information (WRKPRDINF) command.

This is a required parameter.

Top

---

## Device description (FEADEV)

Specifies the Front End Application (FEA) device description.

This is a required parameter.

Top

---

## Destination application (DESTAPP)

Specifies the name for the destination application. The destination application is supplied by the network.

This is a required parameter.

Top

---

## Examples

```
STRSPTN  ACCOUNT(11420880) SPTNUSRID(ACME) SPTNPWD(11111)
          FEADEV(QTIFEA)  DESTAPP(AAAAAA)
```

This command establishes a communication path through the remote support network, for user ID ACME operating under password 11111 at account 11420880. The path allows access to application AAAAAA.

Top

---

## Error messages

None

Top



---

## Start Service Agent (STRSRVAGT)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Service Agent (STRSRVAGT) command allows a user to start an aspect of Service Agent. The aspect to be started is specified by the **Type (TYPE)** parameter.

### Restrictions:

- You must have input/output system configuration (\*IOSYSCFG) special authority and also have use (\*USE) authority to the Create Line Description PPP (CRTLINPPP) command to create service configurations.
- You must have job control (\*JOBCTL) special authority to run the TYPE(\*SBSJOB) of this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
TYPE	Type	*ACTIVATE, *ACTPWD, *MASTERPWD, *SBSJOB	Optional, Positional 1
ACTPWD	Activation password	<i>Character value</i>	Optional
MSTPWD	Master password	<i>Character value</i>	Optional

Top

---

## Type (TYPE)

Specifies the aspect of Service Agent to be started.

This is a required parameter.

### \*SBSJOB

All Service Agent monitoring jobs which normally run in the QSYSWRK subsystem are to be started in that subsystem. This option will have no effect if the QSYSWRK subsystem does not exist or is not started.

### \*ACTPWD

The activation password may be entered as part of Service Agent activation.

### \*ACTIVATE

Service Agent is to be activated. This option may be used only in a batch program. It is useful for those users who have many systems or logical partitions on which to activate Service Agent and would like to distribute a program to do so.

Before this option will run successfully, either the ECS or the Service Agent service configuration must be created. This can be done using the Create Service Configuration (CRTSRVCFG) command.

### \*MASTERPWD

The master password may be entered.

---

## Activation password (ACTPWD)

Specifies the current value of the activation password. Entering this password is required for activation of hardware problem reporting for a network of systems or logical partitions. An activation password is not required to activate inventory collection capabilities or to activate hardware problem reporting for only the local system or logical partition.

The password will not display when you type it.

**Note:** This is a required parameter when TYPE(\*ACTPWD) is specified.

### *character-value*

Specify the activation password.

Top

---

## Master password (MSTPWD)

Specifies the current value of the master password. This may be either the original master password, or the additional value already created.

**Note:** This is a required parameter when TYPE(\*MASTERPWD) is specified.

### *character-value*

Specify the master password.

Top

---

## Examples

```
STRSRVAGT TYPE(*ACTIVATE)
```

This command activates Service Agent.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9899

Error occurred during processing of command.

Top

---

## Start Service Job (STRSRVJOB)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Start Service Job (STRSRVJOB) command starts the remote service operation for a specified job (other than the job issuing the command) so that other service commands can be entered to service the specified job. Any dump, debug, and trace commands can be run in that job until service operation ends. Service operation continues until the End Service Job (ENDSRVJOB) command is run.

### Restrictions:

- To use this command, you must be signed on as QPGMR, QSYSOPR, QSRV, or QSRVBAS, or have use (\*USE) authority to the user profile of the job being serviced.

Top

---

## Parameters

Keyword	Description	Choices	Notes
JOB	Job name	<i>Qualified job name</i>	Required, Positional 1
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
DUPJOB OPT	Duplicate job option	<b>*SELECT</b> , *MSG	Optional

Top

---

## Job name (JOB)

Specifies the job to be serviced. If no job number is given, all of the jobs currently in the system are searched for the simple job name. If duplicates of the specified name are found, messages are sent to the user, and user name and job number must be specified. The job name entered cannot be the name of the job issuing the command.

This is a required parameter.

### Qualifier 1: Job name

*name* Specify the name of the job.

### Qualifier 2: User

*name* Specify the user name that identifies the user profile under which the job was run.

### Qualifier 3: Number

**000000-999999**

Specify the system-assigned job number of the job to be serviced.

Top

---

## Duplicate job option (DUPJOB OPT)

Specifies the action taken when duplicate jobs are found by this command.

### \*SELECT

The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, an escape message is issued.

\*MSG An escape message is issued when duplicate jobs are found.

Top

---

## Examples

```
STRSRVJOB JOB(ABCD)
```

This command starts the remote service operation so that any trace, debug, or dump commands entered in this job are applied to the job named ABCD.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3501

Job is already being serviced, traced, or debugged.

#### CPF3520

Job not found.

#### CPF3524

More than one job with specified name found.

#### CPF3531

Job cannot be serviced.

#### CPF3536

Job completed and cannot be serviced.

#### CPF3549

Job &1/&2/&3 cannot be serviced.

#### CPF3676

Not authorized to service specified job.

#### CPF3909

Service command will not be processed.

#### CPF3918

Service request canceled.

#### CPF3938

Already servicing another job.

Top

---

## Start System Service Tools (STRSST)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start System Service Tools (STRSST) command shows the System Service Tools (SST) menu.

**Note:** Improper use of service tools can damage your server.

You can:

- Start a service function
- Work with active service functions
- Work with disk unit configuration and data
- Work with diskette data recovery
- Work with system partitions
- Work with system capacity
- Work with system security
- Work with service tools user IDs

**Restriction:** To use this command, you must have \*SERVICE special authority.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

STRSST

This command shows the Start System Service Tools menu.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPC7210

System Service Tools detected function check. CAUTION: Any device(s) in diagnostic mode not reset.

##### CPC7211

System Service Tools detected function check. See message CPF5263.

- CPF225C**  
Requesting service tools ID not correct.
- CPF225D**  
Requesting service tools ID password not correct.
- CPF366B**  
Password has expired.
- CPF366C**  
Service tools user ID has been disabled.
- CPF7205**  
Service function already started.
- CPF7215**  
System Service Tools already active in this process.
- CPF7222**  
Function not processed. System Service Tools in process of ending.
- CPF7238**  
Not able to start &1.
- CPF7242**  
Not authorized to system service tools.
- CPF7243**  
Previous request not completed.
- CPF305**  
Work with service tools user IDs not allowed when signed on to SST using a default password.
- CPF306**  
Service tools user ID does not have the necessary functional privileges to start &1.

Top

---

## Start TCP/IP (STRTCP)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
Examples  
Error messages

The Start TCP/IP (STRTCP) command initializes and activates TCP/IP processing, starts the TCP/IP interfaces, starts the TCP/IP server jobs, and starts the TCP/IP Point-To-Point (PTP) profiles. A STRTCP command must be issued before any TCP/IP processing can be performed on the iSeries. This includes Simple Network Management Protocol (SNMP) agent processing.

The TCP/IP interfaces that are started are those that have set the AUTOSTART parameter to a value of \*YES using the Add TCP/IP Interface (ADDTCPIFC), the Change TCP/IP Interface (CHGTCPIFC) commands, or the iSeries Navigator.

The Change IPL Attributes (CHGIPLA) command with the STRTCP parameter set to a value of \*YES can be used to automatically submit the STRTCP command at the completion of IPL.

The TCP/IP application server jobs that can be started are:

- Simple Network Management Protocol (SNMP) agent
- Router Daemon (ROUTED)
- Bootstrap Protocol (BOOTP)
- Trivial File Transfer Protocol (TFTP)
- Domain Name Server (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Distributed Data Management (DDM)
- Virtual terminal support (TELNET)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Line Printer Daemon (LPD)
- Hypertext Transfer Protocol (HTTP)
- Post Office Protocol (POP)
- Remote Execution (REXEC)
- Directory Services (DIRSRV)
- Network Station Login Daemon (NSLD)
- Internet Daemon (INETD)
- Management Central (MGTC)
- On Demand (ONDMD)
- NetServer (NETSVR)
- DataLink File Manager (DLFM)
- Virtual Private Network (VPN)
- Extended Dynamic Remote SQL (EDRSQL)
- IBM Host On-Demand (HOD)
- On-Demand Platform Authentication (ODPA)
- Simple Network Time Protocol (NTP)
- Quality of Service (QoS)

- Triggered Cache Manager (TCM)
- Domino
- WebFacing (WEBFACING)
- Common Information Model Object Manager (CIMOM)

The Start TCP/IP command only starts a TCP/IP application job if the AUTOSTART attribute in the application's configuration is set to \*YES when the command is issued. No TCP/IP application jobs are started in the QSYSWRK subsystem if any of the following is true:

- The TCP/IP licensed program product is not installed.
- All of the TCP/IP applications have an AUTOSTART configuration attribute value of \*NO.
- STRSVR(\*NO) is specified for the STRTCP command. See the description of the STRSVR parameter below.

Use the commands or interfaces listed below to change the configuration for an application so that it starts automatically when the Start TCP/IP command is issued.

- For the SNMP agent use the Change SNMP Attributes (CHGSNMPPA) command specifying AUTOSTART(\*YES)
- For the RouteD server use the Change RouteD Attributes (CHGRTDA) command specifying AUTOSTART(\*YES)
- For the BOOTP server use the Change BOOTP Attributes (CHGBPA) command specifying AUTOSTART(\*YES)
- For the TFTP server use the Change TFTP Attributes (CHGTFTP) command specifying AUTOSTART(\*YES)
- For the DNS server use the Change DNS Attributes (CHGDNSA) command specifying AUTOSTART(\*YES)
- For the DHCP server use the Change DHCP Attributes (CHGDHCPA) command specifying AUTOSTART(\*YES)
- For the DDM server use the Change DDM Attributes (CHGDDMTCPA) command specifying AUTOSTART(\*YES)
- For the TELNET application use the Change TELNET Attributes (CHGTELNA) command specifying AUTOSTART(\*YES)
- For the FTP application use the Change FTP Attributes (CHGFTP) command specifying AUTOSTART(\*YES)
- For the SMTP application use the Change SMTP Attributes (CHGSMTPA) command specifying AUTOSTART(\*YES)
- For the LPD application use the Change LPD Attributes (CHGLPDA) command specifying AUTOSTART(\*YES)
- For the HTTP server application use the Change HTTP Attributes (CHGHTTTPA) command specifying AUTOSTART(\*YES).
- For the Post Office Protocol (POP) version 3 mail servers use the Change POP Mail Server Attributes (CHGPOPA) command specifying AUTOSTART(\*YES)
- For the REXEC server application use the Change REXEC Attributes (CHGRXCA) command specifying AUTOSTART(\*YES)
- Change DIRSVR server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change NSLD daemon attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change INETD daemon attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change MGTC daemon attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change ONDMD server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change NETSVR attributes, including the AUTOSTART parameter, using iSeries Navigator.



- Change DLFM server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change EDRSQL server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change HOD server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change ODPA server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change NTP server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change QoS server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change TCM server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change DOMINO server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change WEBFACING server attributes, including the AUTOSTART parameter, using iSeries Navigator.
- Change CIMOM server attributes, including the AUTOSTART parameter, using iSeries Navigator.

When the STRTCP command is issued, the QTCPIP job in the QSYSWRK subsystem is started. The QTCPIP job is used for activating and deactivating TCP/IP interfaces.

**Note:** The Start TCP/IP (STRTCP) CL command does not need to be issued to use socket applications that run over an SNA network.

**Attention:** Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the Add TCP/IP Remote System Information (ADDTCPRSI) command to do this.

Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

**Restrictions:**

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

---

## Parameters

Keyword	Description	Choices	Notes
STRSVR	Start application servers	<u>*YES</u> , *NO	Optional
STRIFC	Start TCP/IP interfaces	<u>*YES</u> , *NO	Optional
STRPTPPRF	Start point-to-point profiles	<u>*YES</u> , *NO	Optional

Top

---

## Start application servers (STRSVR)

Specifies whether or not TCP/IP application servers are started by the Start TCP/IP (STRTCP) command.

\*YES The STRTCP command starts all TCP/IP application servers with a configuration attribute of AUTOSTART(\*YES) when TCP/IP is activated.

\*NO The STRTCP command does not start any TCP/IP application servers when TCP/IP is activated.

**Note:** This parameter can be used when TCP/IP application servers using AnyNet are already active on your system and you intend to also activate TCP/IP. STRTCP STRSVR(\*NO) activates TCP/IP processing without starting additional TCP/IP servers.

When TCP/IP or AnyNet is already active, use the Start TCP/IP Server (STRTCPSVR) command to start additional TCP/IP application servers.

Top

---

## Start TCP/IP interfaces (STRIFC)

Specifies whether or not to activate all TCP/IP interfaces that specify AUTOSTART(\*YES) when TCP/IP is activated.

**\*YES** When the QTCPIP job is started by the STRTCP command, the job will attempt to activate all TCP/IP interfaces for which AUTOSTART(\*YES) is specified.

**\*NO** The AUTOSTART(\*YES) interface parameter will be ignored. No TCP/IP interfaces will be automatically started.

**Note:** Specifying STRIFC(\*NO) only inhibits the activation of TCP/IP interfaces. It has no effect on interfaces for other AnyNet protocols such as IP over SNA or IP over IPX.

Top

---

## Start point-to-point profiles (STRTPPRF)

Specifies whether or not to activate all point-to-point profiles that specify a configuration attribute of AUTOSTART(\*YES) when TCP/IP is activated.

**\*YES** As part of the running of the STRTCP command, the Start Point-to-Point TCP/IP (STRTCPPTP) command will also be run. The STRTCPPTP command will attempt to start all of the point-to-point profiles with a configuration attribute of AUTOSTART(\*YES).

**\*NO** The STRTCPPTP command will not be run.

Top

---

## Examples

### Example 1: Starting TCP/IP

```
STRTCP
```

This command initializes and activates TCP/IP processing, starts the TCP/IP interfaces, and starts the TCP/IP server jobs.

### Example 2: Starting TCP/IP and TCP/IP Servers

```
STRTCP STRSVR(*YES)
```

Because \*YES is the default value for the STRSVR parameter, the result of issuing this command is identical to the Example 1.

### Example 3: Starting TCP/IP But Not the TCP/IP Servers

```
STRTCP STRSVR(*NO)
```

This will start TCP/IP processing without starting any of the TCP/IP application server jobs.

### Example 4: Starting TCP/IP in Restricted State

```
STRTCP STRSVR(*NO) STRIFC(*NO) STRTPPRF(*NO)
```

This will start TCP/IP processing, even if the system is in restricted state. TCP/IP application servers and IP interfaces will not be started.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9848

Cannot open file &1 in library &2 member &3.

#### CPF9849

Error while processing file &1 in library &2 member &3.

#### TCP1A04

&1 currently active.

#### TCP1A12

Error occurred submitting interface job.

#### TCP1A14

Error occurred starting TCP/IP servers.

#### TCP1A77

&1 completed successfully; however errors occurred.

#### TCP1D03

&1 member record length not correct.

#### TCP1D04

Error occurred processing member &1 of &2/&3.

#### TCP9999

Internal system error in program &1.

Top



---

## Start TCP/IP Interface (STRTCPIFC)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
Examples  
Error messages

The Start TCP/IP Interface (STRTCPIFC) command starts a Transmission Control Protocol/Internet Protocol (TCP/IP) interface. The line associated with the interface is varied on, if required.

This command can be used to:

- Start interfaces that have been specified with the AUTOSTART(\*NO) value on the Add TCP/IP Interface (ADDTCPIFC) and Change TCP/IP Interface (CHGTCPIFC) commands.
- Start an interface that was previously ended by the End TCP/IP Interface (ENDTCPIFC) command.

Routes are bound to interfaces using a best match first algorithm. This algorithm is based on the state of the interface and on the type of service (TOS) specified for the route and interface. When starting an interface, routes associated with an inactive interface can move to the interface started provided that interface can be used to reach the next hop gateway of the route for the requested TOS.

**Attention:** Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the Add TCP/IP Remote System Information (ADDTCPRSI) command to do this. Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

Top

---

### Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	Character value, *AUTOSTART	Required, Positional 1

Top

---

### Internet address (INTNETADR)

Specifies the interface to be started.

#### *character-value*

Specifies the internet address of an interface that had previously been added to the TCP/IP configuration with the ADDTCPIFC command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

#### **\*AUTOSTART**

The TCP/IP interfaces that are started are those that have set the AUTOSTART parameter to a value of \*YES using the Add TCP/IP Interface (ADDTCPIFC) or the Change TCP/IP Interface (CHGTCPIFC) commands.

---

## Examples

### Example 1: Starting an Interface

```
STRTCPIFC  INTNETADR('9.5.11.125')
```

This command causes the TCP/IP protocol stack to activate the interface associated with the internet address 9.5.11.125.

### Example 2: Starting another Interface

```
STRTCPIFC  INTNETADR('156.93.81.7')
```

This command causes the TCP/IP protocol stack to activate the interface associated with the internet address 156.93.81.7.

### Example 3: Starting AUTOSTART(\*YES) Interfaces

```
STRTCPIFC  INTNETADR(*AUTOSTART)
```

This command causes the TCP/IP protocol stack to activate all interfaces that have the AUTOSTART parameter set to \*YES using the Add TCP/IP Interface (ADDTCPIFC) or Change TCP/IP Interface (CHGTCPIFC) commands.

---

## Error messages

### \*ESCAPE Messages

#### TCP1B01

Unable to determine if &1 interface started.

#### TCP1B02

Cannot determine if &1 interface started.

#### TCP1B05

&2 interface not started. Reason &1.

#### TCP1B10

&2 interface not started.

#### TCP1B11

Interface &1 line &2 not started. Maximum &7 active interfaces allowed.

#### TCP1B12

&1 interface not started. &1 interface already active.

#### TCP1B13

&1 interface not started. &1 interface not defined in the TCP/IP configuration.

#### TCP1B14

&1 interface not started. Line description &2 not found.

#### TCP1B15

Line description &2 unusable. Internal errors encountered.

#### TCP1B16

&2 interface not started.

**TCP1B25**

&1 interface not started.

**TCP1B26**

&1 interface not started

**TCP1B27**

&1 interface not started for \*TDLC line description &2.

**TCP265F**

INTNETADR parameter value &2 not valid.

**TCP9999**

Internal system error in program &1.

Top





---

## Start Point-to-Point TCP/IP (STRTCPPTP)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
Examples  
Error messages

The Start Point-to-Point TCP/IP (STRTCPPTP) command is used to start a point-to-point TCP/IP session job. A session job operates in one of two possible modes:

1. Answer mode (\*ANS) sessions allow a remote system to contact this iSeries and establish a point-to-point TCP/IP session.
2. Dial mode (\*DIAL) sessions are used to have this iSeries contact a remote system and establish a point-to-point TCP/IP session.

**Note:** You can start any profiles of linetype \*PPP with this command. You must use iSeries Navigator to configure \*PPP profiles.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	<i>Character value</i> , *AUTOSTART	Required, Positional 1
OUTPUT	Script dialog output	*ERROR, *NONE, *PRINT	Optional
RESTART	Restart	*NO, *YES	Optional
SNDINQMSG	Send inquiry message	*NO, *YES	Optional
AUTODLTCFG	Autodelete configuration	*NO, *YES	Optional

Top

---

### Configuration profile (CFGPRF)

Specifies the point-to-point configuration profile to start. \*AUTOSTART indicates that all point-to-point configuration profiles with this attribute should be started. The profile specifies all the attributes and values necessary to define a session. The mode specified in the profile determines whether this iSeries will be contacting a remote system (\*DIAL session), or if a remote system will be contacting this iSeries (\*ANS session). All other information about the characteristics of the session is also contained in the point-to-point configuration profile.

This is a required parameter.

#### **\*AUTOSTART**

Start all point-to-point configuration files marked with this attribute.

#### **character-value**

Specify the name of a valid, inactive point-to-point configuration profile.

Top

---

## Script dialog output (OUTPUT)

Specifies whether or not to print a copy of the script dialog between this iSeries and the remote system. You can use this dialog to diagnose problems that prevent a point-to-point TCP/IP session from being successfully established.

**Note:** This parameter does not affect job log creation or output.

### \*ERROR

Print the script dialog only if errors occur while establishing the TCP/IP point-to-point session job.

### \*NONE

Do not print the script dialog.

### \*PRINT

Print the script dialog with the spooled output of the job that issued the STRTCPPTP command, regardless of whether or not any errors occurred.

Top

---

## Restart (RESTART)

Specifies whether to restart the TCP/IP point-to-point session job that is running the profile specified on the **Configuration profile (CFGPRF)** parameter.

This parameter is only valid when specified for an active TCP/IP point-to-point profile. If the TCP/IP point-to-point profile is not active, this parameter is ignored.

\*NO Do not restart the TCP/IP point-to-point session job.

\*YES Restart the TCP/IP point-to-point session job.

How the TCP/IP point-to-point session job is restarted depends on the mode of point-to-point session that is running. By mode, the session is restarted as follows:

### **Switched Line Dial (Dial remote systems)**

Reset the session to start by redialing the remote system.

### **Switched Line Answer (Answering incoming calls)**

Reset the session to Ring Wait state (wait for remote system to dial in).

### **Leased Line (Initiator or Terminator) for PPP**

Reset back to LCP Configure Request state.

### **Leased Line (Initiator or Terminator) for SLIP**

Reset back to an 'Active' state

### **Dial-on-Demand**

Reset the session to wait for the next Dial-on-Demand request.

Top

---

## Send inquiry message (SNDINQMSG)

Specifies whether or not to send an inquiry message after any initialization has been completed. The inquiry message will be sent to the QTCP message queue.

Until you respond to the inquiry message, the point-to-point session job is held. While the job is held, you can set up trace tools for diagnosing problems related to this TCP/IP point-to-point session job.

**Note:** This parameter is ignored for any profiles of linetype \*PPP.

**\*NO** Do not send an inquiry message after initialization has been completed.

**\*YES** The point-to-point session job will send an inquiry message after it has completed its initialization. If the configuration profile has the attribute for automatic creation of the controller and device, the inquiry message will be sent after the controller description and device description have been created.

Top

---

## Autodelete configuration (AUTODLTCFG)

Specifies whether or not controller description and device description configuration objects that were automatically created during STRTCPPTP initialization should be automatically deleted when the point-to-point session job ends.

**\*NO** Do not delete any controller description and device description configuration objects that were automatically created by STRTCPPTP. This allows the controller description and device description to be reused the next time a TCP/IP point-to-point session job is started using the specified configuration profile.

**\*YES** Delete any controller description and device description configuration objects that were automatically created during STRTCPPTP initialization.

Top

---

## Examples

### Example 1: Start a Point-To-Point TCP/IP Session Job.

```
STRTCPPTP  CFGPRF(DIALPRF)
```

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "DIALPRF" contains the information that will determine whether this iSeries 400 will be contacting a remote system (\*DIAL session), or if a remote system will be contacting this iSeries 400 (\*ANS session).

### Example 2: Start a Session and Capture the Script Dialog.

```
STRTCPPTP  CFGPRF(DIALPRF)  OUTPUT(*PRINT)
```

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "DIALPRF" contains the information that the iSeries 400 will use to contact the remote system and establish a point-to-point TCP/IP session with it. A copy of the dialog exchanged between the two systems prior to establishing a point-to-point TCP/IP session is written to the default output queue for the job that issued the STRTCPPTP command.

### Example 3: Send an Inquiry Message After Initializing the TCP/IP Session Job.

```
STRTCPPTP  CFGPRF(ANSWERPRF)  SNDINQMSG(*YES)
```

This command uses the SNDINQMSG parameter, which is normally used only if it is necessary to collect detailed internal trace information when the controller and device description used by the point-to-point session job are automatically created.

This command starts a point-to-point TCP/IP session job. The point-to-point configuration profile "ANSWERPRF" contains the information that will be used to establish a point-to-point TCP/IP session when a remote system contacts this iSeries 400.

After automatically creating the controller and device description it will use, the point-to-point session job will send an inquiry message to the QTCP message queue. Depending on the response, the job will either continue running or will be cancelled.

**Example 4: Restarting a Running Point-To-Point Profile.**

```
STRTCPPTP CFGPRF(ANSWERPRF) RESTART(*YES)
```

This command will restart the running point-to-point profile ANSWERPRF. How the session is reset (restarted) depends on the connection type of the line (switched or unswitched) and the mode (dial or answer). For more information on how the session would be reset, see the details for RESTART(\*YES) above.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP1A1F

Cannot process request while &3/&2/&1 using &6.

#### TCP8205

Required object &2/&1 type \*&3 not found.

#### TCP8207

STRTCPPTP &1 not performed. Job &6/&5/&4 using &1.

#### TCP8208

STRTCPPTP &1 not performed. See previous messages.

Top

# Start TCP/IP Server (STRTCPSVR)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Conditional

Parameters  
 Examples  
 Error messages

The Start TCP/IP Server (STRTCPSVR) command is used to start the TCP/IP application servers that are shipped with OS/400 or the TCP/IP product. The number of server jobs started by this command is specified, where appropriate, in the configuration for each TCP/IP application.

The STRTCPSVR command can only be used when TCP/IP is fully operational. The interface server job, QTCPIP, must be available. If TCP/IP processing was started by running the STRTCP (Start TCP/IP) command when the iSeries was in restricted state, the STRTCPSVR command will fail.

All servers have an autostart (AUTOSTART) parameter on the associated configuration command (for example, Change FTP Attributes (CHGFTP)). This parameter indicates if the server should be started when the Start TCP/IP (STRTCP) command is entered. The STRTCPSVR command ignores the value of a server's autostart attribute, unless \*AUTOSTART is specified for the **Server application (SERVER)** parameter. If a server has \*YES for the AUTOSTART attribute, then the STRTCPSVR command will start the server when SERVER(\*AUTOSTART) is specified. Additional servers can automatically be added to list of servers that STRTCPSVR will support by using the ADDTCPSVR (Add TCP/IP Server) CL command.

## Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

## Parameters

Keyword	Description	Choices	Notes
SERVER	Server application	Single values: *ALL, *AUTOSTART Other values (up to 300 repetitions): <i>Character value</i>	Optional, Positional 1
RESTART	Restart server	*NONE, *HTTP, *DNS, *DHCP, *QOS	Optional
HTTPSVR	HTTP server	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Name</i> , *ADMIN	
	Element 2: Instance startup values	<i>Character value</i> , *NONE	
DNSSVR	DNS server	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Character value</i>	
	Element 2: Instance startup values	<i>Character value</i> , *NONE	
TCMSVR	TCM server	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Instance name	<i>Character value</i> , *ALL	
	Element 2: Instance startup values	<i>Character value</i> , *NONE	

Keyword	Description	Choices	Notes
TOMCATSVR	ASFTOMCAT server	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Server instance name	<i>Character value</i> , *ALL	
NTPSRV	SNTP service	*ALL, *CFGFILE, *CLIENT, *SERVER	Optional

Top

---

## Server application (SERVER)

Specifies the TCP/IP application servers to be started by this command.

### Single values

**\*ALL** All of the TCP/IP application servers and all HTTP, DNS, and TCM server instances are started.

### \*AUTOSTART

All of the TCP/IP application servers that are defined with \*YES for the AUTOSTART attribute are to be started.

### Other values (up to 300 repetitions)

#### \*ASFTOMCAT

The Apache Software Foundation (ASF) Tomcat server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*ASFTOMCAT) will result in a diagnostic message if the Apache Software Foundation (ASF) Tomcat server has already been started.

#### \*BOOTP

The Bootstrap Protocol (BOOTP) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*BOOTP) will result in a diagnostic message if the BOOTP server job has already been started.

#### \*CIMOM

The Common Information Model Object Manager (CIMOM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*CIMOM) will result in a diagnostic message if the CIMOM server has already been started.

**\*DBG** The Debug (DBG) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*DBG) will result in a diagnostic message if the DBG server has already been started.

#### \*DDM

The Distributed Data Manager (DDM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*DDM) will result in a diagnostic message if the DDM server job has already been started.

#### \*DHCP

The Dynamic Host Configuration Protocol (DHCP) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*DHCP) will result in a diagnostic message if the DHCP server has already been started.

#### \*DIRSRV

The Lightweight Directory Access Protocol (LDAP) server is started. The LDAP server is also known as the directory services (DIRSRV) server. Subsequent use of the STRTCPSVR command specifying SERVER(\*DIRSRV) will result in a diagnostic message if the LDAP server has already been started.

**\*DLFM**

The DataLink File Manager (DLFM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*DLFM) will result in a diagnostic message if the DLFM server has already been started.

**\*DNS** One or more instances of the Domain Name System (DNS) server are started. See the parameter documentation for the **DNS server (DNSSVR)** parameter for more information on DNS server instances. Subsequent use of the STRTCPSVR command specifying SERVER(\*DNS) will result in a diagnostic message if the DNS server has already been started.

**\*DOMINO**

The Lotus Domino (DOMINO) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*DOMINO) will result in a diagnostic message if the DOMINO server has already been started.

**\*EDRSQL**

The Extended Dynamic Remote SQL (EDRSQL) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*EDRSQL) will result in a diagnostic message if the EDRSQL server has already been started.

**\*FTP** The File Transfer Protocol (FTP) servers are started, based on the number of servers configured with the Change FTP Attributes (CHGFTPA) command. Subsequent use of the STRTCPSVR command specifying SERVER(\*FTP) will start one additional FTP server.

**Note:** Having more than one FTP server job running can improve the performance of initiating a session when multiple users attempt to connect to the server in a short period of time.

**\*HOD** The IBM Host On-Demand (HOD) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*HOD) will result in a diagnostic message if the HOD server has already been started.

**\*HTTP**

One or more instances of the World Wide Web HyperText Transfer Protocol (HTTP) server is started. See the parameter documentation for the **HTTP server (HTTPSVR)** parameter for more information on HTTP server instances. Subsequent use of the STRTCPSVR command specifying SERVER(\*HTTP) will result in a diagnostic message if the HTTP server has already been started.

**\*INETD**

The Internet Daemon (INETD) is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*INETD) will result in a diagnostic message if the INETD server has already been started.

**\*LPD** The line printer daemon (LPD) servers are started, based on the number of servers configured with the Change LPD Attributes (CHGLPDA) command. Subsequent usage of the STRTCPSVR command specifying SERVER(\*LPD) will start one additional LPD server.

**Note:** LPD works most efficiently when two or more servers are running. Running only one server will work, but no jobs can be received while a current job is running. If a large print job is running, new jobs have to wait before LPD is ready to accept any new LPR requests.

**\*MGTC**

The Management Central (MGTC) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*MGTC) will result in a diagnostic message if the MGTC server has already been started.

**\*NETSVR**

The NetServer (NETSVR) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*NETSVR) will result in a diagnostic message if the NETSVR server has already been started.

**\*NSLD**

The Network Station Login Daemon (NSLD) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*NSLD) will result in a diagnostic message if the NSLD server has already been started.

**\*NTP** The Simple Network Time Protocol (SNTP) services servers are started. See the parameter documentation for the **SNTP service (NTPSRV)** parameter for more information on specifying which SNTP services to start. Subsequent use of the STRTCPSVR command specifying SERVER(\*NTP) will result in a diagnostic message if the indicated SNTP services server has already been started.

**Note:** If both client and server SNTP services have been started, running the ENDTCPSSVR command specifying SERVER(\*NTP) will end both client and server services. To change which SNTP services are active, run the End TCP/IP Server (ENDTCPSVR) command specifying SERVER(\*NTP), which will end all active SNTP services. Then run the STRTCPSVR command again, specifying SERVER(\*NTP), and either NTPSRV(\*CLIENT) or NTPSRV(\*SERVER).

**\*ODPA**

The On-Demand Platform Authentication (ODPA) server is started. Subsequent use of the STRTCPSVR SERVER(\*ODPA) command results in a diagnostic message if the ODPA server has already been started.

**\*ONDMD**

The On Demand Server (ONDMD) is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*ONDMD) will result in a diagnostic message if the ONDMD server has already been started.

**\*POP** The Post Office Protocol (POP) version 3 mail servers are started based on the number of servers configured with the Change POP Server Attributes (CHGPOPA) command. Subsequent use of the STRTCPSVR command specifying SERVER(\*POP) will start one additional POP server.

**\*QOS** The Quality of Service (QoS) server is started. Subsequent use of the STRTCPSVR SERVER(\*QOS) command results in a diagnostic message if the QoS server has already been started.

**\*REXEC**

The Remote Execution (REXEC) servers are started based on the number of servers configured with the Change REXEC Attributes (CHGRXCA) command. Subsequent use of the STRTCPSVR command specifying SERVER(\*REXEC) will start one additional REXEC server.

**\*ROUTED**

The Router Daemon (ROUTED) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*ROUTED) will result in a diagnostic message if the Routed server has already been started.

**\*SMTP**

The Simple Mail Transfer Protocol (SMTP) client and server jobs are started. Additional SMTP client and server jobs cannot be started. Subsequent use of the STRTCPSVR SERVER(\*SMTP) command results in a diagnostic message if the SMTP server jobs have already been started.

**\*SNMP**

The Simple Network Management Protocol (SNMP) agent server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*SNMP) will result in a diagnostic message if the SNMP server has already been started.

**\*TCM** One or more instances of the Triggered Cache Manager (TCM) server is started. See the parameter documentation for the **TCM server (TCMSVR)** parameter for more information on specifying which TCM server instances to start. Subsequent use of the STRTCPSVR command specifying SERVER(\*TCM) will result in a diagnostic message if the indicated TCM server instance has already been started.



#### \*TELNET

The TELNET server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*TELNET) will start one additional TELNET server.

**Note:** Having more than one TELNET server job running reduces the chances of having connection attempts refused.

\*TFTP The Trivial File Transfer Protocol (TFTP) servers are started based on the number of servers configured with the Change TFTP Attributes (CHGTFTPA) command. Subsequent use of the STRTCPSVR SERVER(\*TFTP) command results in a diagnostic message if the TFTP server jobs have already been started.

\*VPN The Virtual Private Network (VPN) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*VPN) will result in a diagnostic message if the VPN server has already been started.

#### \*WEBFACING

The WebFacing (WEBFACING) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*WEBFACING) will result in a diagnostic message if the WebFacing server has already been started.

Top

---

## Restart server (RESTART)

Specifies whether to restart the selected server when the STRTCPSVR command is run.

The SERVER parameter value specified must be one of the following:

- \*ALL
- \*DNS
- \*DHCP
- \*HTTP
- \*QOS

or this parameter is ignored.

#### \*NONE

Do not restart any server.

#### \*HTTP

Restart the HTTP server using the values specified on the **HTTP server (HTTPSVR)** parameter. If the server is not currently running when the STRTCPSVR command is run, this parameter is ignored and the server is started.

#### \*DHCP

The Dynamic Host Configuration Protocol (DHCP) server job that is already running will reprocess its configuration file and initialize with any changes that it found. If the DHCP processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

\*DNS A Domain Name System (DNS) server job that is already running will reprocess its configuration file and initialize with any changes that it found. If the DNS processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

\*QOS The Quality of Service (QoS) server job that is already running will reprocess its configuration file and initialize with any changes that it found. If the QoS processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

Top

---

## HTTP server (HTTPSVR)

Specifies the HTTP server instance to be started as well as any additional startup values to be used by the HTTP server to control the server instance. (This server is also known as the IBM HTTP Server).

If multiple HTTP server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

For more information on the HTTP Server, visit the HTTP Server homepage:

<http://www.ibm.com/eserver/series/software/http>

### Single values

**\*ALL** All defined instances of the HTTP server will be started.

### Element 1: Server instance

#### **\*ADMIN**

The Administration Server will be started. The Administration Server is an instance of the HTTP server that allows administration of certain iSeries functions using a Web browser.

*name* Specify the name of the HTTP server instance to be started.

### Element 2: Instance startup values

#### **\*NONE**

No additional startup values are defined.

#### *character-value*

Specify the additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified server instance.

**Note:** The user must have input/output system configuration (\*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have \*IOSYSCFG authority, the start request will be rejected.

**Note:** IBM recommends that you use these overrides with caution; they are intended only for special circumstances. To specify startup values, IBM recommends you use the IBM HTTP Server Administration web-based interface rather than this command. Startup parameter values override specific server instance values, configuration directive values, global server values, and default values.

Instance startup values for the HTTP Server (powered by Apache).

#### **-netccsid [nnn]**

Overrides the DefaultNetCCSID directive

#### **-fscsid [nnn]**

Overrides the default DefaultFsCCSID directive

#### **-d [serverroot]**

Set the initial value for the ServerRoot variable to serverroot. The default is /QIBM/UserData/HTTPPA/logs.

#### **-f [configuration]**

Use the values in the configuration on startup. If the configuration does not begin with a /, then it is treated as a path relative to the ServerRoot. The default is conf/httpd.conf.

#### **-C [directive]**

Process the given "directive" (just as if it had been part of a configuration file) before actually reading the regular configuration files.

- c [directive]**  
Process the given "directive" after reading all the regular configuration files.
- vv [verbose level service trace]**  
Turn on verbose level service tracing .
- vi [informational level service trace]**  
Turn on informational level service tracing.
- ve [error level service trace]**  
Turn on error level service tracing.
- V [no value is provided]**  
Display the base version of the server, its build date, and a list of compile time settings which influence the behavior and performance of the server, then exit.
- l [no value is provided]**  
Display a list of all modules compiled into the server, then exit.
- t [no value is provided]**  
Test the configuration file syntax (i.e., read all configuration files and interpret them) but do not start the server. If the configuration contains errors, display an error message and exit with a non-zero exit status, otherwise display "Syntax OK" and terminate with a zero exit status. This command checks to see if all DocumentRoot entries exist and are directories.

Top

---

## DNS server (DNSSVR)

Specifies the Domain Name System (DNS) server instance to be started as well as any additional startup values to be used by the DNS server to control the server instance.

If multiple DNS server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

### Single values

**\*ALL** All defined instances of the DNS server will be started.

### Element 1: Server instance

#### *character-value*

Specify the name of the DNS server instance to be started.

### Element 2: Instance startup values

#### **\*NONE**

No additional startup values are defined.

#### *character-value*

Specifies additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified server instance.

**Note:** The user must have input/output system configuration (\*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have \*IOSYSCFG authority, the start request will be rejected.

The list of instance startup values for the DNS Server follows:

**-d [n]** where n is a number from 1 to 11. This sets the debug level of the server to start.

Top

---

## TCM server (TCMSVR)

Specifies the Triggered Cache Manager (TCM) server instance to be started as well as any additional startup values to be used by the TCM server to control the server instance.

If multiple TCM server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

### Single values

#### \*NONE

No defined instances of the TCM server will be started.

### Element 1: Instance name

**\*ALL** All defined instances for the TCM server will be started.

#### *character-value*

Specify the name of the TCM server instance to be started.

### Element 2: Instance startup values

#### \*NONE

No additional startup values are defined.

#### *character-value*

Specify the additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified instance name.

**Note:** The user must have input/output system configuration (\*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have \*IOSYSCFG authority, the start request will be rejected.

Top

---

## ASFTOMCAT server (TOMCATSVR)

Specifies the Tomcat server instance to be started.

If multiple Tomcat server instance names have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

### Single values

#### \*NONE

No defined instances of the Tomcat server will be started.

### Element 1: Server instance name

**\*ALL** All defined instances for the Tomcat server will be started.

#### *character-value*

Specify the name of the Tomcat server instance to be started.

**Note:** The user must have input/output system configuration (\*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have \*IOSYSCFG authority, the start request will be rejected.

Top

---

## SNTP service (NTPSRV)

Specifies the Simple Network Time Protocol (SNTP) services to be started.

### Single values

#### \*CLIENT

The SNTP client is started.

#### \*SERVER

The SNTP server is started.

\*ALL Both the client and server are started.

#### \*CFGFILE

The values of the keywords Client Autostart (AUTOSTART) and Server Autostart (SVRAUTOSTR) in the Change SNTP Attributes (CHGNTPA) command configuration file are used to determine what services to start. If the value is set to \*YES, that service is started.

Top

---

## Examples

### Example 1: Starting All TCP/IP Servers with AUTOSTART(\*YES)

```
STRTCPSVR  SERVER(*AUTOSTART)
```

This command starts all of the TCP/IP application servers that have the AUTOSTART attribute in the application configuration set to \*YES.

### Example 2: Starting All TCP/IP Servers

```
STRTCPSVR  SERVER(*ALL)
```

This command starts all of the TCP/IP application servers that have been configured. For example: If the Change FTP Attributes (CHGFTPA) command was previously used to configure two FTP servers, both servers are started when STRTCPSVR is issued. This example is also true for other TCP/IP application servers.

Where appropriate, the number of servers to start is based on the number of servers configured for the server being started.

### Example 3: Starting the TELNET Server

```
STRTCPSVR  SERVER(*TELNET)
```

This command starts the TCP/IP TELNET application server. If the TELNET server was previously started, one additional TELNET server job is started.

### Example 4: Restarting the HTTP Server

```
STRTCPSVR  SERVER(*HTTP)  RESTART(*HTTP)
```

This command restarts the TCP/IP HTTP application server for all instances of the HTTP server. If the HTTP server was not currently running, then all defined instances of the HTTP server would be started.

#### **Example 5: Starting an HTTP Server Instance**

```
STRTCPSVR  SERVER(*HTTP)  HTTPSVR('http1')
```

This command starts the TCP/IP HTTP application server instance named 'http1' using the startup values previously defined for this server instance.

#### **Example 6: Specifying Startup Values for an HTTP Instance**

```
STRTCPSVR  SERVER(*HTTP)  HTTPSVR(HTTP1 '-p 81 -sslport 443')
```

This command starts the TCP/IP HTTP application server instance named HTTP1, and specifies that the server instance should listen on port 81 for unsecure requests and on port 443 for secure requests. The ports defined here will override any previously defined ports to be used by this server instance.

#### **Example 7: Starting a DNS Server Instance**

```
STRTCPSVR  SERVER(*DNS)  DNSSVR('dns1')
```

This command starts the TCP/IP DNS application server instance named 'dns1' using the startup values previously defined for this server instance.

#### **Example 8: Specifying Startup Values for a DNS Instance**

```
STRTCPSVR  SERVER(*DNS)  DNSSVR(DNS1 '-d 5')
```

This command starts the TCP/IP DNS application server instance named DNS1, and specifies that the server instance should turn on debug level 5.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF3894**

Cancel reply received for message &1.

#### **TCP1A11**

&1 failed.

#### **TCP1A77**

&1 completed successfully; however errors occurred.

Top

---

## Start TIE Session (STRTISSN)

### Where allowed to run:

- Batch job (\*BATCH)
- Batch program (\*BPGM)
- Batch REXX procedure (\*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (\*EXEC)

Threadsafe: No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Start Technical Information Exchange Session (STRTISSN) command establishes the data link for a TIE batch session. This command must precede other TIE batch commands.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
SPTUSRID	User ID	<i>Name</i>	Required, Positional 1
SPTPWD	Password	<i>Character value</i>	Required, Positional 2
ACCOUNT	Account	<i>Character value, <u>*RTV</u></i>	Optional

[Top](#)

---

## User ID (SPTUSRID)

Specifies the user identifier needed to sign on to the remote support network.

This is a required parameter.

[Top](#)

---

## Password (SPTPWD)

Specifies the password needed to sign on to the remote support network.

This is a required parameter.

[Top](#)

---

## Account (ACCOUNT)

Specifies the network account number needed to sign on the remote support network. If the account number is not specified, the account number from the contact database is used.

\*RTV The account number from the contact database is used.

*account-number*

Specify the account number being used.

[Top](#)

---

## Examples

```
STRTISSN SPTUSRID(ACME) SPTPWD(11111) ACCOUNT(11420880)
```

This command displays the TIE main menu for account number 11420880.

[Top](#)

---

## Error messages

None

[Top](#)



## Start Trace (STRTRC)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Start Trace (STRTRC) command starts traces of original program model (OPM) programs and Integrated Language Environment (ILE) procedures and Java programs (both compiled and JIT). Tracing can be done for multiple jobs using this command. Any number of trace sessions can be started, but active trace session identifiers must be unique across the system. This command can trace call-return flow, data returned by trace points defined in the operating system, component trace information or all three.

The trace session continues until ended with the End Trace (ENDTRC) command or automatically by the watch for trace event functionality. A trace session can be ended from the same job or a different job.

### Restrictions:

- To use this command, you must have service (\*SERVICE) special authority, or be authorized to the Service Trace function of OS/400 through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- If you specify a generic user name for the **Job name (JOB)** parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Trace Any User function of OS/400 through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_ALLOBJ\_TRACE\_ANY\_USER, can also be used to change the list of users that are allowed to perform trace operations.
- A trace cannot be defined to trace all job names and all users.
- When the WCHJOB parameter is specified, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (\*JOBCTL) special authority.

Top

## Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	Name, *GEN	Required, Positional 1
JOB	Jobs	Single values: * Other values (up to 8 repetitions): <i>Element list</i>	Optional
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name, *ALL</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</i>	
	Qualifier 3: Number	000001-999999, * <u>ALL</u>	
	Element 2: Thread ID to include	Values (up to 20 repetitions): <i>Character value, *<u>ALL</u>, *INITIAL, *SELECT</i>	
JOBTYP	Job types	* <u>ALL</u> , *ACTIVE, *NEW	Optional
MAXSTG	Maximum storage to use	1024-4000000, <u>10000</u>	Optional
TRCFULL	Trace full	*STOPTRC, * <u>WRAP</u>	Optional

Keyword	Description	Choices	Notes
JOBTRCTYPE	Trace type	Values (up to 2 repetitions): <u>*ALL</u> , *FLOW, *DATA, *TRCTYPE	Optional
TRCTYPE	Trace type	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Component	<i>Character value</i> , *ASPMGT, *CLUSTER, *DIRSRV, *ENVVAR, *HTTP, *IFS, *IPC, *LOCKSPACE, *OPASSIST, *POSIXMISC, *POSIXPROC, *PTHREAD, *QSHELL, *RAS, *SECURITY, *SOCKETS, *THREADMGT	
	Element 2: Trace level	*ERROR, *INFO, *VERBOSE	
TRCFTR	Trace filter	<i>Name</i> , <u>*NONE</u>	Optional
WCHMSG	Watch for message	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Message identifier	<i>Name</i>	
	Element 2: Comparison data	<i>Character value</i> , <u>*NONE</u>	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional
	Element 1: Message queue	Single values: *SYSOPR, *JOBLOG, *HSTLOG Other values: <i>Qualified object name</i>	
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u>	
WCHJOB	Watched job	Single values: * Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000001-999999, <u>*ALL</u>	
WCHLICLOG	Watch for LIC log entry	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Major code	<i>Character value</i> , *ALL	
	Element 2: Minor code	<i>Character value</i> , *ALL	
	Element 3: Comparison data	<i>Character value</i> , <u>*NONE</u>	
WCHTIMO	Length of time to watch	1-43200, <u>1440</u> , *NOMAX	Optional
TRCPGM	Trace program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Trace program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u>	
TRCPMITV	Time interval	1-9999, <u>*NONE</u>	Optional

Top

## Session ID (SSNID)

Specifies a session identifier for this trace. This identifier must be unique across all active traces on the system.

This is a required parameter.

**\*GEN** The system will generate a unique session identifier for this trace.

**name** Specify the session identifier for this trace.

Top

---

## Job name (JOB)

Specifies which jobs are to be traced.

### Single values

\*        Only the job that issues the STRTRC (Start Trace) command is to be traced.

### Element 1: Job name

#### Qualifier 1: Job name

##### *generic-name*

Specify the generic name of the jobs to be traced. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

\*ALL All jobs names with the specified job user name are traced. \*ALL for the job name is considered to be a generic job specification because it will trace all jobs that meet the job user name qualifiers that you specified.

*name* Specify the name of the job to be traced. Up to eight job names may be specified, but only one generic job name is allowed.

#### Qualifier 2: User

##### *generic-name*

Specify the generic user name of the jobs from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic user name specifies all jobs with user names that begin with the generic prefix.

\*ALL All job user name with the specified job name are traced. \*ALL for the job user name is considered to be a generic job specification because it will trace all jobs that meet the job name qualifiers that you specified.

*name* Specify the user name of the job to be traced.

#### Qualifier 3: Number

\*ALL All jobs with the specified job name and user name are traced. \*ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified.

##### *000001-999999*

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier or generic user name qualifier is specified.

### Element 2: Thread ID to include

**\*ALL** All threads within the specified job name and user name are traced.

**\*INITIAL**

Only the initial thread within the specified job name and user name is traced.

**\*SELECT**

A list of thread identifiers is shown from which the user can select up to twenty to be traced. This value is only valid if the command is run in an interactive job.

*thread-identifier*

Specify the identifiers of up to twenty threads whose information is to be included.

Top

---

## Job types (JOBTYPE)

Specifies the types of jobs for which trace data is to be collected.

**\*ALL** All jobs that meet the job name selection criteria will be included in this trace collection. This includes jobs that are currently active on the system. If a generic job name was specified, new jobs that start after this trace session begins and meet the generic job name selection criteria, will be included in the trace collection as well.

**\*ACTIVE**

Only jobs that are currently active at the time this trace session is being activated and meet the job name selection criteria will be included in this trace collection.

**\*NEW** Only new jobs that begin after this trace session is activated and meet the generic job name selection criteria will be included in this trace collection.

Top

---

## Maximum storage to use (MAXSTG)

Specifies the requested maximum amount of storage, in kilobytes (K), to use for the collected trace records. The system will calculate the minimum amount of storage that is necessary for the trace; this minimum storage size calculation is dependent upon the system's processor configuration. The minimum amount of storage may be significantly larger than the size specified on the MAXSTG parameter; the system will use the larger of the two values.

**10000** Up to 10,000 kilobytes of storage is used.

**1024-4000000**

Specify the maximum amount of storage, in kilobytes, to be used to store trace records (one kilobyte equals 1024 bytes).

Top

---

## Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

**\*WRAP**

When the trace storage is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

**\*STOPTRC**

Tracing stops when the trace storage is full of trace records.

---

## Trace type (JOBTRCTYPE)

Specifies the types of job trace data to be stored in the trace file.

You can specify 2 values for this parameter.

**\*ALL** All the trace data collected is stored in trace records. This includes tracing the flow of control and the trace data itself.

**\*FLOW**

The flow of control is traced when OPM programs and ILE procedures are called and when they return.

**\*DATA**

The data that is provided at predefined trace points within the operating system is stored in trace records. This includes trace records for the CL commands that have run.

**\*TRCTYPE**

Trace the specific components specified for the **Trace type (TRCTYPE)** parameter.

---

## Trace type (TRCTYPE)

Specifies the component to trace and the level to use. You can specify up to 50 trace components to be traced.

### Single values

**\*NONE**

No component trace is specified.

### Element 1: Component

Each trace type is identified by a special value. Specify trace component special values from the following table:

TRACE DESCRIPTION	SPECIAL VALUE
ASP Management	*ASPMGT
Directory Services	*DIRSRV
Cluster Resource Services	*CLUSTER
Environment Variable APIs	*ENVVAR
HTTP Server powered by Apache	*HTTP
Integrated File System	*IFS
Interprocess Communication APIs	*IPC
Lock Space Management	*LOCKSPACE
Operational Assistant	*OPASSIST
POSIX Miscellaneous APIs	*POSIXMISC
POSIX Process-Related APIs	*POSIXPROC
POSIX Thread APIs	*PTHREAD
Qshell	*QSHELL
Remote Access Services	*RAS
Software Security	*SECURITY
Sockets APIs	*SOCKETS
Thread Management	*THREADMGT

### Element 2: Trace level

Specifies the trace level to be used for the specified component.

### **\*ERROR**

The trace level ERROR will be used. Components typically trace error return codes, exception conditions, and invalid input. The amount of data is usually small.

### **\*INFO**

The trace level INFO will be used, which also includes ERROR trace level data. Components typically trace entry and exit from interfaces, parameters and return codes, and major changes of flow or semantics caused by input or other decisions. The amount of data is moderate.

### **\*VERBOSE**

The trace level VERBOSE will be used, which also includes INFO and ERROR trace level data. Components typically trace detailed data that could assist in debugging control flow, data corruption, data structures, environment, call stacks, and resource allocations. The amount of data can be large.

Top

---

## **Trace filter (TRCFTR)**

Specifies the trace filter to be used. The Add Trace Filter (ADDTRCFTR) command must be issued to define the trace filter. The filter determines what information is collected in the trace based on the filter values. The filter values are compared to the actual trace data. If a filter is not specified, then all trace information is collected.

The trace filter parameter can only be specified if \*ALL or \*FLOW has been specified for the **Trace type (JOBTRCTYPE)** parameter. The trace filter applies to the \*FLOW (call/return) trace only.

### **\*NONE**

No trace filter will be used.

*name* Specify the name of the trace filter to be used.

Top

---

## **Watch for message (WCHMSG)**

Specifies up to five message identifiers which are to be watched for. If a value other than \*NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace program is called; if no trace exit program is defined, the trace stops.

### **Single values**

#### **\*NONE**

No messages will be watched for.

### **Element 1: Message identifier**

*name* Specify the 7-character message identifier to be watched for.

### **Element 2: Comparison data**

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data includes the specified text, the watched for condition is true. If the message data does not contain the specified text, the trace function continues.

### \*NONE

No comparison data is specified. If a message matching the specified message ID is added to the specified message queue or log, the watched for condition is true.

### *character-value*

Specify the text string used to compare against the message data of the watched for message. If this text is found anywhere in the message data of a watched for message, the watch condition is considered to be true. This text is case sensitive. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

Top

---

## Watched message queue (WCHMSGQ)

Specifies where to watch for the message identifiers specified on the WCHMSG parameter. You can specify to watch the message being added to the system operator message queue, the history log, other message queues, and job logs. Up to three message queues or special values can be specified.

### Element 1: Message queue

#### Single values

#### \*SYSOPR

Watch messages added to the system operator message queue (QSYSOPR message queue in library QSYS).

#### \*JOBLOG

Watch messages added to the job logs of the jobs specified for the **Watched job (WCHJOB)** parameter.

#### \*HSTLOG

Watch messages added to the history log QHST.

### Qualifier 1: Message queue

*name* Specify the name of the message queue to watch.

### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library where the message queue is located.

Top

---

## Watched job (WCHJOB)

Specifies the job whose job log is watched for the messages specified on the WCHMSG parameter. The specified job will only be watched if \*JOBLOG is specified on the WCHMSGQ parameter. Up to five job names may be specified.

### Single values

\* Only the job log of the job that issued this trace command is watched.

### Element 1: Job name

#### Qualifier 1: Job name

##### *generic-name*

Specify the generic name of the job to be watched. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

*name* Specify the name of the job to be watched.

#### Qualifier 2: User

*name* Specify the user name of the job to be watched.

#### Qualifier 3: Number

\*ALL All jobs with the specified job name and user name are watched.

##### *000001-999999*

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier is specified.

Top

---

## Watch for LIC log entry (WCHLICLOG)

Specifies up to five licensed internal code (LIC) log entry identifiers which are to be watched for. Each LIC log entry contains a major and a minor code. The watched for condition will be met if a LIC log entry is added that matches the specified major and minor codes and any comparison data specified. When the watched for log entry is added to the LIC log, the trace exit program is called, even when the comparison data specified does not match; if no trace exit program is defined, the trace stops.

### Single values

#### \*NONE

No LIC log entries will be watched for.

### Element 1: Major code

\*ALL Any LIC log entry major code will be considered to be a match. If \*ALL is specified for the major code, you cannot specify \*ALL for the LIC log entry minor code.

#### *character-value*

Specify the LIC log major code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### Element 2: Minor code



**\*ALL** Any LIC log entry minor code will be considered to be a match. If \*ALL is specified for the minor code, you cannot specify \*ALL for the LIC log entry major code.

***character-value***

Specify the LIC log minor code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

**Element 3: Comparison data**

Specify comparison data to be used if a log entry matching the specified major and minor codes is added to the licensed internal code (LIC) log. If this text is found in the LIC log entry data fields of the watched for log entry, the watched for condition is true. If this text is not found in the LIC log entry data fields of the watched for log entry and no exit program is specified on the TRCPGM parameter, the trace function continues. If the log entry matches the specified major and minor codes and an exit program is specified on the TRCPGM parameter, but the entry data does not contain the specified text, the exit program is called to determine if the trace should continue or stop.

**\*NONE**

No comparison data is specified. If a LIC log entry matching the specified major and minor codes is added to the LIC log, the watched for condition is true.

***character-value***

Specify the text string used to compare against the entry data of the watched for log entry. If this text is found in the LIC log entry data fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server name, job name, user ID, job number, thread ID, exception ID, LIC module compile binary timestamp, LIC module offset, LIC module RU name, LIC module name, LIC module entry point name. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

When watching for an exception ID, all four hexadecimal digits of the exception ID must be specified. Also, the prefix MCH may be specified if you want to compare only against the exception ID field and avoid possible substring matches with the other fields.

Top

---

## Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

**1440** The time limit for watching for a particular message or LIC log entry is 1440 minutes (24 hours).

**\*NOMAX**

There is no time limit for watching for a particular message or LIC log entry.

**1-43200**

Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

---

## Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

There are three input parameters and one output parameter associated with the trace program. The four parameters are required:

1	Trace option setting	Input	Char(10)
2	Reserved	Input	Char(10)
3	Error detected	Output	Char(10)
4	Comparison data	Input	Char(*)

Allowed values for the "Trace option setting" parameter are:

**\*ON** The watch for trace facility is starting when the collection of trace information is started.

**\*MSGID**

A match on a message id specified on WCHMSG parameter occurred.

**\*LICLOG**

A match on a LIC log specified on the WCHLICLOG parameter occurred.

**\*CMPDATA**

The major and minor code of a LIC log matched, but the comparison data did not.

**\*INTVAL**

The time interval specified on TRCPGMITV parameter is elapsed.

**\*WCHTIMO**

The length of time to watch specified on WCHTIMO parameter is elapsed.

The "Reserved" parameter must be set to blanks.

Allowed values for the "Error detected" parameter are:

**\*CONTINUE**

The trace and the watch for trace event facility will continue running.

**\*STOP**

The trace and the watch for trace event facility will be ended.

**\*ERROR**

Error detected by customer trace program.

Allowed values for the "Comparison data" parameter when \*MSGID is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(7)	Message ID
11 B	CHAR(9)	Reserved
20 14	BINARY(4)	Offset to comparison data
24 18	BINARY(4)	Length of comparison data
* *	CHAR(*)	Message comparison data

Allowed values for the "Comparison data" parameter when \*LICLOG or \*CMPDATA is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(4)	LIC Log major code
8 8	CHAR(4)	LIC Log minor code
12 C	CHAR(8)	LIC Log identifier
20 14	BINARY(4)	Offset to comparison data
24 18	BINARY(4)	Length of comparison data
* *	CHAR(*)	LIC log comparison data

Allowed values for the "Comparison data" parameter when \*ON, \*INTVAL or \*WCHTIMO is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information (always 4).

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at <http://www.iseries.ibm.com/infocenter>.

### Single values

#### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

#### Qualifier 1: Trace program

*name* Specify the name of the trace exit program.

#### Qualifier 2: Library

\*LIBL All libraries in the job's library list are searched until the first match is found.

*name* Specify the name of the library where the user exit program is located.

Top

---

## Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

#### \*NONE

No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

**1-9999** Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

---

## Examples

### Example 1: Start a Trace on Your Job

STRTRC SSNID(\*GEN)

This command starts a trace on the job that called the STRTRC command. A unique trace session identifier will be generated. The session identifier will be returned in the message data of a completion message sent after the trace session starts successfully.

#### Example 2: Start a Trace of Another Job

```
STRTRC JOB(123456/QSYS/QSYSARB) SSNID(ARBTRACE)
```

This command starts a new trace on the job with job name QSYSARB, job user name of QSYS, and job number of 123456. Only this one job is traced. The trace session identifier is ARBTRACE.

#### Example 3: Start a Generic Job Trace

```
STRTRC SSNID(*GEN) JOB(*ALL/QSYS/QTVELNET*) MAXSTG(80000)
```

This command starts a new trace on all jobs on the system that have a job name that begins with QTVELNET and are running under the QSYS user profile. The maximum storage used for the trace will be 80000 kilobytes. A unique session identifier will be generated.

#### Example 4: Start a Job Trace with a Filter

```
ADDTRCFTR FTR(MYFTR) PGMTRG(*EQ MYLIB/MYPGM *ALL *PGM *ENTRY)
STRTRC SSNID(MYTRACE)JOB(123456/MYUSER/MYJOB) TRCFTR(MYFTR)
```

These commands add a trace filter for MYPGM in MYLIB. When the trace is started, call/return trace records will not be collected until MYLIB/MYPGM is called. Once MYLIB/MYPGM is called, the trace will collect call/return trace records until the trace is ended.

#### Example 5: Start a Job Trace with Component Defined

```
STRTRC JOB(123456/MYUSER/MYJOB) SSNID(MYTRACE)
      TRCTYPE(*ENVVAR *VERBOSE)
```

This command starts a new trace on the job with job name MYJOB, job user name of MYUSER, and job number of 123456. Only this one job is traced. The trace session identifier is MYTRACE. The trace will collect call/return trace information as well as information for the ENVVAR component at VERBOSE level.

#### Example 6: Start a Job Trace with Threads Selection

```
STRTRC JOB((123456/MYUSER/MYJOB (1111111 2222222)))
      SSNID(MYTRACE)
```

This command starts a new trace on the job with job name MYJOB, job user name of MYUSER, and job number of 123456. Only threads 1111111 and 2222222 are traced. The trace session identifier is MYTRACE.

#### Example 7: Start a Trace and Watch for a Message to End the Trace

```
STRTRC SSNID(*GEN) WCHMSG((MCH2804))
      WCHMSGQ((*SYSOPR) (*JOBLOG))
      WCHJOB((*ALL/MYUSER/MYJOBNAME))
      TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts a trace on the job that called the STRTRC command. The trace will be ended when MCH2804 message is found on the System Operator message queue or within the \*ALL/MYUSER/MYJOBNAME job log. Also, MYLIB/TRCEXTPGM is specified as a trace exit program.

#### Example 8: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
STRTRC SSNID(*GEN) WCHLICLOG(('99??' 9932 MYJOBNAME))
      WCHTIMO(*NOMAX)
```

This command starts a trace on the job that called the STRTRC command. The trace will be ended when a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 is generated on the system. Also, the LIC log information should contain the text "MYJOBNAME". \*NOMAX on WCHTIMO parameter indicates that the trace will be active until the event occurs or ENDTRC command is issued manually.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPC3921

STRTRC session ID &1 successfully started.

#### CPC3922

STRTRC session ID &1 started but some jobs not found.

#### CPF0006

Errors occurred in command.

#### CPF39C7

STRTRC session ID &1 already exists.

#### CPF39C9

Unexpected STRTRC failure, see low-level messages.

#### CPF39C5

Job name \*ALL and user \*ALL not allowed

#### CPF39C6

Not authorized to trace with generic job users.

#### CPF39CC

No active jobs found, STRTRC session not started.

#### CPI36CE

Job &3/&2/&1 not found.

#### CPF39F2

Cannot allocate library &1

#### CPF98A2

Not authorized to &1 command.

#### CPF39D3

Unable to start/end the trace.

Top



---

## Start Trap Manager (STRTRPMGR)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

Use the Start Trap Manager (STRTRPMGR) command to start the OS/400 SNMP trap manager. An optional Forward Trap parameter may be specified, which enables traps that are received on the system to be forwarded to other Network Management stations, as configured in the Simple Network Management Protocol (SNMP) agent and Distributed Protocol Interface (DPI) interface.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FWDTRP	Forward traps	*YES, <u>*NO</u>	Optional, Positional 1

Top

---

### Forward Trap (FWDTRP)

Specifies whether traps received on the system are to be forwarded to other network management stations.

The possible values are:

\*NO Received traps will not be forwarded. Traps will only be enqueued.

\*YES Received traps will be forwarded using the facilities provided in the SNMP agent and DPI interface.

Top

---

### Examples

#### Example 1: Start Trap Manager Job (Enqueue Traps Only)

```
STRTRPMGR
```

This command starts the trap manager job. Traps received by the trap manager are enqueued only.

#### Example 2: Start Trap Manager Job (Enqueue & Forward Traps)

```
STRTRPMGR FWDTRP(*YES)
```

This command starts the trap manager job. Traps received by the trap manager are enqueued and forwarded.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA80D

Trap manager job already active.

#### CPFA802

Trap manager not started.

[Top](#)



---

## Transfer Batch Job (TFRBCHJOB)

### Where allowed to run:

- Batch job (\*BATCH)
- Batch program (\*BPGM)
- Batch REXX procedure (\*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (\*EXEC)

**Threadsafe:** No

Parameters  
Examples  
Error messages

The Transfer Batch Job (TFRBCHJOB) command transfers a batch job to the specified job queue. The job queue does not have to be allocated to an active subsystem at the time of the batch job transfer. The batch job that is transferred is the one in which this command is issued. Routing data and request data can be specified for the batch job when it is transferred. The routing data specified is processed in the subsystem in which the job queue is active. The request data follows other request data for the job. The transferred batch job resumes running the request data following the transfer.

### Notes:

1. Running this command causes loss of spooled inline files because they cannot be accessed in the new routing step.
2. If you are working in a System/36 environment, the TFRBCHJOB command does not transfer the System/36 environment to the new routing step.
3. If objects allocated to the previous routing step, are needed in the new routing step, they must be allocated again. If files opened in the previous routing step, are needed in the new routing step, they must be opened again.
4. If the TFRBCHJOB command is issued in a CL program, all subsequent commands in the CL program are bypassed.
5. A batch job transferred to a job queue by the TFRBCHJOB command exists through an initial program load (IPL) if the batch job was residing on the job queue at the time the system was powered down. A batch job's temporary objects are destroyed during the power down.
6. The QTEMP library of a batch job that has been transferred by the TFRBCHJOB command is always empty when the next routing step is started. Caution must be used with the library list in conjunction with a batch job that was transferred to a job queue by the TFRBCHJOB command. The TFRBCHJOB function saves the library list to recover the job on a job queue if an IPL occurs. When the routing step for the transferred batch job is started, the libraries in the saved library list must exist in the system or the job's routing step ends.

### Restrictions:

1. To use this command, you must have:
  - use (\*USE) authority to the job queue and execute (\*EXECUTE) authority to the library that contains that job queue.
2. The job being transferred must be a batch job that started from a job queue.
3. The TFRBCHJOB command is not allowed to run in a batch communications job (a batch job that was started as the result of a program start request) or a batch immediate job.

Top

---

## Parameters

Keyword	Description	Choices	Notes
JOBQ	Job queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Job queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
RTGDTA	Routing data	<i>Character value, QCMDB, *RQSDTA</i>	Optional
RQSDTA	Request data or command	<i>Character value, *NONE, *RTGDTA</i>	Optional

Top

---

### Job queue (JOBQ)

Specifies the qualified name of the job queue to which the job is transferred.

This is a required parameter.

#### Qualifier 1: Job queue

*name* Specify the name of the job queue.

#### Qualifier 2: Library

**\*LIBL** All libraries in the thread's library list are searched until a match is found.

#### \*CURLIB

The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the library where the job queue is located.

Top

---

### Routing data (RTGDTA)

Specifies the routing data used to start the next routing step in the job. The routing data is used to determine the routing entry that identifies the program that the job runs.

#### QCMDB

This routing data matches a routing entry in the IBM-supplied subsystem description, which starts a routing step processed by the IBM-supplied control language processor QCMD.

#### \*RQSDTA

The first 80 characters of the request data specified in the RQSDTA parameter of this command are used as the routing data for the routing step.

#### *character-value*

Specify the character string that is used as the routing data for starting the routing step. A maximum of 80 characters can be entered, enclosed in apostrophes if necessary.

Top

---

### Request data or command (RQSDTA)

Specifies the request data that is placed as the last entry in this job's message queue. The request data can be a CL command to be run or a string of characters used by another program.

### \*NONE

No request data is placed in the job's message queue.

### \*RTGDTA

The routing data specified in the **Routing data (RTGDTA)** parameter is placed at the end of the job's message queue.

### *character-value*

Specify the character string that is placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered, enclosed in apostrophes if necessary.

Top

---

## Examples

```
TFRBCHJOB  JOBQ(QGPL/APPLICQ)  RTGDTA(APPLICS)
```

This command transfers the batch job in which the command is entered to the APPLICQ job queue that is in the QGPL library. The job is routed using the routing data APPLICS. The job must be a batch job.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1288

Job queue &1 in library &2 damaged.

#### CPF1289

Transfer job is not allowed.

#### CPF1291

Job &3/&2/&1 cannot be transferred.

#### CPF1368

&1 not authorized to job queue &2 in library &3.

#### CPF1369

Job queue &1 in &2 not found.

#### CPF1370

Job queue &1 in &2 not accessible.

#### CPF1372

Job not transferred. Job currently being ended.

#### CPF1375

Job not transferred. Single active job not allowed to transfer.

#### CPF1376

Library on library search list deleted.

#### CPF1377

Library on library search list damaged.

Top



---

## Transfer Control (TFRCTL)

### Where allowed to run:

- Batch program (\*BPGM)
- Interactive program (\*IPGM)

Threadsafe: Yes

Parameters  
Examples  
Error messages

The Transfer Control (TFRCTL) command calls the specified program, passes control to it, and removes the transferring program from the return stack. Because the transferring program is removed from the call stack, control does not return to it when the called program returns control. Instead, control is returned to the command following the last call to the transferring program.

### Restrictions:

- This command is valid only within original program model (OPM) CL programs.
- You must have object operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the program to be called, and \*EXECUTE authority to the library where the program is located.
- The TFRCTL command is threadsafe, meaning that it can be used to call a program when the TFRCTL command is run in a job with multiple threads. No checking is done whether or not the program to be called is threadsafe.

Top

---

## Parameters

Keyword	Description	Choices	Notes
PGM	Program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
PARM	Parameter CL variable names	Values (up to 255 repetitions): <i>CL variable name</i>	Optional, Positional 2

Top

---

## Program (PGM)

Specifies the program that receives control from the program transferring control.

This is a required parameter.

### Qualifier 1: Program

*name* Specify the name of the program.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### \*CURLIB

The current library for the thread is used to locate the program. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the name of the library where the program is located.

Top

---

## Parameter CL variable names (PARM)

Specifies one or more CL variables to be passed to the program that is to receive control. The variables passed can only be parameters that were passed to the program currently transferring control.

### *CL-variable-name*

Specify the name of the CL variable to be passed. A maximum of 255 variables can be specified.

Top

---

## Examples

```
TFRCTL PGM(PROGA) PARM(&PARM1)
```

This command transfers control to the program PROGA and passes the parameter &PARM1 to it. The parameter &PARM1 must previously have been passed to the program issuing this command.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF0805

Error found when program &1 in &2 started.

#### CPF0809

Transfer control (TFRCTL) to C program not allowed.

Top

---

## Transfer to Group Job (TFRGRPJOB)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Transfer to Group Job (TFRGRPJOB) command suspends the job that issued the TFRGRPJOB command, and the group job specified by the **Group job (GRPJOB)** parameter is resumed (if it already exists) or is created (if it does not exist). In both cases, control is transferred to the job specified by the GRPJOB parameter. The job issuing the TFRGRPJOB command remains suspended until control is passed back to it and the job is resumed.

Top

---

### Parameters

Keyword	Description	Choices	Notes
GRPJOB	Group job	Name, <u>*PRV</u> , *SELECT	Optional, Positional 1
INLGRPPGM	Initial group program	Qualified object name	Optional, Positional 2
	Qualifier 1: Initial group program	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
SPCENV	Special environment	<u>*DFT</u> , *INLGRPPGM, *S36, *NONE	Optional
RSTDSP	Restore display	<u>*NO</u> , *YES	Optional
TEXT	Text 'description'	Character value, <u>*BLANK</u>	Optional

Top

---

### Group job (GRPJOB)

Specifies the group job to which control is transferred.

**\*PRV** Control is transferred to the previously active job in the group. If the previously active job no longer exists, then the most recently active job in the group is resumed. This special value is valid only if there is another group job in the group.

**\*SELECT**

The Group Job Selection display is shown. You can choose which group job to transfer to or create a new group job and transfer to it.

**name** Specify the name of the group job to which control is transferred.

Top

---

### Initial group program (INLGRPPGM)

Specifies the qualified name of the job's first group program. This parameter only has meaning when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

#### Qualifier 1: Initial group program

*name* Specify the name of the job's first group program.

### Qualifier 2: Library

**\*LIBL** All libraries in the thread's library list are searched until a match is found.

**\*CURLIB**

The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the library where the program is located.

Top

---

## Special environment (SPCENV)

Specifies the environment in which the group job starts. This parameter is valid only when this command creates a group job. If control is transferring to an existing group, this parameter is ignored.

**\*DFT** The group job starts in the environment in which the command is run. The group job starts in the System/36 environment if one of the following is true:

- The System/36 environment is active in the job in which this command is running
- The user profile specifies that the user runs in the System/36 environment, and the first program called in the group job is QCMD.

**\*INLGRPPGM**

The new group starts in the environment determined by the first group called in the group job. If the first group program is QCMD, the special environment value in the user profile is used to determine the environment.

**\*S36** The new group starts in the System/36 environment.

**\*NONE**

The new group does not start in any special environment.

Top

---

## Restore display (RSTDSP)

Specifies whether data being shown at a display device by this display file is saved at the time the file is suspended (made temporarily inactive) so that a different display file can be used to show different data on the same device.

**\*NO** The data being shown by this file is not saved when the file is suspended.

**\*YES** The data being shown when the file is suspended is saved so it can be restored to the display of the device when the file is used again.

Top

---

## Text 'description' (TEXT)

Specifies text that describes the group job. This parameter only has meaning when a group job is created. If the group job being transferred to already exists, this parameter is ignored.

**\*BLANK**

The text is all blanks.



### *character-value*

Specify a maximum of 50 characters of text, enclosed in apostrophes if necessary.

Top

---

## Examples

```
TFRGRPJOB  GRPJOB(GROUPJ1)  INLGRPPGM(QGPL/PROGRAM1)
```

This command suspends running of the current job. If group job GROUPJ1 already exists, it is resumed at the point where it was suspended (the next high-level language command following the TFRGRPJOB request).

If group job GROUPJ1 does not exist, group job GROUPJ1 is created and runs the program PROGRAM1 in library QGPL.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1E15

Problem occurred while calling Operational Assistant.

#### CPF1310

Request to transfer to group job failed with reason code &1.

#### CPF1313

Value &1 for parameter &2 not allowed name.

#### CPF1314

Value &1 for parameter &2 not allowed.

#### CPF9871

Error occurred while processing.

Top



---

## Transfer Job (TFRJOB)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Transfer Job (TFRJOB) command transfers a job to the specified job queue. The job that is transferred is the one where this command is issued. If the job being transferred is an interactive job, it is given the highest priority on the job queue. New routing data and request data can be specified for the job when it is transferred.

If objects allocated to the previous routing step are needed in the new routing step, they must be allocated again. If files opened in the previous routing step are needed in the new routing step, they must be opened again.

### Notes:

1. Running this command in a batch job causes loss of spooled inline files because they cannot be accessed in the new routing step.
2. If the target subsystem is ended (by running the End Subsystem (ENDSBS) command, the End System (ENDSYS) command, or the Power Down System (PWRDWNSYS) command) while an interactive transferring job is on a job queue, the job is canceled as part of subsystem ending.
3. Because a PWRDWNSYS command inhibits new jobs or routing steps from being started by any subsystem, a batch job transferred to a job queue (by the TFRJOB command) is not completed before the system is powered down.
4. The temporary objects associated with a transferring job (such as the library list, the QTEMP library, and all objects in it) are destroyed during the PWRDWNSYS command, so that during a re-initial program load (IPL), the system is unable to restore the job to its previous state. During re-IPL, the system removes the job from the job queue and produces its job log.
5. If the TFRJOB command is issued in a CL program, all subsequent commands in the CL program are bypassed.

### Restrictions:

1. To use this command, you must have:
  - use (\*USE) authority to the job queue and execute (\*EXECUTE) authority to the library that contains that job queue.
  - use (\*USE) authority to the subsystem description associated with the subsystem that has the job queue allocated and execute (\*EXECUTE) authority to the library that contains that subsystem description. If the job being transferred is a batch job and the job queue is not allocated to a subsystem description, this restriction does not apply.
2. If the job being transferred is an interactive job, the following restrictions apply:
  - The job queue on which the job is placed must be associated with an active subsystem.
  - The work station associated with the job must have a corresponding work station entry in the subsystem description associated with the new subsystem.
  - The work station associated with the job must not have another job associated with it that has been suspended by means of the Sys Req (system request) key. The suspended job must be canceled before the Transfer Job command can be run.
  - The job must not be a group job.
3. The job must not be a communications batch job (started as a result of a program start request), unless it meets one of the following criteria:

- It was started from an APPC communications device.
- The session on the communications device has ended.

Top

---

## Parameters

Keyword	Description	Choices	Notes
JOBQ	Job queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Job queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
RTGDTA	Routing data	<i>Character value, QCMDI, *RQSDTA</i>	Optional
RQSDTA	Request data or command	<i>Character value, *NONE, *RTGDTA</i>	Optional

Top

---

## Job queue (JOBQ)

Specifies the qualified name of the job queue to which the job is transferred.

This is a required parameter.

### Qualifier 1: Job queue

*name* Specify the name of the job queue.

### Qualifier 2: Library

**\*LIBL** All libraries in the thread's library list are searched until a match is found.

### \*CURLIB

The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the library where the job queue is located.

Top

---

## Routing data (RTGDTA)

Specifies the routing data used to start the next routing step in the job. The routing data is used to determine the routing entry that identifies the program that the job runs.

### QCMDI

This routing data matches a routing entry in the IBM-supplied subsystem descriptions, which starts a routing step processed by the IBM-supplied control language processor, QCMD, in the QSYS library.

### \*RQSDTA

The first 80 characters of the request data specified in the **Request data or command (RQSDTA)** parameter of this command are also used as the routing data for the routing step.

### *character-value*

Specify the character string that is used as the routing data for starting the routing step. A maximum of 80 characters can be entered, enclosed in apostrophes if necessary.

---

## Request data or command (RQSDTA)

Specifies the request data that is placed as the last entry in this job's message queue. The request data can be a CL command to be run or a string of characters used by another program.

### \*NONE

No request data is placed in the job's message queue.

### \*RTGDTA

The routing data specified in the **Routing data (RTGDTA)** parameter is also placed at the end of the job's message queue.

### *character-value*

Specify the character string that is placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered, enclosed in apostrophes if necessary.

---

## Examples

```
TFRJOB JOB(QGPL/APPLICQ) RTGDTA(APPLICS)
```

This command transfers the job in which the command is entered to the APPLICQ job queue in the QGPL library. The job is routed using the routing data APPLICS. If the job is an interactive job, the job queue must be allocated by an active subsystem.

---

## Error messages

### \*ESCAPE Messages

#### CPF1289

Transfer job is not allowed.

#### CPF1315

Command &1 not allowed in this environment.

#### CPF1357

Job not transferred.

#### CPF1364

Job not transferred. Job queue &1 in library &2 not active.

#### CPF1365

Job not transferred. Subsystem &1 ending.

#### CPF1366

Subsystem &1 has no usable work station entry for &2.

#### CPF1367

User &1 not authorized to subsystem &2

#### CPF1368

&1 not authorized to job queue &2 in library &3.

**CPF1369**

Job queue &1 in &2 not found.

**CPF1370**

Job queue &1 in &2 not accessible.

**CPF1372**

Job not transferred. Job currently being ended.

**CPF1373**

Job not transferred. System request in effect for job.

**CPF1375**

Job not transferred. Single active job not allowed to transfer.

[Top](#)

---

## Transfer Pass-Through (TFRPASTHR)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Transfer Pass-Through (TFRPASTHR) command allows you to transfer from a pass-through system to a source system. It performs the same function as a System Request (SYS REQ) option 10, 11, 13, or 14, and is valid only on a target pass-through system.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TOJOB	To job	<u>*SRC</u> , *ALT, *HOME, *HOMEALT	Optional, Positional 1

Top

---

### To job (TOJOB)

Specifies the program that is given control when you are transferred to the home system or the previous system.

The possible values are:

- \*SRC** The job on the current system is suspended, and control is transferred back to the program specified on the System request program prompt (SRQ10PGM parameter) of the Start Pass-Through (STRPASTHR) command on the previous system. When the specified program ends, the target system gets control.
- \*ALT** The job on the target system is suspended, and control is transferred back to the alternate job on the previous system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.
- \*HOME**  
The job on the target system is suspended, and control is transferred back to the program specified on the System request program prompt (SRQ10PGM parameter) of the Start Pass-Through (STRPASTHR) command on the home system. When the specified program ends, the target system gets control.
- \*HOMEALT**  
The job on the target system is suspended, and control is transferred back to the alternate job on the home system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.

Top

---

## Examples

TFRPASTHR TOJOB(\*HOME)

This command transfers control back to the source job on the home system.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Transfer Secondary Job (TFRSECJOB)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Transfer Secondary Job (TFRSECJOB) command creates a secondary interactive job at your work station, then transfers control between the primary and secondary jobs. The first time you issue this command, you receive the sign-on prompt for the secondary job. Once you sign on, a secondary job is created, allowing you to receive the basic working display of the new job. Your primary job remains suspended as long as you remain in your secondary job. The next time you issue the TFRSECJOB command, your current job is suspended, and you return to the first job at the point at which you left it. When you sign off either job, you are automatically returned to the remaining job.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

TFRSECJOB

This command causes the job that is currently running to be suspended. If a secondary job does not exist, the SIGNON prompt is displayed. Otherwise, the secondary job resumes running.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF1380

Transfer to secondary interactive job not valid.

##### CPF1381

Transfer to secondary interactive job not valid.

##### CPF1383

Transfer to secondary interactive job not valid.

##### CPF1384

Transfer to a secondary interactive job not valid.

[Top](#)



## Trace TCP/IP Route (TRACEROUTE)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Trace TCP/IP Route (TRCTCPRTE) command, also known as TRACEROUTE, traces the route of IP packets to a user-specified destination system. The route can involve many different systems along the way. Each system along the route is referred to as a **hop**. You can trace all hops along the route or specify the starting and ending hops to be traced.

The route is traced by sending packets (called **probes**) to the destination system. Each probe contains an upper limit (called **Time To Live** or **TTL**) on the number of hop systems the probe can pass through.

**Note:** In IP Version 6, **Time To Live** (TTL) is called the **hop limit**.

A route is traced by successively incrementing the TTL of the probe packets by one hop. The trace ends when either a probe response is received from the destination system or when the probe Time To Live value equals the maximum allowed.

Responses from the probe packets are sent as messages to the job log or as queue entries to a user-specified data queue.

Top

### Parameters

Keyword	Description	Choices	Notes
RMTSYS	Remote system	<i>Character value</i>	Required, Positional 1
RANGE	Range of hops to probe	<i>Element list</i>	Optional
	Element 1: Starting probe TTL (hop limit)	1-255, <u>1</u>	
	Element 2: Maximum probe TTL (hop limit)	1-255, <u>30</u>	
PROBES	Probes sent per hop	1-64, <u>3</u>	Optional
WAITTIME	Response wait time	1-120, <u>3</u>	Optional
PKTLEN	Packet length (in bytes)	40-65535, <u>40</u>	Optional
OUTPUT	Output	<u>*MSG</u> , *VERBOSE, *DTAQ	Optional
DTAQ	Data queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Data queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *CURLIB, *LIBL	
ADRVERFMT	Address version format	*CALC, *IP4, *IP6	Optional
LCLINTNETA	Source IP address	<i>Character value</i> , *ANY	Optional
RMTPORT	Base remote port	1-65535, <u>33434</u>	Optional
NAMELOOKUP	Lookup host names	*YES, *NO	Optional
PROBEPCL	Probing protocol	*ICMP, *UDP	Optional
FRAGMENT	Allow fragmentation	*TCPA, *NO, *YES	Optional

---

## Remote system (RMTSYS)

Specifies the remote system name (255 characters) or IP address of the destination system.

### *character-value*

Specify the remote system name or IP address. Either a valid IP Version 4 or IP Version 6 address will be accepted.

Top

---

## Range of hops to probe (RANGE)

Specifies the range of hop systems from which probe responses are expected. Each probe specifies a TTL (Time To Live) integer value. This TTL value is the maximum number of hops the probe can traverse. For example, a probe packet with a TTL of 3 can pass through at most 3 hop systems before the hop system discards the probe and sends information back to the system from which the probe originated.

Element 1 specifies the first TTL value sent in probe packets. Element 2 specifies the last TTL value sent in probe packets. Trace information is generated from each hop system which discards a probe packet because the TTL value in the probe is reached or when the destination system is reached.

### Element 1: Starting probe TTL (hop limit)

1 The default starting hop is 1.

1-255 Specify the first hop limit TTL number used for probe packets.

### Element 2: Maximum probe TTL (hop limit)

30 The default ending hop is 30.

1-255 Specify the maximum number of hops a probe can traverse to reach the destination system.

Top

---

## Probes sent per hop (PROBES)

Specifies the number of probe packets sent to each hop system for each probe TTL (hop limit) value in the range specified by the RANGE parameter.

3 The default number of probes is three.

1-64 Specify the number of probes to send.

Top

---

## Response wait time (WAITTIME)

Specifies the maximum time, in seconds, to wait for a response from a hop system to each probe.

3 Wait up to 3 seconds for a response.

1-120 Specify the maximum number of seconds to wait for a response.

Top

---

## Packet length (in bytes) (PKTLEN)

Specifies the total length, in bytes, of the IP packet sent for each probe.

**40** The probe packet length is 40 bytes.

**40-65535**  
Specify the number of bytes in the probe IP packet.

Top

---

## Output (OUTPUT)

Specifies where the results obtained from sending the probe packets is sent. Information is sent for each hop until the destination system is reached, including hop count, average round-trip time, IP address of the hop and host name of the hop.

**\*MSG** Results are output as messages sent to the job log of the job in which the command is issued.

**\*VERBOSE**

Results are output as messages sent to the job log of the job in which the command is issued. All responses received are displayed. Results are not limited to ICMP TIME\_EXCEEDED and PORT\_UNREACHABLE responses.

**\*DTAQ**

Results from probes are placed on the data queue specified by the Data Queue (DTAQ) parameter.

Top

---

## Data queue (DTAQ)

Specifies the data queue on which entries are placed. When a data queue is specified, messages are not sent to the job log unless an error occurs.

Each queue entry contains the response to a probe if one was received or indicates that no probe response was received. The specified data queue must have a queue entry length of at least 32 characters and must exist when this command is issued.

### Qualifier 1: Data queue

*name* Specify the name of the data queue.

### Qualifier 2: Library

**\*LIBL** All libraries in the job's library list are searched.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to search.

Top

---

## Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

### \*CALC

The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. TRCTCPRTE (TRACEROUTE) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution is used in determining the IP address.

**\*IP4** Use the IP Version 4 host name resolution method.

**\*IP6** Use the IP Version 6 host name resolution method.

Top

---

## Source IP address (LCLINTNETA)

Specifies how the source IP address in the probe packet is chosen.

**\*ANY** The source IP address in the probe packets is chosen by the system. The system may use any active local interface which can reach the remote system.

### *character-value*

Specify the local interface to use as the source IP address.

Top

---

## Base remote port (RMTPORT)

Specifies the base UDP port number used in probes.

**33434** Use the default base UDP port number of 33434.

**1-65535**

Specify the base UDP port number to be used in probes.

Top

---

## Lookup host names (NAMELOOKUP)

Specifies whether IP addresses will be resolved to the host name.

**\*YES** The address will be resolved to the host name.

**\*NO** The address will not be resolved to the host name.

Top

---

## Probing protocol (PROBEPCL)

Specifies the protocol used when sending probe packets.

### \*ICMP

The probes sent to the destination system are ICMP (Internet Control Message Protocol) Echo Request packets.

**\*UDP** The probes sent to the destination system are UDP (User Datagram Protocol) packets.

Top

---

## Allow fragmentation (FRAGMENT)

Specifies how the setting of the "Do Not Fragment" option in the IP header of the probe packet is determined.

### \*TCPA

The system sets the option based on the setting of the IP Path MTU Discovery TCP/IP attribute.

**Note:** Use the Change TCP/IP Attributes (CHGTCPA) command to change the value of this attribute.

**\*NO** The "Do Not Fragment" option is always specified.

**\*YES** The "Do Not Fragment" option is never specified.

Top

---

## Examples

### Example 1: Trace Entire Route

```
TRCTCPRTE  RMTSYS('130.14.3.5')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is '130.14.3.5'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP Echo Request packet. Results received are sent as messages to the job log.

### Example 2: Trace Partial Route

```
TRCTCPRTE  RMTSYS('AAA.BBBB.COM')  RANGE(3 7)
           PROBES(5)  PROBEPL(*UDP)
           OUTPUT(*DTAQ)  DTAQ(MYLIB/MYDATAQ)
```

This command traces the route between the local iSeries system and the destination system whose host name is 'AAA.BBBB.COM'. Five probe packets will be sent for the starting range value of 3. Each probe will be a UDP packet inside an IP packet that is 40 bytes long. Each of these 5 probes will specify a TTL of 3. If system AAA.BBB.COM can be reached by passing through at most 2 hop systems then the trace will terminate at this point.

If system AAA.BBB.COM is further than 2 hops, another set of 5 probe packets will be sent to the destination AAA.BBB.COM. Each of these 5 probes will specify a TTL of 4. This is repeated until either system AAA.BBB.COM responds to a probe or 5 probes with a TTL of 7, the ending range value, are sent. Any results received are placed as queue entries on data queue MYDATAQ in library MYLIB.

### Example 3: Trace Route with an IP Version 6 Address

```
TRCTCPRTE  RMTSYS('1:2:3:4:5:6:7:8')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is 1:2:3:4:5:6:7:8. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

**Note:** A colon character (:) found in the parameter value signifies an IP Version 6 address and will cause an ICMP6 echo request packet to be generated.

### Example 4: Trace Route with an IP Version 6 Host Name

```
TRCTCPRTE  RMTSYS('IP6HOST')
```

This command traces the entire route between the local iSeries system and the destination system whose host name is 'IP6HOST'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

The default "Address version format" is \*CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (\*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to trace the route.

#### **Example 5: Trace Route with an IP Version 6 Host Name and Explicitly Use IP Version 6 Host Name Resolution**

```
TRCTCP RTE RMTSYS('IP6HOST') ADRVERFMT(*IP6)
```

This command traces the entire route between the local iSeries system and the destination system whose host name is 'IP6HOST'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

This example differs from example 4 in that only a valid IP version 6 resolved address, for IP6HOST, will be used when attempting to trace the route.

Top

---

## **Error messages**

### \*ESCAPE Messages

#### **TCP3250**

DTAQ parameter value required with OUTPUT(\*DTAQ).

#### **TCP3251**

DTAQ parameter not valid when OUTPUT(\*MSG) specified.

#### **TCP3252**

Starting range value greater than range limit.

Top



---

## Trace ASP Balance (TRCASPBAL)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Trace ASP Balance (TRCASPBAL) command controls the function that gathers the auxiliary storage pool (ASP) usage statistics. The trace function monitors the frequency that data is accessed on the disk units within the specified ASP. The 'high' use data and the 'low' use data on the units is identified. The tracing of the usage of data on the units can be started on a specific ASP or for multiple ASPs. The trace may be started for a specific length of time. The trace can be stopped by specifying \*OFF for the **Trace option setting (SET)** parameter. The trace can be ended at any time and restarted at a later time. The statistics that are collected are cumulative. For example, if the trace is started and ended and then restarted without clearing the statistics, the second group of statistics are added to the first collection.

After statistics have been collected the ASP may be balanced using the Start ASP Balance (STRASPBAL) command, specifying TYPE(\*USAGE) or TYPE(\*HSM). After the balance has run to completion, the statistics will be cleared automatically by the balance function.

The balancing of the ASP should be done shortly after the statistics have been collected. The usefulness of the balance is diminished as the trace statistics age. If the statistics are not current, the statistics may be cleared by specifying \*CLEAR for the SET parameter.

A message will be sent to the system history (QHST) log when the trace function is turned on, when it is stopped, or the trace data is cleared.

For more information about ASP balancing, see the Hierarchical Storage Management Use, SC41-5351.

### Restrictions:

- You must have all object (\*ALLOBJ) special authority to run this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	*ON, *OFF, *CLEAR	Optional
ASP	ASP number	Single values: *ALL Other values (up to 32 repetitions): 1-32	Optional, Positional 1
ASPDEV	ASP device	Single values: *ALLAVL Other values (up to 32 repetitions): <i>Name</i>	Optional, Positional 2
TIMLMT	Time limit	1-9999, *NOMAX	Optional

Top

---

## Trace option setting (SET)

Specifies whether to start collecting statistics, end the collection of statistics, or delete previously collected usage statistics for an auxiliary storage pool (ASP).

**Note:** A value must always be specified for this parameter.

**\*ON** The tracing of the statistics will be started.

**\*OFF** The tracing of the statistics will be ended.

**\*CLEAR**

The statistics for the specified ASP will be cleared.

Top

---

## ASP number (ASP)

Specifies the auxiliary storage pool (ASP) for which the ASP tracing function will be started, ended, or cleared.

**Note:** A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter.

### Single values

**\*ALL** ASP tracing will be started, ended, or cleared for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

### Other values (up to 32 repetitions)

**1-32** Specify the number of the ASP for which ASP tracing is to be started, ended, or cleared.

Top

---

## ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device for which the tracing function will be started, ended, or cleared.

**Note:** A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter.

### Single values

**\*ALLAVL**

ASP tracing will be started, ended, or cleared for all ASP devices that currently have a status of 'Available'.

### Other values (up to 32 repetitions)

**name** Specify the name of the independent ASP device for which ASP balancing is to be started.

Top

---

## Time limit (TIMLMT)

Specifies the amount of time, in minutes, that the ASP balancing function will be allowed to run. When the time limit is reached the function will end. The trace function will not run across an IPL.

**Note:** A value must be specified for this parameter if \*ON is specified for the **Trace option setting (SET)** parameter.

**1-9999** Specify the number of minutes that the trace function will be allowed to run.

Top

---

## Examples

### Example 1: Start Trace for ASP 1

```
TRCASPBAL ASP(1) SET(*ON) TIMLMT(9999)
```

This command allows the user to start the ASP tracing function for ASP 1. This function will run until the user ends the trace or 9999 minutes have passed.

### Example 2: End Tracing for All ASPs

```
TRCASPBAL ASP(*ALL) SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP that currently has a trace running.

### Example 3: Clear the Trace Data for ASP 1

```
TRCASPBAL ASP(1) SET(*CLEAR)
```

This command allows the user to clear the trace data for ASP 1.

### Example 4: End Tracing for All ASP Devices

```
TRCASPBAL ASPDEV(*ALLAVL) SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP device that currently has a trace running.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1890

\*ALLOBJ authority required for requested operation.

#### CPF18A9

ASP tracing for ASP &1 already started.

#### CPF18AA

ASP tracing not active for ASP &1.

#### CPF18AD

ASP &1 must contain more than a single unit.

#### CPF18AE

ASP &1 does not contain trace data.

**CPF18B1**

Trace function currently running for ASP &1.

**CPF18B2**

Balance function running for ASP &1.

**CPF9829**

Auxiliary storage pool &1 not found.

[Top](#)

---

## Trace Connection (TRCCNN)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Trace Connection (TRCCNN) command allows the tracing of encrypted data flowing over internet protocol (IP) and Secure Sockets Layer (SSL) connections. Specific types of traces are started and stopped by using this command.

TRCCNN uses the Trace Internal (TRCINT) command to collect the trace records and generate an intermediate spooled file named QPCSMPRT. The QPCSMPRT spooled file data is used to generate a spooled file named QSYSPRT. The user data for the QSYSPRT file is 'TRCCNN'.

You can also use TRCCNN with a QPCSMPRT spooled file generated by using TRCINT directly. TRCCNN can extract and format the IP and SSL connection-related trace records. This allows you to use TRCINT to collect many types of trace records and then use TRCCNN to format the subset of trace records related to IP or SSL connections.

### Restrictions:

- To use this command, you must have service (\*SERVICE) special authority, or be authorized to the Service Trace function of OS/400 through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- The following user profiles have private authorities to use the command:
  - QSRV
- When the WCHJOB parameter is specified, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (\*JOBCTL) special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	*ON, *OFF, *END, *FORMAT	Required, Positional 1
TRCTYPE	Trace type	Values (up to 2 repetitions): *IP, *SSL	Optional
TRCFULL	Trace full	*WRAP, *STOPTRC	Optional
TRCTBL	Trace table name	Character value, *GEN	Optional
SIZE	Size	Single values: *MAX, *MIN Other values: <i>Element list</i>	Optional
	Element 1: Number of units	1-998000, <b>16000</b>	
	Element 2: Unit of measure	*KB, *MB	
FMTDTA	Trace data to be formatted	72-99999, *CALC	Optional
CCSID	Coded character set identifier	1-65533, *EBCDIC, *ASCII	Optional

Keyword	Description	Choices	Notes
JOB	Job name	Single values: * Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, *ONLY, *LAST	Optional
JOBSYSNAME	Job system name	<i>Name</i> , *ONLY, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: *ONLY, *LAST Other values: <i>Element list</i>	Optional
	Element 1: Creation date	<i>Date</i>	
	Element 2: Creation time	<i>Time</i> , *ONLY, *LAST	
TCPDTA	TCP/IP data	<i>Element list</i>	Optional
	Element 1: Protocol	*TCP, *UDP, *ICMP, *IGMP, *ARP, *ICMP6	
	Element 2: Local port	Values (up to 2 repetitions): 1-65535	
	Element 3: Remote port	Values (up to 2 repetitions): 1-65535	
	Element 4: Local IP address	<i>Character value</i>	
	Element 5: Remote IP address	<i>Character value</i>	
	Element 6: Line description	<i>Name</i>	
WCHMSG	Watch for message	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Message identifier	<i>Name</i>	
	Element 2: Comparison data	<i>Character value</i> , *NONE	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional
	Element 1: Message queue	Single values: *SYSOPR, *JOBLOG, *HSTLOG Other values: <i>Qualified object name</i>	
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL	
WCHJOB	Watched job	Single values: * Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000001-999999, *ALL	
WCHLICLOG	Watch for LIC log entry	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Major code	<i>Character value</i> , *ALL	
	Element 2: Minor code	<i>Character value</i> , *ALL	
	Element 3: Comparison data	<i>Character value</i> , *NONE	
WCHTIMO	Length of time to watch	1-43200, <b>1440</b> , *NOMAX	Optional
TRCPGM	Trace program	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Trace program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL	
TRCPMITV	Time interval	1-9999, *NONE	Optional

Top

---

## Trace option setting (SET)

Specifies whether tracing is started, stopped or ended. Also, you can select to format trace record data collected previously using the TRCCNN or TRCINT (Trace Internal) command.

This is a required parameter.

- \*ON** The collection of internal trace records is started for the trace types specified in the TRCTYPE parameter. If \*GEN is specified in the TRCTBL parameter then the trace table name will be QTRCCNNxxxxxx where xxxxxx is the job number of the current job. Otherwise the trace table name will be the name specified on the TRCTBL parameter.
- \*OFF** Collection of trace records stops. A spooled file named QPCSMPRT is generated by the TRCINT command and contains the collected trace record data. TRCCNN formats this data in a second spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. The trace table is deleted after the spooled files are generated.
- \*END** Collection of trace records stops and the trace table is deleted. No spooled output is generated.
- \*FORMAT**  
Formats trace data in a QPCSMPRT spooled file created by a previous invocation of TRCCNN or TRCINT. The formatted data is written to a spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. Use the TRCTYPE parameter to specify which connection-related trace records to format. Use the JOB, SPLNBR, JOBSYSNAME and CRTDATE parameters to identify which QPCSMPRT file to use.

Top

---

## Trace type (TRCTYPE)

If SET(\*ON) is specified, identifies the types of trace records to start collecting. If SET(\*FORMAT) is specified, identifies the types of collected trace records to format. Multiple trace types may be specified.

- \*IP** Trace IP (internet protocol) data.
- \*SSL** Trace SSL (Secure Sockets Layer) connection data.

Top

---

## Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or stop tracing when the trace table is full.

- \*WRAP**  
When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.
- \*STOPTRC**  
Tracing is stopped when the trace table is full of trace records.

Top

---

## Trace table name (TRCTBL)

Specifies the trace table to hold the collected trace data.

- \*GEN** The trace table name will be QTRCCNNxxxxxx where xxxxxx is the job number of the current job.

*name* Specify the name of the trace table to be used. If SET(\*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created.

Top

---

## Size (SIZE)

Specifies the size of the trace table. The amount of storage to be allocated can be specified in units of kilobytes (\*KB) or megabytes (\*MB). If the size is specified in kilobytes, the amount of storage allocated for the table will be rounded up to the nearest megabyte. Valid table size values range from one megabyte to 258048 megabytes.

**Note:** The amount of storage specified by this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage space is not dynamically allocated as it is needed. This storage space will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

**Note:** If tracing data over a gigabit Ethernet line the trace table size should be greater than 128000 kilobytes or 128 megabytes.

### Single values

**\*MAX** The trace table is set to the maximum size of 258048 megabytes.

**\*MIN** The trace table is set to the minimum size of one megabyte.

### Element 1: Number of units

**16000** The trace table size is either 16000 kilobytes or 16000 megabytes, depending on the value specified for the second element of this parameter.

**1-998000**

Specify the size of the trace table in kilobytes or megabytes.

### Element 2: Unit of measure

Specifies whether the value specified for the first element should be treated as number of kilobytes or number of megabytes.

**\*KB** The trace table size is specified in kilobytes.

**\*MB** The trace table size is specified in megabytes.

Top

---

## Trace data to be formatted (FMTDTA)

Specifies the number of bytes of traced data to be formatted.

**\*CALC**

The system determines the number of bytes of data to be formatted.

**72-99999**

Specify the number of bytes of data to be formatted.

Top



---

## Coded character set identifier (CCSID)

Specifies whether the extended binary-coded decimal interchange code (\*EBCDIC- 37) or the American National Standard Code for Information Interchange (\*ASCII- 819) character code or any other is used for the formatted output.

### \*EBCDIC

The EBCDIC (37) character code is used.

### \*ASCII

The ASCII (819) character code is used.

### 1-65533

Specify the coded character set identifier (CCSID) value to be used when formatting the trace data.

Top

---

## Job name (JOB)

Specifies the name or qualified name of the job that created the input QPCSMPT spooled file (SPLNBR parameter). This parameter is valid only if SET(\*FORMAT) is specified.

### Single values

\*        The job that issued this command is the job that created the input QPCSMPT spooled file.

### Qualifier 1: Job name

*name*    Specify the name of the job that created the input QPCSMPT spooled file.

### Qualifier 2: User

*name*    Specify the user name that identifies the user profile under which the job was run that created the input QPCSMPT spooled file.

### Qualifier 3: Number

#### 000000-999999

Specify the system-assigned job number of the job that created the input QPCSMPT spooled file.

Top

---

## Spooled file number (SPLNBR)

Specifies the file number of the QPCSMPT spooled file from the job (JOB parameter) that created the spooled file. This parameter is valid only if SET(\*FORMAT) is specified.

### \*LAST

The highest-numbered spooled file named QPCSMPT created by the specified job is used.

### \*ONLY

Only one spooled file named QPCSMPT was created by the specified job; therefore, the number of the spooled file is not necessary. If \*ONLY is specified and more than one spooled file for the specified job is named QPCSMPT, an error message is issued.

### 1-999999

Specify the number of the QPCSMPT spooled file created by the specified job.

---

## Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name and spooled file number parameter requirements have been met.

### \*ONLY

There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number and spooled file creation date and time.

### \*CURRENT

The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number and creation date and time is used.

**\*ANY** The job system name is not used to determine which spooled file is used. Use this value when the spooled file creation date and time parameter is to take precedence over the job system name when selecting a spooled file.

*name* Specify the name of the system where the job that created the spooled file ran.

Top

---

## Spooled file created (CRTDATE)

Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number and job system name parameter requirements have been met.

### Single values

### \*ONLY

There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number and job system name.

### \*LAST

The spooled file with the latest creation date and time of the specified job name, user name, job number, spooled file name, spooled file number and job system name is used.

### Element 1: Creation date

*date* Specify the date the spooled file was created.

### Element 2: Creation time

### \*ONLY

There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date.

### \*LAST

The spooled file with the latest creation time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file creation date is used.

*time* Specify the time the spooled file was created.

Top

---

## TCP/IP data (TCPDTA)

Specifies whether a subset of TCP/IP and/or SSL trace data should be collected. Each parameter element is optional; if no element value is specified, no filtering of trace data is done for that element. For example, if \*TCP is specified for element 1, only trace records where the TCP protocol is used are collected. If no value is specified for element 1, trace records using all TCP/IP protocols are collected.

### Element 1: Protocol

Specify a TCP/IP protocol to be traced.

**\*TCP** Enable trace for transmission control protocol.

**\*UDP** Enable trace for user datagram protocol.

**\*ICMP**  
Enable trace for internet control message protocol.

**\*IGMP**  
Enable trace for internet group management protocol

**\*ARP** Enable trace for address resolution protocol. This will only apply for TCP/IP.

**\*ICMP6**  
Enable trace for internet control message protocol version 6.

### Element 2: Local port

**1-65535**  
Specify one or two local port numbers for which trace data is collected.

### Element 3: Remote port

**1-65535**  
Specify one or two remote port numbers for which trace data is collected

### Element 4: Local IP address

*character-value*  
Specify a local internet protocol address.

### Element 5: Remote IP address

*character-value*  
Specify a remote internet protocol address.

### Element 6: Line description

*name* Specify the name of a line description for which TCP/IP trace data is to be collected.

Top

---

## Watch for message (WCHMSG)

Specifies up to five message identifiers which are to be watched for. If a value other than \*NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace exit program is called; if no trace exit program is defined, the trace stops.

### Single values

### \*NONE

No messages will be watched for.

#### **Element 1: Message identifier**

*name* Specify the 7-character message identifier to be watched for.

#### **Element 2: Comparison data**

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data includes the specified text, the watched for condition is true. If the message data does not contain the specified text, the trace function continues.

### \*NONE

No comparison data is specified. If a message matching the specified message ID is added to the specified message queue or log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the message data of the watched for message. If this text is found anywhere in the message data of a watched for message, the watch condition is considered to be true. This text is case sensitive. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

Top

---

## **Watched message queue (WCHMSGQ)**

Specifies where to watch for the message identifiers specified on the WCHMSG parameter. You can specify to watch the message being added to the system operator message queue, the history log, other message queues, and job logs. Up to three message queues or special values can be specified.

#### **Element 1: Message queue**

##### **Single values**

### \*SYSOPR

Watch messages added to the system operator message queue (QSYSOPR message queue in library QSYS).

### \*JOBLOG

Watch messages added to the job logs of the jobs specified for the **Watched job (WCHJOB)** parameter.

### \*HSTLOG

Watch messages added to the history log QHST.

#### **Qualifier 1: Message queue**

*name* Specify the name of the message queue to watch.

#### **Qualifier 2: Library**

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library where the message queue is located.

---

## Watched job (WCHJOB)

Specifies the job whose job log is watched for the messages specified on the WCHMSG parameter. The specified job will only be watched if \*JOBLOG is specified on the WCHMSGQ parameter. Up to five job names may be specified.

### Single values

\* Only the job log of the job that issued this trace command is watched.

### Element 1: Job name

#### Qualifier 1: Job name

##### *generic-name*

Specify the generic name of the job to be watched. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

*name* Specify the name of the job to be watched.

#### Qualifier 2: User

*name* Specify the user name of the job to be watched.

#### Qualifier 3: Number

\*ALL All jobs with the specified job name and user name are watched.

##### *000001-999999*

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier is specified.

---

## Watch for LIC log entry (WCHLICLOG)

Specifies up to five licensed internal code (LIC) log entry identifiers which are to be watched for. Each LIC log entry contains a major and a minor code. The watched for condition will be met if a LIC log entry is added that matches the specified major and minor codes and any comparison data specified. When the watched for log entry is added to the LIC log, the trace exit program is called, even when the comparison data specified does not match; if no trace exit program is defined, the trace stops.

### Single values

### \*NONE

No LIC log entries will be watched for.

### **Element 1: Major code**

**\*ALL** Any LIC log entry major code will be considered to be a match. If **\*ALL** is specified for the major code, you cannot specify **\*ALL** for the LIC log entry minor code.

#### *character-value*

Specify the LIC log major code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### **Element 2: Minor code**

**\*ALL** Any LIC log entry minor code will be considered to be a match. If **\*ALL** is specified for the minor code, you cannot specify **\*ALL** for the LIC log entry major code.

#### *character-value*

Specify the LIC log minor code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### **Element 3: Comparison data**

Specify comparison data to be used if a log entry matching the specified major and minor codes is added to the licensed internal code (LIC) log. If this text is found in the LIC log entry data fields of the watched for log entry, the watched for condition is true. If this text is not found in the LIC log entry data fields of the watched for log entry and no exit program is specified on the TRCPGM parameter, the trace function continues. If the log entry matches the specified major and minor codes and an exit program is specified on the TRCPGM parameter, but the entry data does not contain the specified text, the exit program is called to determine if the trace should continue or stop.

### \*NONE

No comparison data is specified. If a LIC log entry matching the specified major and minor codes is added to the LIC log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the entry data of the watched for log entry. If this text is found in the LIC log entry data fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server name, job name, user ID, job number, thread ID, exception ID, LIC module compile binary timestamp, LIC module offset, LIC module RU name, LIC module name, LIC module entry point name. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

When watching for an exception ID, all four hexadecimal digits of the exception ID must be specified. Also, the prefix MCH may be specified if you want to compare only against the exception ID field and avoid possible substring matches with the other fields.

Top

---

## **Length of time to watch (WCHTIMO)**

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

**1440** The time limit for watching for a particular message or LIC log entry is 1440 minutes (24 hours).

**\*NOMAX**

There is no time limit for watching for a particular message or LIC log entry.

**1-43200**

Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

---

## Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

There are three input parameters and one output parameter associated with the trace program. The four parameters are required:

1	Trace option setting	Input	Char(10)
2	Reserved	Input	Char(10)
3	Error detected	Output	Char(10)
4	Comparison data	Input	Char(*)

Allowed values for the "Trace option setting" parameter are:

**\*ON** The watch for trace facility is starting when the collection of trace information is started.

**\*MSGID**

A match on a message id specified on WCHMSG parameter occurred.

**\*LICLOG**

A match on a LIC log specified on the WCHLICLOG parameter occurred.

**\*CMPDATA**

The major and minor code of a LIC log matched, but the comparison data did not.

**\*INTVAL**

The time interval specified on TRCPGMITV parameter is elapsed.

**\*WCHTIMO**

The length of time to watch specified on WCHTIMO parameter is elapsed.

The "Reserved" parameter must be set to blanks.

Allowed values for the "Error detected" parameter are:

**\*CONTINUE**

The trace and the watch for trace event facility will continue running.

**\*STOP**

The trace and the watch for trace event facility will be ended.

**\*ERROR**

Error detected by customer trace program.

Allowed values for the "Comparison data" parameter when \*MSGID is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(7)	Message ID
11 B	CHAR(9)	Reserved
20 14	BINARY(4)	Offset to comparison data
24 18	BINARY(4)	Length of comparison data
* *	CHAR(*)	Message comparison data

Allowed values for the "Comparison data" parameter when \*LICLOG or \*CMPDATA is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(4)	LIC Log major code
8 8	CHAR(4)	LIC Log minor code
12 C	CHAR(8)	LIC Log identifier
20 14	BINARY(4)	Offset to comparison data
24 18	BINARY(4)	Length of comparison data
* *	CHAR(*)	LIC log comparison data

Allowed values for the "Comparison data" parameter when \*ON, \*INTVAL or \*WCHTIMO is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information (always 4).

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at <http://www.iseries.ibm.com/infocenter>.

## Single values

### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

### Qualifier 1: Trace program

*name* Specify the name of the trace exit program.

### Qualifier 2: Library

\*LIBL All libraries in the job's library list are searched until the first match is found.

*name* Specify the name of the library where the user exit program is located.

Top

---

## Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

### \*NONE

No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

**1-9999** Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.



---

## Examples

### Example 1: Starting SSL Traces

```
TRCCNN SET(*ON) TRCTYPE(*SSL)
```

This command starts tracing for Secure Sockets Layer (SSL) connections.

### Example 2: Starting IP Traces

```
TRCCNN SET(*ON) TRCTYPE(*IP)
```

This command starts tracing for connections at the internet protocol (IP) level.

### Example 3: Stopping Traces and Clearing Trace Storage

```
TRCCNN SET(*END)
```

This command stops all traces and deletes the trace table. No spooled output is generated.

### Example 4: Printing Traces

```
TRCCNN SET(*OFF)
```

This command stops all traces and generates a spooled file (QPCSMTRC) that contains the trace records collected by the TRCINT (Trace Internal) command, and a spooled file (QSYSPRT) that contains the formatted trace data.

### Example 5: Formatting Trace Data from TRCINT Command

```
TRCINT SET(*ON) TRCTYPE(*SCK)
TRCINT SET(*OFF)
TRCCNN SET(*FORMAT) TRCTYPE(*SSL) JOB(*) SPLNBR(*LAST)
```

The TRCINT (Trace Internal) commands are used to start collecting trace records related to all usage of sockets, and to stop collecting trace records and create a spooled file named QPCSMTRC. The TRCCNN command will use the trace information in the last spooled file named QPCSMTRC for the current job, and format the trace records related to SSL (Secure Sockets Layer) in a spooled file named QSYSPRT.

### Example 6: Specifying a Trace Table

```
TRCCNN SET(*ON) TRCTYPE(*IP) TRCTBL(USER)
```

This command starts tracing for connections at the internet protocol (IP) level and stores the trace data in the USER trace table.

### Example 7: Specifying a Trace Table Size in Megabytes

```
TRCCNN SET(*ON) TRCTYPE(*IP) SIZE(20000 *MB)
```

This command starts tracing for connections at the internet protocol (IP) level and stores the data in a 20000-megabyte trace table.

### Example 8: Specifying a CCSID for Trace Data

```
TRCCNN SET(*OFF) CCSID(*ASCII)
```

This command stops all traces and generates a spooled file (QSYSPRT). ASCII (819) CCSID will be used when formatting the trace data.

### Example 9: Start a Trace and Watch for a Message to End the Trace

```
TRCCNN SET(*ON) TRCTYPE(*IP) WCHMSG((MCH2804))
        WCHMSGQ((*SYSOPR) (*JOBLOG))
        WCHJOB((*ALL/MYUSER/MYJOBNAME))
        TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts tracing for connections at the internet protocol (IP) level. The trace will be ended when MCH2804 message is found on the System Operator message queue or within the \*ALL/MYUSER/MYJOBNAME job log. Also, MYLIB/TRCEXTPGM is specified as a trace exit program.

### Example 10: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
TRCCNN SET(*ON) TRCTYPE(*IP)
        WCHLICLOG(('99??' 9932 MYJOBNAME))
        WCHTIMO(*NOMAX)
```

This command starts tracing for connections at the internet protocol (IP) level. The trace will be ended when a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 is generated on the system. Also, the LIC log information should contain the text "MYJOBNAME". \*NOMAX on WCHTIMO parameter indicates that the trace will be active until the event occurs or ENDTRC command is issued manually.

[Top](#)

---

## Error messages

Unknown

[Top](#)

## Trace CPI Communications (TRCCPIC)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Trace Common Programming Interface Communications (TRCCPIC) command controls tracing of all CPI Communications that occur in the job in which the command is entered. The command sets a trace on or off, and traces (1) CPI Communications calls issued by a program and (2) data that is sent and received.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCCPIC command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

**Restrictions:** (1) The record format of the database output file must match the record format of the IBM-supplied output file, QACM0TRC. (2) The user must have specific authority from the security officer to use this command.

Top

### Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	*ON, *OFF, *END	Optional, Positional 1
MAXSTG	Maximum storage to use	1-16000, <u>200</u>	Optional, Positional 2
TRCFULL	Trace full	*WRAP, *STOPTRC	Optional, Positional 3
DTALEN	User data length	0-4096, <u>128</u>	Optional, Positional 4
OUTPUT	Output	*PRINT, *OUTFILE	Optional
OUTFILE	Output file	<i>Qualified object name</i>	Optional
	Qualifier 1: Output file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	*REPLACE, *ADD	

Top

---

## Trace option setting (SET)

Specifies whether a CPI Communications trace is started or ended.

The possible values are:

- \*ON** The trace is started. If the trace storage area becomes full, the action specified on the TRCFULL parameter is taken.
- \*OFF** The trace is ended. No other trace information is recorded, and the current information is written to the spooled output file or a database file.
- \*END** The trace ends. No other trace information is recorded and all current trace information is deleted. No output is generated.

Top

---

## Maximum storage to use (MAXSTG)

Specifies the maximum amount of storage (in kilobytes) used for the created trace records.

The possible values are:

**200** Up to 200KB of storage is used for trace records.

*number-of-kilobytes*

Specify the number of kilobytes of storage to use for trace records. Valid values range from 1 through 16000.

Top

---

## Trace full (TRCFULL)

Specifies the action taken when the maximum storage specified is full.

The possible values are:

**\*WRAP**

When the trace storage area is full, new trace information is written over the old information, starting at the beginning of the storage area.

**\*STOPTRC**

When the trace storage area is full, no new trace information is saved.

Top

---

## User data length (DTALEN)

Specifies the maximum length (in bytes) of user data that can be saved for each trace entry in the storage area. If the value specified is greater than the length of data received or sent across the communications line, only the actual data is traced. If the value specified is less than the data length received or sent, only the data length specified on this parameter is traced.

The possible values are:

**128** The maximum length of user data saved is 128 bytes.

*number-of-bytes*

Specify the maximum length of user data saved. Valid values range from 0 through 4096.

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

**\*PRINT**

The output is printed with the job's spooled output.

**\*OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

---

## Output file (OUTFILE)

Specifies the qualified name of the physical file to which the trace output is directed. If the file already exists, the system uses it. If the file does not exist, the system creates it. If the file is created, the text is "Output file for TRCCPIC." The possible library values are:

**\*LIBL** The library list is used to locate the database file.

**\*CURLIB**

The current library for the job is used to locate the database file. If no library is specified as the current library for the job, the QGPL library is used.

*library-name*

Specify the name of the library where the database file is located.

*file-name*

Specify the name of the physical database file to which the trace output is sent.

---

## Output member options (OUTMBR)

Specifies the name of the member in the physical file that receives the trace output. If the file is created by the system, a member is created with the name specified on this parameter. If the file exists but the member does not, a member with the specified name is created.

### Element 1: Member to Receive Output

**\*FIRST**

The first member of the specified file is used.

*member-name*

Specify the name of the member in the file that receives the trace output.

### Element 2: Operation to Perform on Member

### \*REPLACE

The new data replaces the existing data.

**\*ADD** The system adds the new records to the end of the existing records.

Top

---

## Examples

### Example 1: Starting Trace Operation

```
TRCCPIC  MAXSTG(350)  DTALEN(256)
```

This command traces the CPI Communications calls of the current job. The trace file contains 350KB of storage and wraps to the beginning if that amount of storage is filled with trace records. In addition, this command traces up to 256 bytes of user data on each input/output operation.

### Example 2: Stopping Trace Operation

```
TRCCPIC  SET(*OFF)  OUTPUT(*OUTFILE)  OUTFILE(TRACELIB/CPICTRACE)
          OUTMBR(TRACEMBR)
```

This command stops the trace and directs the output to the database file CPICTRACE in library TRACELIB. The output is directed to the member TRACEMBR.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C90

Maximum storage specified too small.

#### CPF2C94

Error occurred during OUTFILE processing. Trace stopped.

#### CPF3B30

No CPI-Communications calls were run. Trace ended.

#### CPF3B31

Job is already being serviced or traced.

#### CPF3B32

Trace already off.

#### CPF3B33

Unexpected Trace CPI Communications error occurred.

#### CPF3B34

Cannot deactivate trace, trace started from another job.

#### CPF3548

Serviced job completed running.

#### CPF3936

Job being serviced ended before trace started.

#### CPF9847

Error occurred while closing file &1 in library &2.

**CPF9848**

Cannot open file &1 in library &2 member &3.

**CPF9849**

Error while processing file &1 in library &2 member &3.

[Top](#)





---

## Trace ICF (TRCICF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Trace Intersystems Communications Functions (TRCICF) command is used to start and stop the tracing of language operations and Intersystem Communications Functions (ICF) issued by your program. TRCICF can be started from the command entry display or from a CL program.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCICF command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

### Restrictions:

1. The record format of the database output file must match the record format of the IBM-supplied output file, QAIIFTRCF.
2. The user must have specific authority from the security officer to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	* <u>ON</u> , *OFF, *END	Optional, Positional 1
MAXSTG	Maximum storage to use	1-16000, <u>200</u>	Optional, Positional 2
TRCFULL	Trace full	* <u>WRAP</u> , *STOPTRC	Optional, Positional 3
DTALEN	User data length	0-4096, <u>128</u>	Optional, Positional 4
OUTPUT	Output	* <u>PRINT</u> , *OUTFILE	Optional
OUTFILE	Output file	<i>Qualified object name</i>	Optional
	Qualifier 1: Output file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , * <u>FIRST</u>	
	Element 2: Replace or add records	* <u>REPLACE</u> , *ADD	

Top

---

## Trace option setting (SET)

Specifies whether an ICF trace is started, stopped, or ended.

- \*ON** Specifies that the trace ICF is started. If the trace storage area becomes full, the action specified on the TRCFULL parameter is taken.
- \*OFF** Specifies that the trace ICF is stopped. No further ICF activity is recorded and the trace records created are written to the job's spooled printer file or a database file.
- \*END** Specifies that the trace ICF is ended, and all existing trace information is deleted. No output is generated.

Top

---

## Maximum storage to use (MAXSTG)

Specifies the maximum amount of storage (in kilobytes) to be used for generated trace records. This parameter is valid only if \*ON is specified for the **Trace option setting (SET)** parameter.

**200** Specifies that a maximum of 200 kilobytes is used.

**1-16000**

Specify the number of kilobytes of storage to use for created records.

Top

---

## Trace full (TRCFULL)

Specifies whether new trace records replace old trace records with new records or to stop the trace function when all of the storage specified for the **Maximum storage to use (MAXSTG)** parameter has been used. This parameter is valid only if \*ON is specified for the **Trace option setting (SET)** parameter.

**\*WRAP**

When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are generated.

**\*STOPTRC**

Tracing is stopped when the trace table is full. You must still enter the TRCICF command and specify \*OFF to get the trace output.

Top

---

## User data length (DTALEN)

Specifies the maximum length (in bytes) of user data that is traced. This parameter is valid only if \*ON is specified for the **Trace option setting (SET)** parameter.

**128** A maximum of 128 bytes of user data is traced.

**0-4096** Specify the maximum number of bytes of user data to be traced.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

**\*PRINT**

The output is printed with the job's spooled output.

## \*OUTFILE

The output is saved in a user-specified database file.

Top

---

## Output file (OUTFILE)

Specifies the name and library of the physical file to which the trace ICF output is directed. If the database file specified already exists, its record format must match the record format of the IBM-supplied output file, QAIFTRCF.

### Qualifier 1: Output file

*name* Specify the name of the physical file to which the trace output is directed.

### Qualifier 2: Library

\*LIBL The library list is used to locate the file.

### \*CURLIB

The current library for the job is used to locate the file. If no library is specified as the current library for the job, QGPL is used.

*name* Specify the name of the library where the file is located.

Top

---

## Output member options (OUTMBR)

Specifies the name of the member in the physical file that receives the trace output. If the file specified for the **Output file** (OUTFILE) parameter is created by the system, a member is created for the file with the name specified. If the OUTFILE exists but the OUTMBR does not, a member with the specified name will be added. This parameter is valid only if SET(\*OFF) is specified.

### Element 1: Member to receive output

#### \*FIRST

The first member of the file specified for the OUTFILE parameter receives the trace output. If the file is created and \*FIRST is specified, the name of the created member is the same as that of the created file.

*name* Specify the name of the member, within the file specified for the OUTFILE parameter, that receives the trace output.

### Element 2: Replace or add records

#### \*REPLACE

The new data replaces existing data.

**\*ADD** The system adds the new records to the end of the existing records.

Top

---

## Examples

### Example 1: Starting Trace Operation

```
TRCICF  MAXSTG(350)  DTALEN(256)
```

This command traces the ICF input/output operations of the current job. The trace file contains 350K of storage and wraps to the beginning if that amount of storage is filled with trace records. In addition, this command traces up to 256 bytes of user data on each input/output operation.

### Example 2: Stopping Trace Operation

```
TRCICF SET(*OFF) OUTPUT(*OUTFILE) OUTFILE(TRACELIB/ICFTRACE)
      OUTMBR(TRACEMBR)
```

This command stops the trace and directs the output to the database file ICFTRACE in library TRACELIB. The output is directed to the member TRACEMBR.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C90

Maximum storage specified too small.

#### CPF2C93

No trace records logged.

#### CPF2C94

Error occurred during OUTFILE processing. Trace stopped.

#### CPF2C95

Trace already active.

#### CPF2C96

Trace already off.

#### CPF3B34

Cannot deactivate trace, trace started from another job.

#### CPF3205

File not created.

#### CPF3501

Job is already being serviced, traced, or debugged.

#### CPF3530

Conflicting entries in index QSERVICE.

#### CPF3548

Serviced job completed running.

#### CPF3925

Cannot open file &1.

#### CPF3936

Job being serviced ended before trace started.

#### CPF3950

Error message &2 received for file &1. Request ended.

#### CPF3951

File &1 cannot be overridden by file name &2.

#### CPF3969

Error during close of file &1. Output may not be complete.

**CPF5004**

Printer overflow line detected for file &1.

**CPF9847**

Error occurred while closing file &1 in library &2.

**CPF9848**

Cannot open file &1 in library &2 member &3.

**CPF9849**

Error while processing file &1 in library &2 member &3.

[Top](#)



---

## Trace Internal (TRCINT)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Trace Internal (TRCINT) command is the command interface to the Trace Licensed Internal Code service tool and is used for problem analysis. Specific types of traces are started and stopped by using this command. While previously started internal traces are being performed, additional internal traces can be started through this command. The output created by the trace is placed in a trace table. The records from the trace table can be written to a spooled printer file, to a database file, or to tape or optical media.

### Restrictions:

1. To use this command, you must have service (\*SERVICE) special authority, or be authorized to the Service Trace function of OS/400 through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_SERVICE\_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
2. The following user profiles have private authorities to use the command:
  - QSRV
3. When the WCHJOB parameter is specified, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (\*JOBCTL) special authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	*ON, *OFF, *END, *HOLD, *SAVE, *SIZE	Required, Positional 1
TRCTBL	Trace table name	<i>Character value</i> , *SYSDFT	Optional
SIZE	Trace table size	Single values: *NOCHG, *MAX, *MIN Other values: <i>Element list</i>	Optional
	Element 1: Number of units	1-998000	
	Element 2: Unit of measure	*KB, *MB	
TRCFULL	Trace full	*NOCHG, *WRAP, *STOPTRC	Optional, Positional 4

Keyword	Description	Choices	Notes
TRCTYPE	Trace type	Values (up to 50 repetitions): 000000-999999, *SVL, *MPL, *TNS, *TTPERF, *ACTCALL, *APPCOVRTCP, *APPCPS, *APPNALL, *APPNCPM, *APPNCPPS, *APPNDS, *APPNLM, *APPNMST, *APPNTRS, *AUTMGT, *AUXSTGALL, *AUXSTGMGT, *BSSMGT, *CCIAM, *CLUE, *CMNACCMTH, *CMNTRC, *CMTMGT, *COMMON, *CRPSRV, *CSTALL, *CSTCMN, *CTXMGT, *DBGINT, *DBMGT, *DLUR, *DSPPASTHR, *EREP, *ERRLOG, *EVTMGT, *EXCMGT, *FRCA, *HDWRSC, *HEAPMGT, *IDXMGT, *IFS, *IPCF, *ISDN, *JRNMG, *LNKTST, *LODDMP, *MCHOBS, *MITFMALL, *MITFMEVAL, *MITFMHEAP, *MITFMINT, *MITFMMI, *MITFMSTG, *MODMGT, *MOD2, *MSCP, *MSMCALL, *MSMDTL, *M36ALL, *M36ASC, *M36BSC, *M36CSP, *M36DKT, *M36DSK, *M36ILAN, *M36PRT, *M36SDL, *M36TAP, *M36TRN, *M36WS, *M36X25, *NTBTC, *OPC, *PASE, *PFRCOLSRV, *PGBMND, *PGMMGT, *PORTUTIL, *PRCMGT, *PRCTBL, *PSEUDOTERM, *PWRMGT, *QMGT, *QSMGT, *RCYMGT, *RMTSPT, *RSCMGT, *SCK, *SCKASCIO, *SCKNET, *SCKOTHER, *SCKOVRMPTN, *SCKOVRPEC, *SCKRSLV, *SCKSEL, *SCKSSL, *SCKSTDIO, *SIG, *SMBSVR, *SPCOBJMGT, *SRCSINK, *STGMGTALL, *STM, *SYNCGMT, *SYSCALL, *SYSJRNMG, *TCPIP, *TLI, *TRXMGT, *VRTDEVMGT, *VRTIO	Optional, Positional 2
JOB	Job name	Single values: <b>*NOCHG</b> Other values (up to 10 repetitions): <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Generic name, name, *ALL</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</i>	
	Qualifier 3: Number	000000-999999, *ALL	
SLTTHD	Thread ID to include	Single values: <b>*NOCHG</b> , *ALL, *SELECT Other values (up to 20 repetitions): <i>Hexadecimal value</i>	Optional
SVRTYPE	Server type	Single values: *ALL, <b>*NOCHG</b> Other values (up to 5 repetitions): <i>Generic name, name</i>	Optional
TASK	Task name	Single values: *ALL, <b>*NOCHG</b> Other values (up to 10 repetitions): <i>Generic name, name</i>	Optional
TASKNBR	Task number	Single values: *ALL, <b>*NOCHG</b> Other values (up to 10 repetitions): <i>Hexadecimal value</i>	Optional
SLTTRCPNT	Select trace points	Single values: <b>*NOCHG</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Trace point qualifier	0-65535	
	Element 2: To trace point qualifier	0-65535	
OMTTRCPNT	Omit trace points	Single values: <b>*NOCHG</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Trace point qualifier	0-65535	
	Element 2: To trace point qualifier	0-65535	



Keyword	Description	Choices	Notes
STOPTRCPNT	Stop on trace point	Single values: <b>*NOCHG</b> Other values (up to 4 repetitions): <i>Element list</i>	Optional
	Element 1: Trace point type	<i>Character value</i>	
	Element 2: Trace point qualifier	0-65535	
	Element 3: Trace point entry	1-65535	
	Element 4: Trace point entry offset	<i>Hexadecimal value</i>	
	Element 5: Trace point match value	<i>Character value</i>	
JOBTYPE	Job types	Single values: <b>*NONE</b> , <b>*ALL</b> Other values (up to 12 repetitions): <b>*DFT</b> , <b>*ASJ</b> , <b>*BCH</b> , <b>*EVK</b> , <b>*INT</b> , <b>*MRT</b> , <b>*RDR</b> , <b>*SBS</b> , <b>*SYS</b> , <b>*WTR</b> , <b>*PDJ</b> , <b>*PJ</b> , <b>*BCI</b>	Optional
JOBTRCIV	Job trace interval	0.1-9.9, <b>0.5</b>	Optional
TCPDTA	TCP/IP data	<i>Element list</i>	Optional
	Element 1: Protocol	<b>*TCP</b> , <b>*UDP</b> , <b>*ICMP</b> , <b>*IGMP</b> , <b>*ARP</b> , <b>*ICMP6</b>	
	Element 2: Local port	Values (up to 2 repetitions): 1-65535	
	Element 3: Remote port	Values (up to 2 repetitions): 1-65535	
	Element 4: Local IP address	<i>Character value</i>	
	Element 5: Remote IP address	<i>Character value</i>	
	Element 6: Line description	<i>Name</i>	
	Element 7: Line type	<b>*PPP</b> , <b>*OPC</b>	
SCKDTA	Sockets data	<i>Element list</i>	Optional
	Element 1: Address family	<b>*INET</b> , <b>*UNIX</b> , <b>*NS</b> , <b>*TELEPHONY</b> , <b>*NETBIOS</b>	
	Element 2: Socket type	<b>*STREAM</b> , <b>*DGRAM</b> , <b>*RAW</b> , <b>*SEQPACKET</b>	
	Element 3: Descriptor	Values (up to 2 repetitions): 0-65535	
	Element 4: Socket option	<b>*SODEBUG</b>	
DEV	Device	Single values: <b>*NONE</b> Other values (up to 16 repetitions): <i>Name</i>	Optional, Positional 3
CTL	Controller	Single values: <b>*NONE</b> Other values (up to 16 repetitions): <i>Element list</i>	Optional
	Element 1: Controller	<i>Name</i>	
	Element 2: Attached devices	<b>*NODEV</b> , <b>*ALLDEV</b>	
LIN	Line	Single values: <b>*NONE</b> Other values (up to 16 repetitions): <i>Element list</i>	Optional
	Element 1: Line	<i>Name</i>	
	Element 2: Attached controllers	<b>*NOCTL</b> , <b>*ALLCTL</b>	
NWI	Network interface	Single values: <b>*NONE</b> Other values (up to 16 repetitions): <i>Name</i>	Optional
NWS	Network server	Single values: <b>*NONE</b> Other values (up to 16 repetitions): <i>Name</i>	Optional
RSRCNAME	Resource name	Single values: <b>*NONE</b> Other values (up to 10 repetitions): <i>Name</i>	Optional
OUTDEV	Output device	<i>Name</i>	Optional
TASKINF	Task information	<b>*ALL</b> , <b>*TRCREF</b>	Optional
OUTPUT	Output	<b>*PRINT</b> , <b>*OUTFILE</b>	Optional

Keyword	Description	Choices	Notes
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	
WCHMSG	Watch for message	Single values: *NONE Other values (up to 5 repetitions): Element list	Optional
	Element 1: Message identifier	Name	
	Element 2: Comparison data	Character value, *NONE	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): Element list	Optional
	Element 1: Message queue	Single values: *SYSOPR, *JOBLOG, *HSTLOG Other values: Qualified object name	
	Qualifier 1: Message queue	Name	
	Qualifier 2: Library	Name, *LIBL	
WCHJOB	Watched job	Single values: * Other values (up to 5 repetitions): Element list	Optional
	Element 1: Job name	Qualified job name	
	Qualifier 1: Job name	Generic name, name	
	Qualifier 2: User	Name	
	Qualifier 3: Number	000001-999999, *ALL	
WCHLICLOG	Watch for LIC log entry	Single values: *NONE Other values (up to 5 repetitions): Element list	Optional
	Element 1: Major code	Character value, *ALL	
	Element 2: Minor code	Character value, *ALL	
	Element 3: Comparison data	Character value, *NONE	
WCHTIMO	Length of time to watch	1-43200, 1440, *NOMAX	Optional
TRCPGM	Trace program	Single values: *NONE Other values: Qualified object name	Optional
	Qualifier 1: Trace program	Name	
	Qualifier 2: Library	Name, *LIBL	
TRCPGMITV	Time interval	1-9999, *NONE	Optional

Top

## Trace option setting (SET)

Specifies whether internal tracing is started, stopped, ended, held, or saved. You can also specify whether the trace table size is changed.

This is a required parameter.

**\*ON** The collection of internal trace records is started for the trace types specified for the **Trace type (TRCTYPE)** parameter. If the trace table already contains trace records, the new trace records are added to the table. If the table is full, the action specified for the **Trace full (TRCFULL)** parameter is taken. If a trace table name other than \*SYSDFT is specified for the TRCTBL parameter and the table does not exist, it will be automatically created.

- \*OFF** Collection of internal trace records requested through previous Trace Internal (TRCINT) commands stops, and the records are written to the spooled printer file QPCSMPT or sent to a database file as indicated by **Output (OUTPUT)** parameter.
- \*END** Internal tracing ends and the internal trace records are deleted. No spooled output is generated. If a trace table name other than \*SYSDFT is specified for the TRCTBL parameter, it will be automatically deleted.
- \*HOLD**  
Internal traces are stopped, and the collected internal trace records are held in the trace table. Held records can be printed later if another Trace Internal (TRCINT) command is entered that specifies \*OFF for this prompt; or, the held records can be put on tape or optical media if \*SAVE is specified.
- \*SAVE**  
Internal traces are stopped, and the trace records are written to a tape or optical device specified by the OUTDEV parameter.
- \*SIZE** The size of the trace table is changed. The new size is specified on the **Trace table size (SIZE)** parameter.

Top

---

## Trace table name (TRCTBL)

Specifies the trace table to hold the collected trace data.

### \*SYSDFT

The system default trace table is used.

*name* Specify the name of the trace table to be used. If SET(\*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created.

Top

---

## Trace table size (SIZE)

Specifies the size of the trace table. This parameter can be specified only when \*SIZE is specified for the **Trace option setting (SET)** parameter or if SET(\*ON) is specified and tracing is not currently active for the trace table specified (TRCTBL parameter).

**Note:** The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage is not dynamically allocated as it is needed. This storage space will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

**Note:** If tracing data over a gigabit Ethernet line the trace table size should be greater than 128000 kilobytes or 128 megabytes.

### Single values

### \*NOCHG

The trace table size is not changed. If a new trace table is specified (TRCTBL parameter), a default size of 128 kilobytes will be used.

**\*MAX** The trace table is set to the maximum size of 258048 megabytes.

\***MIN** The trace table is set to the minimum size of 128 kilobytes.

### Element 1: Number of units

Specify the size of the trace table.

**1-998000**

Specify the size of the trace table in kilobytes or megabytes.

### Element 2: Unit of measure

Specifies whether the value specified for the first element should be treated as number of kilobytes or number of megabytes.

\***KB** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

\***MB** The trace table size is specified in megabytes. The valid range is 1 through 258048.

Top

---

## Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or tracing stops when the trace table is full. This parameter can be specified when \*ON is specified for the **Trace option setting (SET)** parameter.

\***NOCHG**

The trace table full action is not changed. If a new trace table is specified (TRCTBL parameter), the default action is for trace records to wrap when the trace table becomes full.

\***WRAP**

When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

\***STOPTRC**

Tracing is stopped when the trace table is full of trace records.

Top

---

## Trace type (TRCTYPE)

Specifies the type of traces to start. The two groups of trace types are:

- Component data trace codes cause active procedures to be traced within the system.
- General trace codes cause the iSeries instruction supervisor linkage, multiprogramming, transaction, or task and thread performance functions to be traced.

If trace code types are specified, \*ON must be specified for the **Trace option setting (SET)** parameter. If a value other than \*ON is specified for the SET parameter, TRCTYPE is ignored. Each trace type is identified by a special value or 6-digit code; all 6 digits must be specified. For a complete list of trace codes and special values, position the cursor on this parameter while prompting this command and press F4. Specify up to 50 types from the following code table:

TYPE OF TRACE	COMPONENT OR GENERAL TRACE CODE	SPECIAL VALUE
MI instruction supervisor linkage (SVL)	030000	*SVL
Multiprogramming level (MPL)	040000	*MPL
Transaction	080000	*TNS

Task/thread performance	090000	*TTPERF
Activation/Call	014400	*ACTCALL
MPTN - APPC over TCP/IP MPTN	014203	*APPCOVRTCP
APPC presentation services	014301	*APPCPS
APPN -- (all)	012506	*APPNALL
APPN control point management	012501	*APPNCPM
APPN control point presentation services	012504	*APPNCPPS
APPN directory services	012502	*APPNDS
APPN location management	012505	*APPNLM
APPN management services	012507	*APPNMST
transport		
APPN topology and routing services	012503	*APPNTRS
Authority management	010900	*AUTMGT
Auxiliary storage management -- detailedd	011104	*AUXSTGALL
Auxiliary storage management	011101	*AUXSTGMT
Byte string space management	012600	*BSSMGT
Common class input/output management (CCIOM)	011900	*CCIOM
Cluster engine	016402	*CLUE
Communications access method	015900	*CMNACCMTH
Commit management	011700	*CMTMGT
Communications trace service function	012300	*CMNTRC
Common functions	011200	*COMMON
Cryptographic services	013600	*CRPSRV
Cluster -- (all)	016400	*CSTALL
Cluster communications	016401	*CSTCMN
Context management	011000	*CTXMGT
Debugger interpreter	014500	*DBGINT
Database management (events for all database files are traced)	010400	*DBMGT
Dependent LU Requester (DLUR) Communication	015400	*DLUR
Display station pass-through	010804	*DSPPASTHR
Environmental recording, editing and printing (EREP)	012200	*EREP
Error log	012100	*ERRLOG
Event management	010600	*EVTMGT
Exception management	010200	*EXCMGT
Fast Response Cache Accelerator	016600	*FRCA
Hardware resources	014700	*HDWRSC
Heap management	013400	*HEAPMGT
Independent index management	011400	*IDXMGT
Integrated File System (IFS)	014800	*IFS
Inter-process communications facility (IPCF)	012000	*IPCF
Communications answer management (ISDN)	012700	*ISDN
Journal management	011600	*JRNMG
Link test service function	012400	*LNKTST
Load/dump (save/restore)	010801	*LODDMP
Machine observation	011300	*MCHOBS
Machine Interface (MI) Transformer	015100	*MITFMALL
MI Transformer - expression evaluation	015101	*MITFMEVAL
MI Transformer - heap operations	015102	*MITFMHEAP
MI Transformer - interpreter instructions	015103	*MITFMINT
MI Transformer - MI Instructions	015104	*MITFMMI
MI Transformer - storage management operations	015105	*MITFMSTG
Module management	013100	*MODMGT
Modula-2 run time support	012800	*MOD2

Machine services control point	010802	*MSCP
Main storage management -- calls	011102	*MSMCALL
Main storage management -- details	011103	*MSMDTL
NetBios on TCP/IP	015700	*NTBTCP
OptiConnect	015500	*OPC
Portable Application Solutions Environment	016100	*PASE
Performance collection services	016200	*PFRCOLSRV
Program binder	013200	*PGMBND
Program management	010300	*PGMMGT
Portability utilities	015200	*PORTUTIL
Process management	010500	*PRCMGT
Process table	015300	*PRCTBL
Pseudo terminal component	016500	*PSEUDOTERM
Power management	012900	*PWRMGT
Queue management	010700	*QMGMT
Queue space management	013300	*QSMGT
Recovery management	013500	*RCYMGT
Remote Support	016300	*RMTSPT
Resource management	010100	*RSCMGT
Sockets -- (all APIs)	014000	*SCK
Sockets Asynchronous and Overlapped Input/Output APIs	014002	*SCKASCIO
Sockets Network APIs	014004	*SCKNET
Other Socket APIs	014007	*SCKOTHER
MPTN AF_INET Sockets over SNA MPTN	014201	*SCKOVRMPTN
MPTN AF_INET Sockets over SNA P <sub>EC</sub>	014202	*SCKOVRPEC
Sockets Berkeley Resolver APIs	014005	*SCKRSLV
Sockets Select API	014003	*SCKSEL
Secure Sockets Layer (SSL) APIs	014006	*SCKSSL
Sockets Standard Input/Output APIs	014001	*SCKSTDIO
Signals	015600	*SIG
SMB server	015800	*SMBSVR
Space object management	011500	*SPCOBJMGT
Source/sink (device support) management	010803	*SRCSINK
Storage management -- (all)	011105	*STGMGTALL
Streams	013900	*STM
Synchronization management	013700	*SYNCMGT
MI system call	015200	*SYSCALL
System journal management	014100	*SYSJRMGT
Transmission Control Protocol/Internet Protocol (TCP/IP)	013800	*TCPIP
Transport Layer Interface	016000	*TLI
Transaction Management	016800	*TRXMGT
Virtual I/O	016700	*VRTIO
Virtual terminal management	013000	*VRTDEVMGT

**Note:** If \*TTPERF trace code is specified, it must be the first one in the TRCTYPE parameter list.

Top

---

## Job name (JOB)

Specifies the jobs from which trace records are to be collected. Only trace records which are generated in the specified job(s) are collected. A list of up to ten qualified job names can be specified. The trace records will be collected if they were generated from a job that matches any of the qualified job name values.

This parameter can be specified when \*ON is specified for the **Trace option setting (SET)** parameter. If a value other than \*ON is specified on the SET parameter, JOB is ignored.

### Single values

#### \*NOCHG

If any qualified job names had been specified for the JOB parameter on a previous TRCINT command for an active trace, the job name filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, \*NOCHG will behave the same as \*ALL/\*ALL/\*ALL for the JOB parameter.

### Qualifier 1: Job name

**\*ALL** All trace records generated by the defined trace are collected, regardless of what job name the trace record was generated from.

#### *generic-name*

Specify the generic name of the job from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job name that begin with the generic prefix.

*name* Specify the name of the job from which trace records are to be collected.

### Qualifier 2: User

**\*ALL** All trace records generated by the defined trace are collected, regardless of what job user name the trace record was generated from.

#### *generic-name*

Specify the generic user name of the job from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic user name specifies all jobs with user names that begin with the generic prefix.

*name* Specify the name of the user of the job from which trace records are to be collected.

### Qualifier 3: Number

**\*ALL** All trace records generated by the defined trace are collected, regardless of what job number the trace record was generated from. \*ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified.

#### **000000-999999**

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name or generic user name is specified.

Top

---

## Thread ID to include (SLTTHD)

Specifies the job threads from which trace records are to be collected. Only trace records which are generated in the specified thread(s) are collected. There can be only one job that has thread IDs associated with it. It must be the first qualified job name specified for the JOB parameter, and the job must be active.

### Single values

#### \*NOCHG

If any thread identifiers had been specified for the SLTTHD parameter on a previous TRCINT

command for an active trace, the thread ID filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, \*NOCHG will behave the same as \*ALL for this parameter.

**\*ALL** All trace records generated by the defined trace are collected, regardless of what thread ID the trace record was generated from.

**\*SELECT**

A list of thread identifiers is shown from which you can select up to twenty threads. Trace records from any of the selected thread identifiers are to be collected. \*SELECT is only valid if the TRCINT command is run in an interactive job.

**Other values**

*hexadecimal-value*

Specify the identifier of the thread from which trace records are to be collected. Up to twenty thread identifiers can be specified.

Top

---

## Server type (SVRTYPE)

Specifies the server type attribute for a job or task which is used to determine whether the trace record is collected. Only trace records which are generated in a job or task with the specified server type are collected. For a list of possible server types, see Work Management information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>

**Single values**

**\*NOCHG**

If a value was specified for the server type on a previous TRCINT command for an active trace, the value is not changed. If no value was specified on a previous TRCINT command, \*NOCHG will behave the same as \*ALL for this parameter.

**\*ALL** All trace records generated by the defined trace are collected, regardless of the server type attribute of the job or task the trace record was generated from.

**Other values**

*generic-name*

Specifies the generic server type for which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic server type specifies all jobs with a server type that begins with the generic prefix.

*name* Specify the server type for which trace records are to be collected. A list of up to five server types can be specified.

Top

---

## Task name (TASK)

Specifies the Licensed Internal Code (LIC) tasks from which trace records are to be collected. Only trace records which are generated from the specified LIC tasks are collected.

**Single values**

**\*NOCHG**

If a value was specified for the task name (TASK parameter) on a previous TRCINT command for



an active trace, the value is not changed. If no value was specified on a previous TRCINT command, \*NOCHG will behave the same as \*ALL for the task name.

**\*ALL** All trace records generated by the defined trace are collected, regardless of what LIC task the trace record was generated from.

#### Other values

##### *generic-name*

Specify the generic name of the LIC tasks for which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic task name specifies all tasks with task names that begin with the generic prefix.

*name* Specify the name of the LIC task for which trace records are to be collected. Up to ten LIC task names can be specified.

Top

---

## Task number (TASKNBR)

Specifies the Licensed Internal Code (LIC) task numbers from which trace records are to be collected. Only trace records which are generated in the specified LIC tasks are collected.

#### Single values

##### \*NOCHG

If a value was specified for the task number on a previous TRCINT command for an active trace, the value is not changed. If no value was specified on a previous TRCINT command, \*NOCHG will behave the same as \*ALL for this parameter.

**\*ALL** All trace records generated by the defined trace are collected, regardless of the task number of the LIC task the trace record was generated from.

#### Other values

##### *hexadecimal-value*

Specify a LIC task number for which trace records are to be collected. A list of up to ten task numbers can be specified.

Top

---

## Select trace points (SLTTRCPNT)

Specifies a list of up to five individual trace points or trace point ranges whose trace records are to be included. Trace records for trace points not specified on SLTTRCPNT will not be collected. If SLTTRCPNT is specified for a trace table that is currently active, the specified trace points will be added to the set of trace points for which trace records are being collected.

**Note:** This parameter and the OMTTRCPNT parameter are mutually exclusive. SLTTRCPNT cannot be specified for an active trace table that is using OMTTRCPNT to exclude specific trace points.

#### Single values

##### \*NOCHG

The list of trace points for which trace records are being collected does not change.

#### Element 1: Trace point qualifier

0-65535

Specify a single trace point qualifier or the start of a range of trace point qualifiers whose trace records are to be included. Up to five individual trace point qualifiers or trace point qualifier ranges may be specified.

#### Element 2: To trace point qualifier

0-65535

Specify the end of a range of trace point qualifiers whose trace records are to be included. A value should not be specified for this parameter element if only a single trace point is to be included.

Top

---

## Omit trace points (OMTTRCPNT)

Specifies a list of up to five individual trace points or trace point ranges whose trace records are to be excluded. Trace records for all trace points not specified on OMTTRCPNT will be collected. If OMTTRCPNT is specified for a trace table that is currently active, the specified trace points will be added to the set of trace points for which trace records are not being collected.

**Note:** This parameter and the SLTTRCPNT parameter are mutually exclusive. OMTTRCPNT cannot be specified for an active trace table that is using SLTTRCPNT to include only specific trace points.

### Single values

#### \*NOCHG

The list of trace points for which trace records are being excluded does not change.

#### Element 1: Trace point qualifier

0-65535

Specify a single trace point qualifier or the start of a range of trace point qualifiers whose trace records are to be excluded. Up to five individual trace point qualifiers or trace point qualifier ranges may be specified.

#### Element 2: To trace point qualifier

0-65535

Specify the end of a range of trace point qualifiers whose trace records are to be excluded. A value should not be specified for this parameter element if only a single trace point is to be excluded.

Top

---

## Stop trace points (STOPTRCPNT)

Specifies one or more trace points which, if they are encountered, will cause collection of trace records to stop. The trace table records are not deleted and can later be written to a spooled file or an output device by invoking TRCINT with SET(\*OFF) or SET(\*SAVE).

Up to four trace points may be specified. Tracing will be stopped if any of the specified trace points match a trace record being added to the specified trace table.

A specified trace point can have either two parts (trace point type and trace point qualifier) or five parts (trace point type, trace point qualifier, trace point entry number, trace point entry offset, and trace point match value). A two-part condition will stop trace data collection if any trace record is collected for the

specified trace point. A five-part condition will stop trace data collection only if the specified trace point match value exactly matches the data at the specified trace point entry offset.

### Single values

#### \*NOCHG

The list of stop trace points associated with the trace table does not change.

### Element 1: Trace point type

#### *character-value*

Specify the two-character trace point type.

### Element 2: Trace point qualifier

#### *0-65535*

Specify the trace point qualifier number.

### Element 3: Trace point entry

#### *1-65535*

Specify the trace point entry number.

### Element 4: Trace point entry offset

#### *hexadecimal-value*

Specify the offset (in hexadecimal) in the trace point entry.

### Element 5: Trace point match value

#### *character-value*

Specify the match value to be compared to the trace record data. The match value may be specified in character or hexadecimal. Character strings will be converted to the equivalent hexadecimal strings.

Top

---

## Job types (JOBTYPE)

Specifies the types of jobs for which trace data is to be collected for use in the batch job trace report. A maximum of 11 job types can be specified. This parameter can be specified only if TRCTYPE(\*MPL) or TRCTYPE(040000) is specified.

**Note:** The value \*DFT includes the values \*ASJ, \*BCH, \*EVK, \*MRT, \*PDJ, \*PJ and \*BCI. The value \*BCH includes the values \*EVK, \*MRT, \*PDJ, \*PJ, and \*BCI.

### Single values

#### \*NONE

No jobs are traced.

\*ALL All of the job types are traced.

### Other values

\*DFT Batch and autostart jobs are traced.

\*ASJ Autostart jobs are traced.

\*BCH Batch jobs are traced.

- \*EVK Jobs started by a procedure start request are traced.
- \*INT Interactive jobs are traced.
- \*MRT Multiple requester terminal jobs are traced.
- \*RDR Reader jobs are traced.
- \*SBS Subsystem monitor jobs are traced.
- \*SYS System jobs are traced.
- \*WRT Writer jobs are traced.
- \*PDJ Print driver jobs are traced.
- \*PJ Prestart jobs are traced.
- \*BCI Batch Immediate jobs are traced.

Top

## Job trace interval (JOBTRCITV)

Specifies the time (in CPU seconds) between each collection of the job trace data. This parameter can be specified only if TRCTYPE(\*MPL) or TRCTYPE(040000) is specified.

**0.5** A time slice interval value of 0.5 CPU seconds is used.

**0.1-9.9** Specify the number of CPU seconds to be used as the trace interval value.

Top

## TCP/IP data (TCPDTA)

Specifies whether a subset of TCP/IP and/or Sockets trace data should be collected. This parameter can be specified only if TRCTYPE(\*TCPIP) or TRCTYPE(013800) is specified, or if one or more of the socket trace types is specified (\*SCK, \*SCKSTDIO, \*SCKASCIO, \*SCKSEL, \*SCKNET, \*SCKRSLV, \*SCKSSL, \*SCKOTHER or 014000, 014001, 014002, 014003, 014004, 014005, 014006, 014007). Each parameter element is optional; if no element value is specified, no filtering of TCP/IP and/or Sockets trace data is done for that element. For example, if \*UDP is specified for element 1, only trace records where the UDP protocol is used are collected. If no value is specified for element 1, trace records using all TCP/IP protocols are collected.

If no values are specified for any element of TCPDTA and tracing of TCP/IP or Sockets data was not already active, no filtering of TCP/IP or Sockets trace data is done. If tracing of TCP/IP or Sockets data was already active and no TCPDTA values are specified, previous data filtering values will remain in effect.

### Element 1: Protocol

Specify a TCP/IP protocol to be traced.

- \*TCP Enable trace for transmission control protocol.
- \*UDP Enable trace for user datagram protocol.
- \*ICMP  
Enable trace for internet control message protocol.
- \*IGMP  
Enable trace for internet group management protocol.

**\*ARP** Enable trace for address resolution protocol. This will only apply for TCP/IP.

**\*ICMP6**

Enable trace for internet control message protocol version 6.

**Element 2: Local port**

Specify one or two local port numbers for which trace data is collected.

**Element 3: Remote port**

Specify one or two remote port numbers for which trace data is collected.

**Element 4: Local IP address**

Specify a local internet protocol address.

**Element 5: Remote IP address**

Specify a remote internet protocol address.

**Element 6: Line description**

Specify the name of a line description for which TCP/IP trace data is to be collected.

**Element 7: Line type**

Specify whether the collection of trace information should be restricted to the specified line type.

**\*PPP** The collection of trace information is restricted to Point-to-point lines.

**\*OPC** The collection of trace information is restricted to Opticonnect.

Top

---

## Sockets data (SCKDTA)

Specifies whether a subset of Sockets trace data should be collected. This parameter can be specified only if one or more of the socket trace types is specified (\*SCK, \*SCKSTDIO, \*SCKASCIO, \*SCKSEL, \*SCKNET, \*SCKRSLV, \*SCKSSL, \*SCKOTHER or 014000, 014001, 014002, 014003, 014004, 014005, 014006, 014007).

Each parameter element is optional; if no element value is specified, no filtering of Sockets trace data is done for that element. For example, if \*INET is specified for element 1, only trace records where the AF\_INET address family is used are collected. If no value is specified for element 1, trace records using all socket address families are collected.

If no values are specified for any element of SCKDTA and tracing of Sockets data was not already active, no filtering of Sockets trace data is done. If tracing of Sockets data was already active and no SCKDTA values are specified, previous data filtering values will remain in effect.

The subset values specified on the SCKDTA parameter are used in combination with any subset values specified on the TCPDTA parameter to generate the complete subsetting criteria.

**Element 1: Address family**

Specify a sockets address family for which trace data is collected.

**\*INET** Enable trace for AF\_INET address family.

**\*UNIX**

Enable trace for AF\_UNIX and AF\_UNIX\_CCSSID address families.

**\*NS** Enable trace for AF\_NS address family.

**\*TELEPHONY**

Enable trace for AF\_TELEPHONY address family.

**\*NETBIOS**

Enable trace for AF\_NETBIOS address family.

**Element 2: Socket type**

Specify a socket type for which trace data is collected.

**\*STREAM**

Enable trace for SOCK\_STREAM (full-duplex stream) socket type.

**\*DGRAM**

Enable trace for SOCK\_DGRAM (datagram) socket type.

**\*RAW** Enable trace for SOCK\_RAW (direct to network protocol) socket type.

**\*SEQPACKET**

Enable trace for SOCK\_SEQPACKET (full-duplex sequenced packet) socket type.

**Element 3: Descriptor**

Specify one or two socket descriptor numbers for which trace data is collected.

**Element 4: Socket option**

Specify a socket option for which trace data is collected.

**\*SODEBUG**

Applications with the SO\_DEBUG Socket Option set on will have trace data collected.

Top

---

## Device (DEV)

Specifies the names of the devices for which the associated internal events are traced. This parameter can be specified only if 010803 or \*SRCSINK is specified on the **Trace type (TRCTYPE)** parameter. Up to 16 device names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the DEV parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(\*ON) multiple times for the same trace table
- specifying \*ALLDEV on the CTL parameter
- specifying \*ALLCTL on the LIN parameter

**\*NONE**

No devices are traced by this command.

*name* Specify the name of the device for which the internal trace is started. The device name must be the same as the name specified in the associated device description.

Top

---

## Controller (CTL)

Specifies the names of the controllers for which the associated internal events are to be traced. This parameter can be specified only if 010803 or \*SRCSINK is specified on the **Trace type (TRCTYPE)** parameter. Up to 16 controller names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the CTL parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(\*ON) multiple times for the same trace table
- specifying \*ALLDEV on the CTL parameter
- specifying \*ALLCTL on the LIN parameter

### Single values

#### \*NONE

No controllers are traced by this command.

### Element 1: Controller

*name* Specify the name of the controller for which the internal trace is started. The controller names must be the same as the names specified in the associated controller description.

### Element 2: Attached devices

Specifies if the devices on a controller are traced.

#### \*NODEV

No attached devices for the specified controller are traced.

#### \*ALLDEV

All attached devices for the specified controller are traced. The attached devices do not count toward the maximum of 16 source/sink objects that can be named on the DEV,CTL,LIN,NWI and NWS parameters. However, the attached devices do count toward the maximum of 256 source/sink objects that can be traced in a single trace table.

Top

---

## Line (LIN)

Specifies the names of the lines for which the associated internal events are to be traced. This parameter can be specified only if 010803 is specified on the **Trace type (TRCTYPE)** parameter. Up to 16 line names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the LIN parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(\*ON) multiple times for the same trace table
- specifying \*ALLDEV on the CTL parameter
- specifying \*ALLCTL on the LIN parameter

### Single values

#### \*NONE

No lines are traced by this command.

### Element 1: Line

*name* Specify the name of the line for which the internal trace is started. The line name must be the same as the name specified in the associated line description.

### Element 2: Attached controllers

Specifies if the controllers on a line are traced.

#### \*NOCTL

No attached controllers for the specified line are traced.

#### \*ALLCTL

All attached controllers for the specified line are traced. The attached controllers do not count toward the maximum of 16 source/sink objects that can be named on the DEV,CTL,LIN,NWI and NWS parameters. However, the attached controllers do count toward the maximum of 256 source/sink objects that can be traced in a single trace table.

Top

---

## Network interface (NWI)

Specifies the names of the network interfaces for which the associated internal events are to be traced. This parameter can be specified only if 010803 or \*SRCSINK is specified for the **Trace type (TRCTYPE)** parameter. Up to 16 network interface names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the LIN parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(\*ON) multiple times for the same trace table
- specifying \*ALLDEV on the CTL parameter
- specifying \*ALLCTL on the LIN parameter

### Single values

#### \*NONE

No network interfaces are traced by this command.

### Other values



*name* Specify the name of the network interface for which the internal trace is started. The network interface name must be the same as the name specified in the associated network interface description.

Top

---

## Network Server (NWS)

Specifies the names of the network servers for which the associated internal events are traced. This parameter can be specified only if 010803 or \*SRCSINK is specified on the **Trace type (TRCTYPE)** parameter. Up to 16 network server names may be specified.

The total number of source/sink objects that can be named on the device(DEV), controller(CTL), line(LIN), network interface(NWI) and network server(NWS) parameters is 16. For example, if you enter 16 values for the LIN parameter, you cannot enter values for the other parameters.

The maximum number of source/sink objects that can be traced in a single trace table is 256. Even if you stay within the limit of 16 named source/sink objects on one TRCINT command, you may exceed the 256 source/sink object limit. Examples of ways to exceed the limit are:

- specifying SET(\*ON) multiple times for the same trace table
- specifying \*ALLDEV on the CTL parameter
- specifying \*ALLCTL on the LIN parameter

### Single values

#### \*NONE

No network servers are traced by this command.

### Other values

*name* Specify the name of the network server for which the internal trace is started. The network server name must be the same as the name specified in the associated network server description.

Top

---

## Hardware Resource (RSRCNAME)

Specifies the names of the hardware resources for which the associated internal events are traced. This parameter can be specified only if 014700 or 011900 is specified on the **Trace type (TRCTYPE)** parameter.

**Note:** A combined total of 10 hardware resources (RSRCNAME) are allowed.

### Single values

#### \*NONE

No hardware resources are traced by this command.

### Other values

*name* Specify the names of up to 10 hardware resources for which the internal trace is started.

Top

---

## Device (OUTDEV)

Specifies the tape device or optical device on which the held trace records are written. This parameter must be specified if \*SAVE is specified for the **Trace option setting (SET)** parameter.

*name* Specify the name of the device description of the tape or optical device.

Top

---

## Task information (TASKINF)

Specifies whether information for all licensed internal code (LIC) tasks is written to a spooled file or output device. This parameter can be specified if SET(\*OFF) or SET(\*SAVE) is specified.

### \*TRCREF

Write information only for LIC tasks that were referenced by trace records in the specified trace table.

\***ALL** Write information for all LIC tasks that were in existence while the trace was active.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is printed with the job's spooled output or sent to a database file.

### \*PRINT

The output is printed with the job's spooled output.

### \*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Top

---

## File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

### Qualifier 1: File to receive output

*name* Specify the name of the database file to which the command output is directed.

### Qualifier 2: Library

\*LIBL The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

### \*CURLIB

The current library for the thread is used to locate the file. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the name of the library to be searched.

**Note:** If a new file is created, system file QASCTRCI in system library QSYS is used as a model.

---

## Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when \*OUTFILE is specified for the **Output (OUTPUT)** parameter.

### Element 1: Member to receive output

#### \*FIRST

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

*name* Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

### Element 2: Replace or add records

#### \*REPLACE

The existing records in the specified database file member are replaced by the new records.

**\*ADD** The new records are added to the existing information in the specified database file member.

---

## Watch for message (WCHMSG)

Specifies up to five message identifiers which are to be watched for. If a value other than \*NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace exit program is called; if no trace exit program is defined, the trace stops.

### Single values

#### \*NONE

No messages will be watched for.

### Element 1: Message identifier

*name* Specify the 7-character message identifier to be watched for.

### Element 2: Comparison data

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data includes the specified text, the watched for condition is true. If the message data does not contain the specified text, the trace function continues.

#### \*NONE

No comparison data is specified. If a message matching the specified message ID is added to the specified message queue or log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the message data of the watched for message. If this text is found anywhere in the message data of a watched for message, the watch condition is

considered to be true. This text is case sensitive. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

Top

---

## Watched message queue (WCHMSGQ)

Specifies where to watch for the message identifiers specified on the WCHMSG parameter. You can specify to watch the message being added to the system operator message queue, the history log, other message queues, and job logs. Up to three message queues or special values can be specified.

### Element 1: Message queue

#### Single values

##### \*SYSOPR

Watch messages added to the system operator message queue (QSYSOPR message queue in library QSYS).

##### \*JOBLOG

Watch messages added to the job logs of the jobs specified for the **Watched job (WCHJOB)** parameter.

##### \*HSTLOG

Watch messages added to the history log QHST.

#### Qualifier 1: Message queue

*name* Specify the name of the message queue to watch.

#### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library where the message queue is located.

Top

---

## Watched job (WCHJOB)

Specifies the job whose job log is watched for the messages specified on the WCHMSG parameter. The specified job will only be watched if \*JOBLOG is specified on the WCHMSGQ parameter. Up to five job names may be specified.

#### Single values

\* Only the job log of the job that issued this trace command is watched.

### Element 1: Job name

#### Qualifier 1: Job name

*generic-name*

Specify the generic name of the job to be watched. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

*name* Specify the name of the job to be watched.

**Qualifier 2: User**

*name* Specify the user name of the job to be watched.

**Qualifier 3: Number**

\*ALL All jobs with the specified job name and user name are watched.

000001-999999

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier is specified.

Top

---

## Watch for LIC log entry (WCHLICLOG)

Specifies up to five licensed internal code (LIC) log entry identifiers which are to be watched for. Each LIC log entry contains a major and a minor code. The watched for condition will be met if a LIC log entry is added that matches the specified major and minor codes and any comparison data specified. When the watched for log entry is added to the LIC log, the trace exit program is called, even when the comparison data specified does not match; if no trace exit program is defined, the trace stops.

**Single values**

\*NONE

No LIC log entries will be watched for.

**Element 1: Major code**

**\*ALL** Any LIC log entry major code will be considered to be a match. If \*ALL is specified for the major code, you cannot specify \*ALL for the LIC log entry minor code.

*character-value*

Specify the LIC log major code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

**Element 2: Minor code**

**\*ALL** Any LIC log entry minor code will be considered to be a match. If \*ALL is specified for the minor code, you cannot specify \*ALL for the LIC log entry major code.

*character-value*

Specify the LIC log minor code to be watched for. You can specify either a hexadecimal digit or a

question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### Element 3: Comparison data

Specify comparison data to be used if a log entry matching the specified major and minor codes is added to the licensed internal code (LIC) log. If this text is found in the LIC log entry data fields of the watched for log entry, the watched for condition is true. If this text is not found in the LIC log entry data fields of the watched for log entry and no exit program is specified on the TRCPGM parameter, the trace function continues. If the log entry matches the specified major and minor codes and an exit program is specified on the TRCPGM parameter, but the entry data does not contain the specified text, the exit program is called to determine if the trace should continue or stop.

#### \*NONE

No comparison data is specified. If a LIC log entry matching the specified major and minor codes is added to the LIC log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the entry data of the watched for log entry. If this text is found in the LIC log entry data fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server name, job name, user ID, job number, thread ID, exception ID, LIC module compile binary timestamp, LIC module offset, LIC module RU name, LIC module name, LIC module entry point name. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

When watching for an exception ID, all four hexadecimal digits of the exception ID must be specified. Also, the prefix MCH may be specified if you want to compare only against the exception ID field and avoid possible substring matches with the other fields.

Top

---

## Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

**1440** The time limit for watching for a particular message or LIC log entry is 1440 minutes (24 hours).

#### **\*NOMAX**

There is no time limit for watching for a particular message or LIC log entry.

#### **1-43200**

Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

---

## Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.

- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

There are three input parameters and one output parameter associated with the trace program. The four parameters are required:

1	Trace option setting	Input	Char(10)
2	Reserved	Input	Char(10)
3	Error detected	Output	Char(10)
4	Comparison data	Input	Char(*)

Allowed values for the "Trace option setting" parameter are:

**\*ON** The watch for trace facility is starting when the collection of trace information is started.

**\*MSGID**

A match on a message id specified on WCHMSG parameter occurred.

**\*LICLOG**

A match on a LIC log specified on the WCHLICLOG parameter occurred.

**\*CMPDATA**

The major and minor code of a LIC log matched, but the comparison data did not.

**\*INTVAL**

The time interval specified on TRCPGMITV parameter is elapsed.

**\*WCHTIMO**

The length of time to watch specified on WCHTIMO parameter is elapsed.

The "Reserved" parameter must be set to blanks.

Allowed values for the "Error detected" parameter are:

**\*CONTINUE**

The trace and the watch for trace event facility will continue running.

**\*STOP**

The trace and the watch for trace event facility will be ended.

**\*ERROR**

Error detected by customer trace program.

Allowed values for the "Comparison data" parameter when \*MSGID is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(7)	Message ID
11 B	CHAR(9)	Reserved
20 14	BINARY(4)	Offset to comparison data
24 18	BINARY(4)	Length of comparison data
* *	CHAR(*)	Message comparison data

Allowed values for the "Comparison data" parameter when \*LICLOG or \*CMPDATA is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information
4 4	CHAR(4)	LIC Log major code
8 8	CHAR(4)	LIC Log minor code

12	C	CHAR(8)	LIC Log identifier
20	14	BINARY(4)	Offset to comparison data
24	18	BINARY(4)	Length of comparison data
*	*	CHAR(*)	LIC log comparison data

Allowed values for the "Comparison data" parameter when \*ON, \*INTVAL or \*WCHTIMO is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of trace information (always 4).

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at <http://www.iseries.ibm.com/infocenter>.

### Single values

#### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

#### Qualifier 1: Trace program

*name* Specify the name of the trace exit program.

#### Qualifier 2: Library

\*LIBL All libraries in the job's library list are searched until the first match is found.

*name* Specify the name of the library where the user exit program is located.

Top

## Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

#### \*NONE

No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

**1-9999** Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

## Examples

### Example 1: Starting Component Data Traces and Call Traces

```
TRCINT SET(*ON) TRCTYPE(010100 010400 050500 051200)
```

This command starts component data traces and call traces for resource management and database. Database operations associated with database files are used to collect component data trace records.

### Example 2: Tracing Lines and Controllers

```
TRCINT SET(*ON) TRCTYPE(*SRCSINK) TRCTBL(*SYSDFT)
      DEV(WS1 WS2 WS3) CTL((C1) (C2)) LIN((L1) (L2))
```



This command starts component data traces for source/sink management (device support) operations involving the devices WS1, WS2, and WS3, lines L1 and L2, and controllers C1 and C2.

### Example 3: Stopping Traces and Clearing Trace Table

```
TRCINT SET(*END) TRCTBL(*SYSDFT)
```

This command stops all traces and deletes the trace records from the system default trace table.

### Example 4: Tracing Communications Trace Service Function

```
TRCINT SET(*ON) TRCTYPE(*CMNTRC)
```

This command starts component data traces for the communications trace service function.

### Example 5: Using Job Filtering Capability

```
TRCINT SET(*ON) TRCTBL(MYFTPTRACE)
        TRCTYPE(*TCP/IP) JOB(QTCP/QTFTP*)
```

This command starts a TCP/IP trace and will only collect trace records for trace points collected in jobs with user name QTCP and job names that begin with the prefix QTFTP. Trace records will be stored in trace table MYFTPTRACE.

### Example 6: Start a Trace and Watch for a Message to End the Trace

```
TRCINT SET(*ON) TRCTYPE(*CMNTRC) WCHMSG((MCH2804))
        WCHMSGQ((*SYSOPR) (*JOBLOG))
        WCHJOB((*ALL/MYUSER/MYJOBNAME))
        TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts component data traces for the communications trace service function. The trace will be ended when MCH2804 message is found on the System Operator message queue or within the \*ALL/MYUSER/MYJOBNAME job log. Also, MYLIB/TRCEXTPGM is specified as a trace exit program.

### Example 7: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
TRCINT SET(*ON) TRCTYPE(*CMNTRC)
        WCHLICLOG(('99??' 9932 MYJOBNAME))
        WCHTIMO(*NOMAX)
```

This command starts component data traces for the communications trace service function. The trace will be ended when a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 is generated on the system. Also, the LIC log information should contain the text "MYJOBNAME". \*NOMAX on WCHTIMO parameter indicates that the trace will be active until the event occurs or ENDTRC command is issued manually.

### Example 8: Sending the Trace Output to an Outfile

```
TRCINT SET(*OFF) OUTPUT(*OUTFILE)
        OUTFILE(MYOUTFILE) OUTMBR(MYOUTMBR)
```

This command generates the outfile MYOUTFILE with member MYOUTMBR with the data traces.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPD3683

TRCFULL parameter only valid with SET(\*ON).

- CPD3684**  
Specified parameters only valid when SET(\*ON) is specified.
- CPD3685**  
SLTTRCPNT or OMTTRCPNT parameters are mutually exclusive.
- CPD3686**  
TCPDTA only valid if TRCTYPE(\*TCPIP) or one or more of the socket TRCTYPE values is specified.
- CPD3687**  
SCKDTA only valid if one or more of the socket TRCTYPE values is specified.
- CPD3688**  
Job types (JOBTYPE) parameter only valid with TRCTYPE(\*MPL).
- CPD3689**  
Job trace interval (JOBTRCITV) parameter only valid with TRCTYPE(\*MPL)
- CPD368A**  
Cannot change trace point selection criteria for active trace table.
- CPD36C0**  
OUTDEV parameter only valid with SET(\*SAVE).
- CPD36C1**  
SIZE parameter only valid with SET(\*ON) or SET(\*SIZE).
- CPD36CD**  
TASKINF parameter only valid with SET(\*OFF) or SET(\*SAVE).
- CPD3983**  
Range of parameter SIZE not valid.
- CPD3990**  
User number qualifier not valid.
- CPD3991**  
Job, thread identifier or task not active.
- CPF3515**  
Too many trace requests or objects.
- CPF3516**  
Trace table is full.
- CPF3517**  
Cannot specify \*SELECT for the thread ID to include.
- CPF3518**  
End time and date earlier than start time and date.
- CPF3659**  
Total of specified CTL, DEV, LIN, NWI, and NWS greater than allowed.
- CPF3679**  
Service function returned completion code &1 qualifier &2.
- CPF3683**  
Error occurred trying to open printer file.
- CPF3684**  
Error occurred while trying to close a print file.
- CPF3685**  
Error occurred while data being put to print file.

**CPF3686**  
Service function ended with error message.

**CPF3687**  
Error occurred while trying to open file.

**CPF3688**  
Error occurred while tape or optical file being closed.

**CPF3689**  
Error occurred while writing data to tape or optical device.

**CPF368A**  
Trace table size not changed.

**CPF3692**  
Error occurred while trying to write data to tape or optical device.

**CPF3693**  
Service function ended because error occurred.

**CPF3694**  
Cannot start service function.

**CPF3695**  
No trace tables exist.

**CPF3696**  
No traces recorded.

**CPF3697**  
Trace type parameter value missing.

**CPF7A11**  
Trace table &1 not found.

**CPF7A13**  
Trace table cannot be created.

**CPF7A15**  
Trace buffer must be cleared.

**CPF7A17**  
Trace already is active.

**CPF7A1A**  
Cannot change trace point selection criteria for active trace table.

**CPF7A1C**  
IP address not valid.

**CPF98A2**  
Not authorized to &1 command.

Top



---

## Trace Job (TRCJOB)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Trace Job (TRCJOB) command controls traces of original program model (OPM) programs and Integrated Language Environment (ILE) procedure calls and returns that occur in the current job or in the job being serviced as a result of the Start Service Job (STRSRVJOB) command directed to that job. The command, which sets a trace on or off, can trace module flow, operating system data acquisition (including CL command traces), or both.

As the trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be written to a spooled printer file, QPSRVTRC. The trace records can also be directed to a database output file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCJOB command, the job that is traced is the one identified by the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

### Restrictions:

1. The record format of the database output file must match the record format of the IBM-supplied output file QATRCJOB.
2. The number of trace records processed between the start and end of the trace must not exceed one million.
3. The Transfer Job (TFRJOB) command must not be issued while TRCJOB command is active.
4. The following user profiles have private authorities to use the command:
  - QPGMR
  - QSRV
  - QSRVBAS
  - QSYSOPR
  - QRJE

Top

---

## Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	<u>*ON</u> , *OFF, *END	Optional, Positional 1
TRCTYPE	Trace type	<u>*ALL</u> , *FLOW, *DATA	Optional, Positional 2
MAXSTG	Maximum storage to use	1-16000, <u>4096</u>	Optional, Positional 3
TRCFULL	Trace full	<u>*WRAP</u> , *STOPTRC	Optional, Positional 4

Keyword	Description	Choices	Notes
EXITPGM	Program to call before trace	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Program to call before trace	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , <b>*CURLIB</b>	
SLTPRC	Select procedures to trace	Single values: <b>*ALL</b> , <b>*NONE</b> Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Program	<i>Qualified object name</i>	
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , <b>*CURLIB</b>	
	Element 2: Type	<b>*PGM</b> , <b>*SRVPGM</b>	
SLTTHD	Thread ID to include	Single values: <b>*ALL</b> , <b>*SELECT</b> Other values (up to 20 repetitions): <i>Hexadecimal value</i>	Optional
OUTPUT	Output	<b>*PRINT</b> , <b>*OUTFILE</b>	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b> , <b>*CURLIB</b>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <b>*FIRST</b>	
	Element 2: Replace or add records	<b>*REPLACE</b> , <b>*ADD</b>	

Top

---

## Trace option setting (SET)

Specifies whether the collection of trace records starts or stops.

**\*ON** The collection of trace records is started.

**\*OFF** The collection of trace records is stopped, and the trace records are written to the spooled printer file or output file.

**\*END** The collection of trace records is stopped, and all existing trace records are deleted. No spooled printer file is created.

Top

---

## Trace type (TRCTYPE)

Specifies the type of trace data to store in a trace file.

**\*ALL** All the trace data collected is stored in trace records. This includes tracing the flow of control and the trace data itself.

**\*FLOW**

The flow of control is traced when OPM programs and ILE procedures are called and when they return control.

**\*DATA**

The data is provided at predefined trace points within the operating system stored in trace records. This includes trace records for the CL commands that have run.

---

## Maximum storage to use (MAXSTG)

Specifies the maximum amount of storage used for collected trace records.

**4096** A maximum of 4096 kilobytes of storage is used.

**1-16000**

Specify the maximum amount of storage, in kilobytes, used to store trace records. (One kilobyte equals 1024 bytes.)

---

## Trace full (TRCFULL)

Specifies whether the trace records are to wrap (replace oldest records with new records) or to set trace off when all of the storage has been used.

**\*WRAP**

When the trace file is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

**\*STOPTRC**

Tracing stops when the trace file is full of trace records.

---

## Program to call before trace (EXITPGM)

Specifies the name and library of a user-written program that is given control just prior to the collection of each trace record.

**Note:** Items being traced can be missed when an exit program is used. Do not use an exit program if you do not want to risk losing a trace record.

### Single values

**\*NONE**

No user-written program is called.

### Qualifier 1: Program to call before trace

**name** Specify the name of the user-written program called before each trace record is collected. This program can examine the trace record passed to it as a parameter, and alter the first two characters.

Information regarding the exit program interface format is provided in the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

*library-name*

Specify the library where the program is located.

Top

---

## Select procedures to trace (SLTPRC)

Specifies which ILE (Integrated Language Environment) procedure calls and returns are included in the trace.

### Single values

**\*ALL** All ILE procedure calls and returns are included in the trace.

**\*NONE**

No ILE procedure calls or returns are included in the trace.

### Element 1: Program

#### Qualifier 1: Program

*name* Specify the names of a maximum of 50 ILE programs or service programs for which all procedure calls and returns are to be included in the trace.

#### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library that contains the program or service program.

### Element 2: Type

**\*PGM** The specified program is a bound program.

**\*SRVPGM**

The specified program is a bound service program.

Top

---

## Thread ID to include (SLTTHD)

Specifies a list of up to twenty threads whose calls and returns are included in the trace. Only trace records for the specified thread identifiers are included.

### Single values

**\*ALL** All threads calls and returns are included in the trace.



### **\*SELECT**

A list of thread identifiers is shown from which the user can select up to twenty whose trace records are to be included.

### **Other values**

#### *hexadecimal-value*

Specify the identifiers of up to twenty threads whose trace records are to be included.

Top

---

## **Output (OUTPUT)**

Specifies whether the output from the command is printed with the job's spooled output or sent to a database file.

### **\*PRINT**

The output is printed with the job's spooled output.

### **\*OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Top

---

## **File to receive output (OUTFILE)**

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

### **Qualifier 1: File to receive output**

*name* Specify the name of the database file to which the command output is directed.

### **Qualifier 2: Library**

**\*LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

### **\*CURLIB**

The current library for the thread is used to locate the file. If no library is specified as the current library for the thread, the QGPL library is used.

*name* Specify the name of the library to be searched.

Top

---

## **Output member options (OUTMBR)**

Specifies the name of the database file member that receives the output of the command.

### **Element 1: Member to receive output**

#### **\*FIRST**

The first member in the file receives the output. If OUTMBR(\*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to**

**receive output (OUTFILE)** parameter. If the member already exists, you have the option to add new records to the end of the existing member or clear the member and then add the new records.

*name* Specify the name of the file member that receives the output. If it does not exist, the system creates it.

## Element 2: Replace or add records

### \*REPLACE

The system clears the existing member and adds the new records.

**\*ADD** The system adds the new records to the end of the existing records.

Top

---

## Examples

### Example 1: Tracing Flow of Control

```
TRCJOB TRCTYPE(*FLOW) MAXSTG(40)
```

This command traces the flow of the current job. Trace records are collected for each OPM program and ILE procedure call and return that occurs in the job. The trace file contains 40K of storage and wraps (oldest records are replaced by new records) if that amount of storage is filled with trace records.

### Example 2: Stopping the Trace Operation

```
TRCJOB SET(*OFF) OUTPUT(*OUTFILE)
        OUTFILE(QGPL/TRCJOB) OUTMBR(TRCDTA)
```

This command stops the trace and directs the output to the database file QGPL/TRCJOB. The output is directed to the member TRCDTA.

### Example 3: Tracing Flow of Control - Selecting Specific ILE Procedures

```
TRCJOB SET(*ON) TRCTYPE(*FLOW)
        SLTPRC((MYLIB/MYPGM1 *PGM) (MYLIB/MYSRVPGM1 *SRVPGM))
```

This command traces the flow of the current job. Trace records are collected for all OPM program calls and returns and the ILE procedure calls and returns of bound program MYPGM1 and bound service program MYSRVPGM1.

### Example 4: Tracing One Thread

```
TRCJOB SET(*ON) SLTTHD(00000001)
```

This command traces only the specified thread of the current job.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2C94

Error occurred during OUTFILE processing. Trace stopped.

#### CPF2C95

Trace already active.

**CPF2C96**  
Trace already off.

**CPF3510**  
User exit program not found in specified library.

**CPF3511**  
Trace already active.

**CPF3512**  
Trace already off.

**CPF3513**  
Cannot set Trace Off, trace started from another job.

**CPF3521**  
Not enough storage for the trace table.

**CPF3530**  
Conflicting entries in index QSERVICE.

**CPF3542**  
Job not traced because it is being serviced.

**CPF3548**  
Serviced job completed running.

**CPF3675**  
Cannot allocate QSYS library.

**CPF3909**  
Service command will not be processed.

**CPF3918**  
Service request canceled.

**CPF3925**  
Cannot open file &1.

**CPF3936**  
Job being serviced ended before trace started.

**CPF3950**  
Error message &2 received for file &1. Request ended.

**CPF3951**  
File &1 cannot be overridden by file name &2.

**CPF3957**  
Not authorized to use exit program library &2.

**CPF3969**  
Error during close of file &1. Output may not be complete.

**CPF6611**  
Error occurred during OUTFILE processing, trace ended.

**CPF6801**  
Command prompting ended when user pressed &1.

**CPF9810**  
Library &1 not found.

**CPF98A4**  
Thread restrictions exist for some other process.



---

## Trace REXX (TRCREX)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Trace REXX (TRCREX) command is used to turn the interpreter function on or off from command entry or from control language (CL) programming level.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	*RESULTS, *ALL, *COMMANDS, *ERROR, *FAILURE, *INTERMEDIATES, *LABELS, *NORMAL, *OFF	Optional, Positional 1

[Top](#)

---

### Trace option setting (SET)

Specifies the initial trace setting for the next REXX procedure that is run. This setting remains in effect unless changed through the REXX TRACE instruction.

The possible values are:

#### **\*RESULTS**

All clauses are traced before processing. Tracing operates as if the TRACE ?R instruction was used from within the REXX procedure.

**\*ALL** All clauses are traced before processing. Tracing operates as if the TRACE ?A instruction was used from within the REXX procedure.

#### **\*COMMANDS**

All host commands are traced before processing and any error return code is displayed. Tracing operates as if the TRACE ?C instruction was used from within the REXX procedure.

#### **\*ERROR**

Any host command resulting in an error return code is traced after processing. Tracing operates as if the TRACE ?E instruction was used from within the REXX procedure.

#### **\*FAILURE**

Any host command resulting in a failure is traced after processing together with the return code from the command. Tracing operates as if the TRACE ?F instruction was used from within the REXX procedure.

#### **\*INTERMEDIATES**

All clauses are traced before processing. Intermediate results during evaluation of expressions and substituted names are also traced. Tracing operates as if the TRACE ?I instruction was used from within the REXX procedure.

#### **\*LABELS**

Labels passed during processing are traced. Tracing operates as if the TRACE ?L instruction was used from within the REXX procedure.

**\*NORMAL**

Any failing host command is traced after processing. Tracing operates as if the TRACE ?N instruction was used from within the REXX procedure. This is the default setting.

**\*OFF** Nothing is traced. Tracing operates as if the TRACE O instruction was used from within the REXX procedure.

[Top](#)

---

## Examples

### Example 1: Tracing Host Commands

```
TRCREX SET(*COMMANDS)
```

This command causes all commands in by the REXX procedure to be shown before they are to be run.

### Example 2: Tracing Failing Host Commands

```
TRCREX SET(*NORMAL)
```

This command causes all commands that result in a FAILURE condition to be shown. This command shows the normal setting for the REXX tracing operation.

[Top](#)

---

## Error messages

None

[Top](#)

## Trace TCP/IP Application (TRCTCPAPP)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Trace TCP/IP Application (TRCTCPAPP) command is used by service personnel when trace information needs to be captured for one of the following TCP/IP applications: File Transfer Protocol (FTP), SMTP server, SMTP client, TELNET/VTAPI, host servers, Distributed Data Management (DDM), Virtual Private Network (VPN), Layer Two Tunneling Protocol (L2TP), certificate services, Point-to-Point Protocol (PPP), Quality Of Service (QOS), simple Network Time Protocol (NTP), directory services, HTTP server powered by Apache or packet rules.

**Restrictions:** To use this command, you must have either \*SERVICE special authority or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. For a given application, there could be only one trace active at a time on the system. The user must have \*USE authority to the line, network interface, or network server to be traced.

When the WCHJOB parameter is specified, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (\*JOBCTL) special authority.

**Note:** Whenever the term **host server** is seen within this help text, it refers to one of the application host servers: \*CENTRAL, \*DTAQ, \*RMTCMD, \*SIGNON, \*NETPRT, \*SVRMAP or \*DATABASE.

Top

### Parameters

Keyword	Description	Choices	Notes
APP	TCP/IP application	*FTP, *SMTPSVR, *SMTPCLT, *TELNET, *VTAPI, *CENTRAL, *DTAQ, *RMTCMD, *SIGNON, *NETPRT, *SVRMAP, *DDM, *VPN, *CERTSRV, *L2TP, *PPP, *QOS, *NTP, *HTTP, *DIRSRV, *DATABASE, *PKTRULES	Required, Positional 1
SET	Trace option setting	*ON, *OFF, *END, *CHK	Optional, Positional 2
MAXSTG	Maximum storage for trace	1-16000, *APP	Optional, Positional 3
TRCFULL	Trace full action	*WRAP, *STOPTRC	Optional, Positional 4
ADLTRC	Additional traces	Single values: *NONE Other values (up to 3 repetitions): *CMNTRC, *TCPIP, *SRCSINK	Optional, Positional 5
TRCPGM	Trace program	Single values: *NONE Other values: <i>Qualified object name</i>	Optional, Positional 6
	Qualifier 1: Trace program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
TITLE	Trace title	<i>Character value</i> , *DFT	Optional, Positional 7
USER	User profile	<i>Simple name</i>	Optional, Positional 8

Keyword	Description	Choices	Notes
MAILADR	Recipient mail address	Character value	Optional, Positional 9
HOST	Recipient host name	Character value	Optional, Positional 10
RMTNETADR	Remote network address	Element list	Optional, Positional 11
	Element 1: Address family	<u>*INET</u>	
	Element 2: IP address	Character value	
	Element 3: Subnet mask	Character value, <u>255.255.255.255</u>	
	Element 4: Port number	1-65535, <u>*ANY</u>	
LCLNETADR	Local network address	Element list	Optional, Positional 12
	Element 1: Address family	<u>*INET</u> , *UNIX	
	Element 2: IP address or UNIX path	Character value	
	Element 3: Subnet mask	Character value, <u>255.255.255.255</u>	
	Element 4: Port number	1-65535, <u>*ANY</u>	
DEVD	Device description	Generic name, name	Optional, Positional 13
DEVTYPE	Device type	Single values: *DSP, *PRT Other values (up to 6 repetitions): 5251, 5291, 5292, 3196, 3488, 3487, 3179, 3180, 5555, 3477, 3277, 3278, 3279, V100, 3812, 5553	Optional, Positional 14
TRCPNT	Trace point	Values (up to 12 repetitions): Character value	Optional, Positional 15
ARGLIST	Argument list	Character value	Optional, Positional 16
VPNSVR	Virtual private network server	Values (up to 2 repetitions): *KEYMGR, *CNNMGR	Optional, Positional 17
CERTTYPE	Certificate services type	<u>*ALL</u> , *DCM, *KEYMGR, *SSL, *OBJSIGN, *OTHER	Optional, Positional 18
DNS	Domain name service	<u>*NO</u> , *YES	Optional, Positional 19
PPPCNNPRF	PPP connection profile	Character value	Optional, Positional 20
TCPTRCDTA	TCP/IP data to trace	<u>*PPPALL</u> , *LCPNCP	Optional, Positional 21
QOSTRCTYPE	QOS trace type	<u>*ALL</u> , *POLICYD, *RSVPD	Optional, Positional 22
HTTPSVR	HTTP server instance	Character value	Optional, Positional 23
TRCLVL	Trace level	<u>*ERROR</u> , *INFO, *VERBOSE	Optional, Positional 24
PKTTRCPNT	Packet rules trace points	<u>*TRAFFIC</u> , *LOAD	Optional, Positional 25
CFGOBJ	Configuration object	Name	Optional, Positional 26
CFGTYPE	Type	*LIN, *NWI, *NWS	Optional, Positional 27
WCHMSG	Watch for message	Single values: <u>*NONE</u> Other values (up to 5 repetitions): Element list	Optional, Positional 28
	Element 1: Message identifier	Name	
	Element 2: Comparison data	Character value, <u>*NONE</u>	



Keyword	Description	Choices	Notes
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional, Positional 29
	Element 1: Message queue	Single values: <b>*SYSOPR</b> , *JOBLOG, *HSTLOG Other values: <i>Qualified object name</i>	
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b>	
WCHJOB	Watched job	Single values: * Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 30
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000001-999999, <b>*ALL</b>	
WCHLICLOG	Watch for LIC log entry	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 31
	Element 1: Major code	<i>Character value</i> , *ALL	
	Element 2: Minor code	<i>Character value</i> , *ALL	
	Element 3: Comparison data	<i>Character value</i> , <b>*NONE</b>	
WCHTIMO	Length of time to watch	1-43200, <b>1440</b> , *NOMAX	Optional, Positional 32
TRCPGMITV	Time interval	1-9999, <b>*NONE</b>	Optional, Positional 33

Top

---

## TCP/IP application (APP)

Specifies the TCP/IP application. This is a required parameter.

### \*CENTRAL

Specifies tracing for the central host server.

### \*CERTSRV

Specifies tracing for certificate services.

### \*DATABASE

Specifies tracing for the database host server.

### \*DDM

Specifies tracing for the Distributed Data Management (DDM) server.

### \*DIRSRV

Specifies tracing for directory services.

### \*DTAQ

Specifies tracing for the data queue host server.

\*FTP Specifies tracing for the File Transfer Protocol (FTP) server.

### \*HTTP

Specifies tracing for the HTTP server powered by Apache.

\*L2TP Specifies tracing for Layer Two Tunneling Protocol (L2TP).

### \*NETPRT

Specifies tracing for the network print host server.

\*NTP Specifies tracing for the Simple Network Time Protocol (SNTP) client.

**\*PKTRULES**

Specifies tracing for packet rules (PKTRULES).

**\*PPP** Specifies tracing for the Point-to-point protocol (PPP).

**\*QOS** Specifies tracing for the Quality of Service (QoS) server.

**\*RMTCMD**

Specifies tracing for the remote command host server.

**\*SIGNON**

Specifies tracing for the signon host server.

**\*SMTPCLT**

Specifies tracing for the SMTP client job(s) handling outbound mail processing connections.

**\*SMTPSVR**

Specifies tracing for the Simple Mail Transfer Protocol (SMTP) server job(s) handling inbound mail processing connections.

**\*SVRMAP**

Specifies tracing for the port mapper host server.

**\*TELNET**

Specifies tracing for the TELNET server.

**\*VPN** Specifies tracing for the Virtual Private Network (VPN) server.

**\*VTAPI**

Specifies tracing for the virtual terminal application programming interfaces.

Top

---

## Trace option setting (SET)

Specifies whether the collection of trace information starts, stops, or status is presented.

**\*ON** The collection of trace information is started.

**\*OFF** The collection of trace information is stopped and the trace information is written to spooled printer files of the user. For PPP traces, the trace files are also included in the OUTQ for the designated PPP profile.

**\*END** Tracing is ended and all trace information is deleted. No trace information output is created.

**\*CHK** The status of tracing for the specified application is checked. Messages are returned indicating whether or not tracing is active for the specified TCP/IP application, the command parameters specified from the last time that TRCTCPAPP was started for this application and other information related to the collection of trace information.

Top

---

## Maximum storage for trace (MAXSTG)

Specifies the maximum amount of storage in kilobytes (K) used for collected trace information.

**\*APP** Each application type defines a default buffer size.

- \*FTP - 4096K bytes per job
- \*SMTPCLT - 4096K bytes per job
- \*SMTPSVR - 4096K bytes per job
- \*TELNET - 16000K bytes per job

- \*VTAPI - 16000K bytes per job
- \*CENTRAL - 16000K bytes per job
- \*RMTCMD - 16000K bytes per job
- \*SIGNON - 16000K bytes per job
- \*DTAQ - 16000K bytes per job
- \*NETPRT - 16000K bytes per job
- \*SVRMAP - 16000K bytes per job
- \*DATABASE - 16000K bytes per job
- \*DDM - 16000K bytes per job
- \*VPN - 16000K bytes per job
- \*PKTRULES - 16000K bytes per job
- \*L2TP - 4096K bytes per job
- \*CERTSRV - 16000K bytes per job
- \*PPP - 4096K bytes per job
- \*QOS - 4096K bytes per job
- \*NTP - 4096K bytes per job
- \*HTTP - 16,000K bytes per job
- \*DIRSRV - 300K bytes per job

#### **1-16000**

Specify the maximum amount of storage, in kilobytes, used to store trace records (one kilobyte 1024 bytes).

Top

---

## **Trace full action (TRCFULL)**

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

### **\*WRAP**

When the trace buffer is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

### **\*STOPTRC**

Tracing stops when the trace buffer is full of trace records.

Top

---

## **Additional traces (ADLTRC)**

Specifies additional trace(s) to be started. When the TRCTCPAPP command is invoked interactively, the user will be prompted for any options to change on each of the selected traces. This parameter is valid for all applications.

### **\*NONE**

No additional trace will be included.

### **\*CMNTRC**

A communications trace will be included in the trace information for the specified application. **Note:** Due to resource limitations of the I/O hardware, multi-connection PPP profiles may not produce trace data for every connection started by the PPP profile.

#### \*TCPIP

A single TCP/IP component trace will be included in the trace information for the specified application.

#### \*SRCSINK

A source/sink component trace will be included in the trace information for the specified application.

Top

---

## Trace program (TRCPGM)

Specifies the name of a program to call for user defined trace commands and procedures. This parameter is valid for all applications.

For SET(\*ON), the trace program will be called:

- Before the application trace starts.
- After the communications and Licensed Internal Code (LIC) traces, if requested, start.
- If WCHMSG or WCHLICLOG parameter is specified, the trace program will be called:
  - After a match of a message identifier specified on WCHMSG parameter or LIC log specified on WCHLICLOG parameter occurred.
  - When the time interval specified on the TRCPGMITV parameter is reached.
  - When the length of time to watch specified on WCHTIMO parameter is reached.

For SET(\*OFF), the trace program will be called:

- Before the LIC traces, if requested, end.
- After the communications trace, if requested, ends.
- After the application trace ends.

For SET(\*END), the trace program will be called:

- After the LIC and communications traces, if requested, end.
- After the application trace ends.

When the TRCTCPAPP CPP detects an error with the trace program, it will display the TCP4537 diagnostic message. If watch for trace event facility is active, the trace and the watch for trace event facility will be ended, and CPI3999 message will be sent to the system operator message queue with reason code 04.

There are three input parameters and one output parameter associated with the trace program. The four parameters are required:

- 1 Trace option setting Input Char(10)
- 2 Application Input Char(10)
- 3 Error detected Output Char(10)
- 4 Comparison data Input Char(\*)

The possible values for the **Trace option setting (SET)** parameter are:

- \*ON, The collection of trace information is started.
- \*OFF, The collection of trace information is stopped and the trace information is written to spooled printer files of the user.
- \*END, Tracing is ended and all trace information is deleted. No trace information output is created.
- \*MSGID, A match on a message id specified on WCHMSG parameter occurred.

- \*LICLOG, A match on a LIC log specified on the WCHLICLOG parameter occurred.
- \*CMPDATA, The major and minor code of a LIC log matched, but the comparison data did not.
- \*INTVAL, The time interval specified on TRCPGMITV parameter is elapsed.
- \*WCHTIMO, The length of time to watch specified on WCHTIMO parameter is elapsed.

The possible values for the "Application" parameter are the same as the values for the APP parameter on the TRCTCPAPP command.

The possible values for the "Error detected" parameter are:

- \*ERROR, Error detected by customer trace program.
- \*CONTINUE, The trace and the watch for trace event facility will continue running.
- \*STOP, The trace and the watch for trace event facility will be ended.

The possible values for the "Comparison data" parameter when \*MSGID is specified on "Trace option setting" parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information

4 4 CHAR(7) Message ID

11 B CHAR(9) Reserved

20 14 BINARY(4) Offset to comparison data

24 18 BINARY(4) Length of comparison data

\* \* CHAR(\*) Message comparison data

The possible values for the "Comparison data" parameter when \*LICLOG or \*CMPDATA is specified on "Trace option setting" parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information

4 4 CHAR(4) LIC Log major code

8 8 CHAR(4) LIC Log minor code

12 C CHAR(8) LIC Log identifier

20 14 BINARY(4) Offset to comparison data

24 18 BINARY(4) Length of comparison data

\* \* CHAR(\*) LIC log comparison data

The possible values for the "Comparison data" parameter when \*ON, \*OFF, \*END, \*INTVAL or \*WCHTIMO is specified on "Trace option setting" parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information (always 4 at this time).

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

### Single values

#### \*NONE

No user supplied trace program will be called. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

#### Qualifier 1: Trace program

*name* Specify the name of the trace program to be called.

#### Qualifier 2: Library

\*LIBL The library list is used to locate the program.

#### \*CURLIB

The current library is used to locate the program. If no library is specified as the current library, the QGPL library is used.

#### *trace-program-library*

Specify the name of the library where the program is located.

Top

---

## Trace title (TITLE)

Specifies the title that is printed on each page of the spooled file which contains the collected trace information. This parameter is only valid when SET(\*OFF) is specified.

\*DFT The default trace description title "TRCTCPAPP Output" is used.

#### *character-value*

Specify up to 50 characters to be used as the title on each page of the trace output spooled file.

Top

---

## User profile (USER)

Only trace information associated with a specific user profile will be collected. This parameter is only valid when APP(\*FTP) is specified.

*name* Specify the name of the user profile for which trace information is to be collected.

Top

---

## Recipient mail address (MAILADR)

Only trace information associated with a specific recipient mail address will be collected. This parameter is only valid when APP(\*SMTPSVR) or APP(\*SMTPCLT) is specified.

### *character-value*

The recipient mail address (up to 255 characters) must have the following format:  
'userid@abc.def.com'

Top

---

## Recipient host name (HOST)

Only trace information associated with a specific recipient host name will be collected. This parameter is only valid when APP(\*SMTPCLT) is specified.

### *character-value*

Specify the recipient host name (up to 255 characters). The name must have the following format:  
'abc.def.com'

Top

---

## Remote network address (RMTNETADR)

The user may limit the amount of information collected by entering an address family, remote TCP/IP address, subnet mask and port number. This parameter is only valid when APP(\*FTP), APP(\*SMTPSVR), APP(\*DDM), APP(host server), APP(\*TELNET), APP(\*VTAPI) or APP(\*L2TP) is specified. **Note:** The only valid filter for L2TP is the IP Address element.

### Element 1: Address family

\*INET The filter for AF\_INET address family.

### Element 2: IP address

#### *character-value*

Specify the remote TCP/IP address for which trace information is to be collected.

### Element 3: Subnet mask

#### 255.255.255.255

Tracing will be done for only the IP address specified as the second element of this parameter.

#### *character-value*

Specify the subnet mask for which trace information is to be collected.

### Element 4: Port number

\*ANY The TCP/IP port number defaults to \*ANY which implies traffic associated with any port on the remote system (and qualified by the IP address and subnet mask) will be traced.

#### 1-65535

Specify the port number to be used. If a number is specified, a subnet mask value must also be specified.

Top

---

## Local network address (LCLNETADR)

The user may limit the amount of information collected by entering an address family, local TCP/IP address, subnet mask and port number. This parameter is only valid when APP(\*DDM), APP(host server), APP(\*TELNET) or APP(\*VTAPI) is specified.

### Element 1: Address family

**\*INET** The filter for AF\_INET address family.

**\*UNIX**

The filter for AF\_UNIX address family. Note that \*UNIX is a valid choice for only APP(\*DDM) or APP(host server).

### Element 2: IP address or UNIX path

#### *character-value*

When \*INET is specified for element 1 of this parameter, specify the local TCP/IP address for which trace information is to be collected.

When \*UNIX is specified for element 1 of this parameter, specify the UNIX path for which trace information is to be collected. Note that a UNIX-path can be entered for only APP(\*DDM) or APP(host server).

### Element 3: Subnet mask

**255.255.255.255**

Tracing will be done for only the IP address specified as the second element of this parameter.

#### *character-value*

Specify the subnet mask for which trace information is to be collected.

### Element 4: Port number

**\*ANY** The TCP/IP port number defaults to \*ANY which implies traffic associated with any port on the local system (and qualified by the IP address and subnet mask) will be traced.

**1-65535**

Specify the port number to be used. If a number is specified, a subnet mask value must also be specified.

Top

---

## Device description (DEV D)

The user may limit the amount of information collected by entering a device description name. Once the device description is associated with a given TELNET or VTAPI session, all trace information associated with it will be collected. This parameter is only valid when APP(\*TELNET) or APP(\*VTAPI) is specified.

**name** Specify the name of a device description for which trace information is to be collected.

#### *generic-name*

Specify a generic name for device descriptions for which trace information is to be collected. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, CMN\*. If a generic name is specified, then all device descriptions with names that begin with the generic name, and for which the user has authority, will have trace information collected.

Top



---

## Device type (DEVTYPE)

One or more valid device types may be specified. Only the trace information associated with activity for those devices will be traced. If \*DSP or \*PRT is specified, no other values may be entered for this parameter. This parameter is only valid when APP(\*TELNET) or APP(\*VTAPI) is specified.

**\*DSP** The information collected is only for display device types.

**\*PRT** The information collected is only for printer device types.

### *device-type*

The information collected is only for the specified device types. Up to six types may be specified. The valid types include: 5251, 5291, 5292, 3196, 3488, 3487, 3179, 3180, 5555, 3477, 3277, 3278, 3279, V100, 3812 and 5553.

Top

---

## Trace point (TRCPNT)

You can limit the trace points that are placed in the trace buffer by entering the list of those trace points for this parameter. Up to 12 trace points may be specified. This parameter is only valid when APP(\*TELNET), APP(\*VTAPI), APP(\*DDM) or APP(host server) is specified.

### *character-value*

Specify the trace point identifier. Each trace point identifier may be up to 8 characters.

For TELNET/VTAPI trace points, specify 'TG#xxxxx', 'TG+xxxxx' or 'TG-xxxxx' where 'xxxxx' defines the specific trace point. The following TELNET/VTAPI trace points can also be specified: TGTELM, TGTELO, TGEXCP, TGREQPO, TGRIO, TGRPO, TGUTIL, TGVTERM, TGV TIN, TGV TINI, TGV TM or TGV TOUT.

For host/DDM server trace points, specify 'Qccxxxx' where 'ccc' is the component ID of the host/DDM server and 'xxxx' defines the specific trace point.

Top

---

## Argument list (ARGLIST)

Only trace information associated with this specific argument list will be included in the trace information collected. The argument list contains data like the debug level and special trace requests. This parameter is only valid when APP(\*VPN), APP(\*QOS), APP(\*PKTRULES), APP(\*PPP) or APP(\*DIRSRV) is specified.

### *character-value*

Specify the argument list (up to 255 characters).

QoS allows the following argument list values:

- lvl=1** The lvl=1 argument logs errors that are associated with system operations. One example might be that the system is out of memory. The result of these types of errors is that the QoS server will not be able to run.
- lvl=2** The lvl=2 argument includes all lvl=1 information. In addition, the lvl=2 argument logs internal errors identified with the operation of the QoS server. The usual cause of these types of errors is that unexpected errors have been encountered in a server operation. A lvl=2 error is considered a condition for an APAR.
- lvl=3** The lvl=3 argument includes all lvl=1 and lvl=2 information. In addition, the lvl=3 argument logs the basic operational activities of the QoS server. Examples might be the loading of rules or the sending of a STRTCPSVR SERVER(\*QOS) RESTART(\*QOS) command.

**lvl=4** The lvl=4 argument includes all lvl=1, lvl=2 and lvl=3 information. In addition, the lvl=4 argument logs all traced activities of the QoS server.

Top

---

## Virtual private network server (VPNSVR)

Specifies whether the trace information is to be collected for the VPN key manager or the VPN connection manager. If no value is specified for this parameter, trace information is to be collected for both the VPN key manager and the VPN connection manager. This parameter is only valid when APP(\*VPN) is specified.

**\*KEYMGR**

Filtering of trace information is done to include the VPN key manager.

**\*CNNMGR**

Filtering of trace information is done to include the VPN connection manager.

Top

---

## Certificate services type (CERTTYPE)

Only trace information associated with a specific certificate services type will be included in the captured trace information. This parameter is only valid when APP(\*CERTSRV) is specified.

**\*ALL** No filtering of trace information is done for certificate services type.

**\*DCM** Filtering of trace information is done to include only the DCM certificate services type.

**\*KEYMGR**

Filtering of trace information is done to include only the VPN key manager certificate services type.

**\*SSL** Filtering of trace information is done to include only the SSL certificate services type.

**\*OBJSIGN**

Filtering of trace information is done to include only the OBJSIGN certificate services type.

**\*OTHER**

Filtering of trace information is done to include only a certificate services type not listed above.

Top

---

## Domain name service (DNS)

Specifies whether only trace information associated with domain name service (DNS) resolution will be collected. This parameter is only valid when APP(\*SMTPCLT) is specified.

**\*NO** No filtering of trace information is done for DNS resolution.

**\*YES** Trace information includes only trace points associated with DNS resolution.

Top

---

## PPP connection profile (PPPCNNPRF)

Trace information associated with a specific PPP connection profile will be collected. The default trace information provided is one joblog and one connection dialog spooled file (containing the PPP dialog trace) for each connection started by the PPP connection profile, one copy of the PPP profile settings, and one copy of the line description used by the profile. When selected by the user there could also be one SRCSINK component trace per connection, one Communications trace per connection and a single TCPIP component trace. This parameter is required when APP(\*PPP) is specified.

### *character-value*

Specify the PPP connection profile for which trace information is to be collected.

Top

---

## TCP/IP data to trace (TCPTRCDTA)

Specifies what additional data will be collected when ADLTRC(\*TCPIP) is selected. This parameter is only valid when APP(\*PPP) is specified. If APP(\*PPP) is specified and ADLTRC(\*TCPIP) is not specified, this parameter will be ignored.

### \*PPPALL

All data on the PPP connection will be traced.

### \*LCPNCP

Only LCP and NCP data on the PPP connection will be traced.

Top

---

## QOS trace type (QOSTRCTYPE)

Only trace information associated with a specific QOS trace type will be included in the trace information collected. This parameter is only valid when APP(\*QOS) is specified.

\*ALL Filtering of trace information is done to include both servers.

### \*POLICYD

Filtering of trace information is done to include the QOS policy server.

### \*RSVPD

Filtering of trace information is done to include the RSVP (Resource reSerVation Protocol) server.

Top

---

## HTTP server instance (HTTPSVR)

This parameter will determine which HTTP server instance to trace. It is only valid and required when APP(\*HTTP) is specified.

Top

---

## Trace level (TRCLVL)

Specifies the level of granularity of the service trace. This parameter is only valid when APP(\*HTTP) is specified.

### \*ERROR

The service trace will contain trace records for all error return codes or exception conditions.

#### **\*INFO**

The service trace will contain \*ERROR level trace records as well as trace records for entry and exit points from application level APIs and API parameters.

#### **\*VERBOSE**

The service trace will contain \*INFO level trace records as well as trace records for debugging control flow or data corruption.

Top

---

## **Packet rules trace points (PKTTRCPNT)**

Specifies a keyword that represents trace point values which will be displayed when the Trace Internal (TRCINT) panel is displayed. This parameter is only valid when APP(\*PKTRULES) and ADLTRC(\*TCPIP) are specified.

#### **\*TRAFFIC**

The following trace points for packet filter evaluation will be displayed on the Trace Internal panel: 8110-8111, 8120-8123 and 8420.

#### **\*LOAD**

The following trace points for the audit and load of rules will be displayed on the Trace Internal panel: 8100-8105 and 8430-8438.

Top

---

## **Configuration object (CFGOBJ)**

Specifies the configuration object to trace. The object can be either a line description, a network interface description, or a network server description. This parameter is only valid when ADLTRC(\*CMNTRC) is specified.

*name* Specify the name of the configuration object to be traced.

Top

---

## **Type (CFGTYPE)**

Specifies the type of configuration description to trace. This parameter is only valid when ADLTRC(\*CMNTRC) is specified.

**\*LIN** The type of configuration object is a line description.

**\*NWI** The type of configuration object is a network interface description.

**\*NWS** The type of configuration object is a network server description.

Top

---

## **Watch for message (WCHMSG)**

Specifies up to five message identifiers which are to be watched for. If a value other than \*NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace exit program is called; if no trace exit program is defined, the trace stops.

### **Single values**

### \*NONE

No messages will be watched for.

#### **Element 1: Message identifier**

*name* Specify the 7-character message identifier to be watched for.

#### **Element 2: Comparison data**

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data includes the specified text, the watched for condition is true. If the message data does not contain the specified text, the trace function continues.

### \*NONE

No comparison data is specified. If a message matching the specified message ID is added to the specified message queue or log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the message data of the watched for message. If this text is found anywhere in the message data of a watched for message, the watch condition is considered to be true. This text is case sensitive. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

Top

---

## **Watched message queue (WCHMSGQ)**

Specifies where to watch for the message identifiers specified for the **Watch for message (WCHMSG)** parameter. You can specify to watch the message being added to the system operator message queue, the history log, other message queues, and job logs. Up to three message queues or special values can be specified.

#### **Element 1: Message queue**

##### **Single values**

### \*SYSOPR

Watch messages added to the system operator message queue (QSYSOPR message queue in library QSYS).

### \*HSTLOG

Watch messages added to the history log QHST.

### \*JOBLOG

Watch messages added to the job logs of jobs specified by the WCHJOB parameter.

#### **Qualifier 1: Message queue**

*name* Specify the name of the message queue to watch.

#### **Qualifier 2: Library**

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library where the message queue is located.

---

## Watched job (WCHJOB)

Specifies the job whose job log is watched for the messages specified on the WCHMSG parameter. The specified job will only be watched if \*JOBLOG is specified on the WCHMSGQ parameter. Up to five job names may be specified.

### Single values

\* Only the job log of the job that issued this trace command is watched.

### Element 1: Job name

#### Qualifier 1: Job name

##### *generic-name*

Specify the generic name of the job to be watched. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

*name* Specify the name of the job to be watched.

#### Qualifier 2: User

*name* Specify the name of the user of the job to be watched.

#### Qualifier 3: Number

\*ALL All jobs with the specified job name and user name are watched.

##### *000001-999999*

Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier is specified.

---

## Watch for LIC log entry (WCHLICLOG)

Specifies up to five licensed internal code (LIC) log entry identifiers which are to be watched for. Each LIC log entry contains a major and a minor code. The watched for condition will be met if a LIC log entry is added that matches the specified major and minor codes and any comparison data specified. When the watched for log entry is added to the LIC log, the trace exit program is called, even when the comparison data specified does not match; if no trace exit program is defined, the trace stops.

### Single values

### \*NONE

No LIC log entries will be watched for.

### **Element 1: Major code**

**\*ALL** Any LIC log entry major code will be considered to be a match. If **\*ALL** is specified for the major code, you cannot specify **\*ALL** for the LIC log entry minor code.

#### *character-value*

Specify the LIC log major code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### **Element 2: Minor code**

**\*ALL** Any LIC log entry minor code will be considered to be a match. If **\*ALL** is specified for the minor code, you cannot specify **\*ALL** for the LIC log entry major code.

#### *character-value*

Specify the LIC log minor code to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the four-digit code. A question mark is a wildcard character that will match any digit in that position. Up to three wildcard characters can be specified.

### **Element 3: Comparison data**

Specify comparison data to be used if a log entry matching the specified major and minor codes is added to the licensed internal code (LIC) log. If this text is found in the LIC log entry data fields of the watched for log entry, the watched for condition is true. If this text is not found in the LIC log entry data fields of the watched for log entry and no exit program is specified on the TRCPGM parameter, the trace function continues. If the log entry matches the specified major and minor codes and an exit program is specified on the TRCPGM parameter, but the entry data does not contain the specified text, the exit program is called to determine if the trace should continue or stop.

### \*NONE

No comparison data is specified. If a LIC log entry matching the specified major and minor codes is added to the LIC log, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the entry data of the watched for log entry. If this text is found in the LIC log entry data fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server name, job name, user ID, job number, thread ID, exception ID, LIC module compile binary timestamp, LIC module offset, LIC module RU name, LIC module name, LIC module entry point name. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

When watching for an exception ID, all four hexadecimal digits of the exception ID must be specified. Also, the prefix MCH may be specified if you want to compare only against the exception ID field and avoid possible substring matches with the other fields.

Top

---

## **Length of time to watch (WCHTIMO)**

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

**1440** The time limit for watching for a particular message or LIC log entry is 1440 minutes (24 hours).

**\*NOMAX**

There is no time limit for watching for a particular message or LIC log entry.

**1-43200**

Specify the number of minutes that the trace will be active while none of the watched for conditions have been met. You can specify up to 43200 minutes (30 days).

Top

---

## Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

**\*NONE**

No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

**1-9999** Specify the interval of time, in seconds, of how often the trace exit program will be called.

Top

---

## Examples

### Example 1: Starting Trace for Database

```
TRCTCPAPP APP(*DATABASE) SET(*ON) TRCPNT(QZDA1050 QZDA1060)
LCLNETADR(*INET
          '9.130.69.22'
          '255.255.255.255' 8471)
ADLTRC(*CMNTRC) TRCPGM(PGMLIB/PROG1)
CFGOBJ(TESTLIN) CFGTYPE(*LIN)
```

This command will start tracing for the database host server. Tracing information associated with the AF\_INET address family, a local TCP/IP address of 9.130.69.22, a subnet mask of 255.255.255.255, port number of 8471 and trace points of QZDA1050 and QZDA1060 will be collected. A communication trace will be included in the trace information. TESTLIN is the name of the configuration object to trace. This object is a line (\*LIN) description. Trace program PROG1 in library PGMLIB, with its user-defined trace commands and procedures, will be called. Tracing for the other TCP applications is not affected.

### Example 2: Check Status of Database Tracing

```
TRCTCPAPP APP(*DATABASE) SET(*CHK)
```

This command is used to check the status of the tracing for the database host server job. Assume that the last command entered was from "Example 1" above. The format of the response to this command would be a set of messages that would look similar to the following:

```
TCP45B7 TRCTCPAPP APP(*DATABASE) SET(*ON)
          TRCPNT(QZDA1050 QZDA1060)
          LCLNETADR(*INET '9.130.69.22'
          '255.255.255.255' 8471)
          MAXSTG(*DFT) TRCFULL(*WRAP) ADLTRC(*CMNTRC)
          TRCPGM(PGMLIB/PROG1) CFGOBJ(TESTLIN)
          CFGTYPE(*LIN)
TCP45B1 Tracing active for *DATABASE at 20:15:14 on 03/15/01 by
043432/TRCUSER/QPADEV000B.
TCP45B2 Data capture begun for *DATABASE.
TCP45B3 Data buffer wrapped for *DATABASE.
```

### Example 3: Ending Database Connection Tracing



```
TRCTCPAPP APP(*DATABASE) SET(*OFF)
```

This command first ends any currently active application trace for the database host server, followed by ending the TCP/IP component trace. If tracing was active, output trace records are formatted and placed into a spool file. A similar message will be found in the user's joblog:

```
TCP4588 Trace data for application DATABASE formatted: QZDA001915.
```

If tracing is not active, then the following message will be returned to the user:

```
TCP4580 Tracing off, SET(*OFF) not valid.
```

#### Example 4: Starting Trace for Packet Rules

```
TRCTCPAPP APP(*PKTRULES) SET(*ON)
          ARGLIST('DebugLvl=1 TraceLvl=2')
          ADLTRC(*TCPIP) PKTTRCPNT(*LOAD)
```

This command will start tracing for packet rules. Tracing information associated with the specific argument list will be collected. A component trace will be included in the trace information, using trace points of 8100-8105 and 8430-8438. Tracing for the other TCP applications is not affected.

#### Example 5: Starting Trace for FTP

```
TRCTCPAPP APP(*FTP) SET(*ON)
          RMTNETADR(*INET '9.130.69.16' '255.255.255.255' 5)
```

This command will start tracing for the FTP server. Tracing information associated with the AF\_INET address family, a remote TCP/IP address of 9.130.69.16, a subnet mask of 255.255.255.255 and port number of 5 will be collected. Tracing for the other TCP applications is not affected.

#### Example 6: Starting Trace for TELNET

```
TRCTCPAPP APP(*TELNET) SET(*ON) DEVD(QPADEV*)
```

This command will start tracing for the TELNET server. Trace information will be collected for all device descriptions with names that begin with "QPADEV". The user must have authority to these specific device descriptions. Tracing for the other TCP applications is not affected.

Top

---

## Error messages

### \*ESCAPE Messages

#### TCP4595

Trace not started.

Top



## Trace TCP/IP Route (TRCTCPRTE)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Trace TCP/IP Route (TRCTCPRTE) command, also known as TRACEROUTE, traces the route of IP packets to a user-specified destination system. The route can involve many different systems along the way. Each system along the route is referred to as a **hop**. You can trace all hops along the route or specify the starting and ending hops to be traced.

The route is traced by sending packets (called **probes**) to the destination system. Each probe contains an upper limit (called **Time To Live** or **TTL**) on the number of hop systems the probe can pass through.

**Note:** In IP Version 6, **Time To Live** (TTL) is called the **hop limit**.

A route is traced by successively incrementing the TTL of the probe packets by one hop. The trace ends when either a probe response is received from the destination system or when the probe Time To Live value equals the maximum allowed.

Responses from the probe packets are sent as messages to the job log or as queue entries to a user-specified data queue.

Top

### Parameters

Keyword	Description	Choices	Notes
RMTSYS	Remote system	<i>Character value</i>	Required, Positional 1
RANGE	Range of hops to probe	<i>Element list</i>	Optional
	Element 1: Starting probe TTL (hop limit)	1-255, <u>1</u>	
	Element 2: Maximum probe TTL (hop limit)	1-255, <u>30</u>	
PROBES	Probes sent per hop	1-64, <u>3</u>	Optional
WAITTIME	Response wait time	1-120, <u>3</u>	Optional
PKTLEN	Packet length (in bytes)	40-65535, <u>40</u>	Optional
OUTPUT	Output	<u>*MSG</u> , *VERBOSE, *DTAQ	Optional
DTAQ	Data queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Data queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *CURLIB, *LIBL	
ADRVERFMT	Address version format	*CALC, *IP4, *IP6	Optional
LCLINTNETA	Source IP address	<i>Character value</i> , *ANY	Optional
RMTPORT	Base remote port	1-65535, <u>33434</u>	Optional
NAMELOOKUP	Lookup host names	*YES, *NO	Optional
PROBEPCL	Probing protocol	*ICMP, *UDP	Optional
FRAGMENT	Allow fragmentation	*TCPA, *NO, *YES	Optional

---

## Remote system (RMTSYS)

Specifies the remote system name (255 characters) or IP address of the destination system.

### *character-value*

Specify the remote system name or IP address. Either a valid IP Version 4 or IP Version 6 address will be accepted.

Top

---

## Range of hops to probe (RANGE)

Specifies the range of hop systems from which probe responses are expected. Each probe specifies a TTL (Time To Live) integer value. This TTL value is the maximum number of hops the probe can traverse. For example, a probe packet with a TTL of 3 can pass through at most 3 hop systems before the hop system discards the probe and sends information back to the system from which the probe originated.

Element 1 specifies the first TTL value sent in probe packets. Element 2 specifies the last TTL value sent in probe packets. Trace information is generated from each hop system which discards a probe packet because the TTL value in the probe is reached or when the destination system is reached.

### Element 1: Starting probe TTL (hop limit)

1 The default starting hop is 1.

1-255 Specify the first hop limit TTL number used for probe packets.

### Element 2: Maximum probe TTL (hop limit)

30 The default ending hop is 30.

1-255 Specify the maximum number of hops a probe can traverse to reach the destination system.

Top

---

## Probes sent per hop (PROBES)

Specifies the number of probe packets sent to each hop system for each probe TTL (hop limit) value in the range specified by the RANGE parameter.

3 The default number of probes is three.

1-64 Specify the number of probes to send.

Top

---

## Response wait time (WAITTIME)

Specifies the maximum time, in seconds, to wait for a response from a hop system to each probe.

3 Wait up to 3 seconds for a response.

1-120 Specify the maximum number of seconds to wait for a response.

Top

---

## Packet length (in bytes) (PKTLEN)

Specifies the total length, in bytes, of the IP packet sent for each probe.

**40** The probe packet length is 40 bytes.

**40-65535**  
Specify the number of bytes in the probe IP packet.

Top

---

## Output (OUTPUT)

Specifies where the results obtained from sending the probe packets is sent. Information is sent for each hop until the destination system is reached, including hop count, average round-trip time, IP address of the hop and host name of the hop.

**\*MSG** Results are output as messages sent to the job log of the job in which the command is issued.

**\*VERBOSE**

Results are output as messages sent to the job log of the job in which the command is issued. All responses received are displayed. Results are not limited to ICMP TIME\_EXCEEDED and PORT\_UNREACHABLE responses.

**\*DTAQ**

Results from probes are placed on the data queue specified by the Data Queue (DTAQ) parameter.

Top

---

## Data queue (DTAQ)

Specifies the data queue on which entries are placed. When a data queue is specified, messages are not sent to the job log unless an error occurs.

Each queue entry contains the response to a probe if one was received or indicates that no probe response was received. The specified data queue must have a queue entry length of at least 32 characters and must exist when this command is issued.

### Qualifier 1: Data queue

*name* Specify the name of the data queue.

### Qualifier 2: Library

**\*LIBL** All libraries in the job's library list are searched.

**\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to search.

Top

---

## Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

### \*CALC

The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. TRCTCPRTE (TRACEROUTE) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution is used in determining the IP address.

**\*IP4** Use the IP Version 4 host name resolution method.

**\*IP6** Use the IP Version 6 host name resolution method.

Top

---

## Source IP address (LCLINTNETA)

Specifies how the source IP address in the probe packet is chosen.

**\*ANY** The source IP address in the probe packets is chosen by the system. The system may use any active local interface which can reach the remote system.

### *character-value*

Specify the local interface to use as the source IP address.

Top

---

## Base remote port (RMTPORT)

Specifies the base UDP port number used in probes.

**33434** Use the default base UDP port number of 33434.

**1-65535**

Specify the base UDP port number to be used in probes.

Top

---

## Lookup host names (NAMELOOKUP)

Specifies whether IP addresses will be resolved to the host name.

**\*YES** The address will be resolved to the host name.

**\*NO** The address will not be resolved to the host name.

Top

---

## Probing protocol (PROBEPCL)

Specifies the protocol used when sending probe packets.

### \*ICMP

The probes sent to the destination system are ICMP (Internet Control Message Protocol) Echo Request packets.

**\*UDP** The probes sent to the destination system are UDP (User Datagram Protocol) packets.

Top

---

## Allow fragmentation (FRAGMENT)

Specifies how the setting of the "Do Not Fragment" option in the IP header of the probe packet is determined.

### \*TCPA

The system sets the option based on the setting of the IP Path MTU Discovery TCP/IP attribute.

**Note:** Use the Change TCP/IP Attributes (CHGTCPA) command to change the value of this attribute.

**\*NO** The "Do Not Fragment" option is always specified.

**\*YES** The "Do Not Fragment" option is never specified.

Top

---

## Examples

### Example 1: Trace Entire Route

```
TRCTCPRTE  RMTSYS('130.14.3.5')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is '130.14.3.5'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP Echo Request packet. Results received are sent as messages to the job log.

### Example 2: Trace Partial Route

```
TRCTCPRTE  RMTSYS('AAA.BBBB.COM') RANGE(3 7)
           PROBES(5) PROBEPC(*UDP)
           OUTPUT(*DTAQ) DTAQ(MYLIB/MYDATAQ)
```

This command traces the route between the local iSeries system and the destination system whose host name is 'AAA.BBBB.COM'. Five probe packets will be sent for the starting range value of 3. Each probe will be a UDP packet inside an IP packet that is 40 bytes long. Each of these 5 probes will specify a TTL of 3. If system AAA.BBB.COM can be reached by passing through at most 2 hop systems then the trace will terminate at this point.

If system AAA.BBB.COM is further than 2 hops, another set of 5 probe packets will be sent to the destination AAA.BBB.COM. Each of these 5 probes will specify a TTL of 4. This is repeated until either system AAA.BBB.COM responds to a probe or 5 probes with a TTL of 7, the ending range value, are sent. Any results received are placed as queue entries on data queue MYDATAQ in library MYLIB.

### Example 3: Trace Route with an IP Version 6 Address

```
TRCTCPRTE  RMTSYS('1:2:3:4:5:6:7:8')
```

This command traces the entire route between the local iSeries system and the destination system whose IP address is 1:2:3:4:5:6:7:8. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

**Note:** A colon character (:) found in the parameter value signifies an IP Version 6 address and will cause an ICMP6 echo request packet to be generated.

### Example 4: Trace Route with an IP Version 6 Host Name

```
TRCTCPRTE  RMTSYS('IP6HOST')
```

This command traces the entire route between the local iSeries system and the destination system whose host name is 'IP6HOST'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

The default "Address version format" is \*CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (\*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to trace the route.

#### **Example 5: Trace Route with an IP Version 6 Host Name and Explicitly Use IP Version 6 Host Name Resolution**

```
TRCTCP RTE RMTSYS('IP6HOST') ADRVERFMT(*IP6)
```

This command traces the entire route between the local iSeries system and the destination system whose host name is 'IP6HOST'. Three probe packets will be sent to each hop system. Each IP probe packet will be 40 bytes long and will contain an ICMP6 Echo Request packet. Results received are sent as messages to the job log.

This example differs from example 4 in that only a valid IP version 6 resolved address, for IP6HOST, will be used when attempting to trace the route.

Top

---

## **Error messages**

### \*ESCAPE Messages

#### **TCP3250**

DTAQ parameter value required with OUTPUT(\*DTAQ).

#### **TCP3251**

DTAQ parameter not valid when OUTPUT(\*MSG) specified.

#### **TCP3252**

Starting range value greater than range limit.

Top



---

## Remove Mounted FS (UNMOUNT)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Remove Mounted File System (UNMOUNT) command will make a previously mounted file inaccessible within the integrated file system name space. The file system to be made inaccessible can be a user-defined file system (\*UDFS) on the local system, a remote file system accessed through a Network File System server (\*NFS), or a local or remote NetWare file system (\*NETWARE). If any of the objects in the file system are in use, the command will return an error message to the user. Note that if any part of the file system has itself been mounted over, then this file system cannot be unmounted until it is uncovered.

This command can also be issued using the following alternative command name:

- RMVMFS

For more information about Network File System commands, see OS/400 Network File System book, SC41-5714

### Restrictions:

1. The user must have input/output (I/O) system configuration (\*IOSYSCFG) special authority to use this command.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS, *NETWARE, *ALL	Required, Positional 1
MNTOVRDIR	Directory mounted over	<i>Path name</i> , *ALL	Optional
MFS	Mounted file system	<i>Path name</i>	Optional

[Top](#)

---

## Type of file system (TYPE)

Specifies the type of file system to be unmounted.

**\*NFS** The file system to be unmounted is a Network File System. When \*NFS is specified, a directory must be specified for the **Directory mounted over (MNTOVRDIR)** parameter.

### \*UDFS

The file system to be unmounted is a user-defined file system. When \*UDFS is specified, either the MNTOVRDIR or the **Mounted file system (MFS)** parameter may be specified.

### \*NETWARE

The file system to be unmounted is a NetWare file system. When \*NETWARE is specified, a directory must be specified for the MNTOVRDIR parameter.

**\*ALL** File systems of all types are to be unmounted. If **\*ALL** is specified, a value must be specified for the **MNTOVRDIR** parameter, and that value may be **\*ALL**.

This is a required parameter.

Top

---

## Directory mounted over (MNTOVRDIR)

Specifies the path name of the directory that was mounted over ('covered') by a previous **ADDMFS** (Add Mounted File System) or **MOUNT** command.

### *'directory-path-name'*

The specified directory that was previously mounted over will be uncovered. If **TYPE(\*ALL)** was specified, all file systems mounted over the specified directory will be unmounted. If a specific file system type was specified for the **Type of file system (TYPE)** parameter, the file system mounted most recently over the specified directory will be unmounted only if it matches the specified **TYPE** value.

**\*ALL** All directories that were previously mounted over will be uncovered. If **\*ALL** is specified, **\*ALL** must be specified for the **TYPE** parameter.

This is a required parameter.

Top

---

## Mounted file system (MFS)

Specifies the path name of the file system to be unmounted. This parameter can only be used to unmount a Block Special File (**\*BLKSF**), when **\*UDFS** is specified for the **Type of file system (TYPE)** parameter.

Top

---

## Examples

The alternative command name for **UNMOUNT** is **RMVMFS**. The following examples use the alternative command name, but **UNMOUNT** can be replaced directly for **RMVMFS** in all of them.

### Example 1: Unmounting a Directory

```
RMVMFS TYPE (*NFS) MNTOVRDIR('/tools')
```

This command unmounts a Network File System that is accessible on directory */tools*.

### Example 2: Unmounting a User-Defined File System

```
RMVMFS TYPE(*UDFS) MFS('/DEV/QASP02/CUST1UDFS')
```

This command unmounts the user-defined file system */dev/qasp02/custudfs*.

Top

---

## Error messages

### \*ESCAPE Messages

**CPFA0A9**

Object not found. Object is &1.

Top



---

## Update Data with Temp Program (UPDDTA)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Control Language (CL) command UPDDTA creates and runs a temporary DFU program. You can use this temporary program to enter new records or change existing records in an existing database file.

---

### Error messages for UPDDTA

#### \*STATUS Messages

##### DFU0251

DFU is creating temporary program &1 for you to run.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	Data base file	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST</i>	Optional, Positional 2

Top

---

### Data base file (FILE)

Specifies the qualified name of the data file to be updated.

**\*LIBL** DFU will use your library list to search for a specified program.

#### **\*CURLIB**

Type \*CURLIB to use your current library. If no current library entry exists in the library list, QGPL is used. If you do not specify a library name, \*LIBL is used.

Top

---

### Member (MBR)

Specifies the member in the file you want to update.

The possible values are:

#### \*FIRST

You will update the first member of the file.

#### **member-name**

Type the name of the member you want to update.

[Top](#)

---

## Examples

None

[Top](#)

---

## Error messages

### \*STATUS Messages

#### DFU0251

DFU is creating temporary program &1 for you to run.

[Top](#)

# Update Program (UPDPGM)

Where allowed to run: All environments (\*ALL)  
 Threadsafte: No

Parameters  
 Examples  
 Error messages

The Update Program (UPDPGM) command can be used to replace modules of an Integrated Language Environment (ILE) bound program with other modules on the system, without requiring you to change or recompile the bound program. Modules being replaced must be module objects (\*MODULE) on the system.

Other jobs running the bound program can run while the program is being updated with this command. The currently running bound program is moved to library QRPLOBJ and an updated version of the bound program is inserted into the library of the bound program. Current activations of the program will continue running, using the version of the program in the QRPLOBJ library.

## Restrictions:

- You must have use (\*USE) and add (\*ADD) authorities to the library of the bound program.
- You must have \*USE, object management (\*OBJMGT), and object existence (\*OBJEXIST) authorities to the bound program.
- You must be the owner of the bound program, or a member of a group who is the owner of the bound program, or be a user with all object (\*ALLOBJ) special authority.
- You must have \*USE authority to the following:
  - \*MODULE objects specified on the **Module (MODULE)** parameter, and execute (\*EXECUTE) authority to the library that the module resides in.
  - \*SRVPGM objects specified on the **Bind service program (BNDSRVPGM)** parameter.
  - \*BNDDIR objects specified on the **Binding directory (BNDDIR)** parameter, and \*EXECUTE authority to the library, and all objects used to resolve external symbols for these \*BNDDIR objects, and their libraries.

Top

## Parameters

Keyword	Description	Choices	Notes
PGM	Program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *USRLIBL, *CURLIB</i>	
MODULE	Module	Single values: *NONE Other values (up to 300 repetitions): <i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Module	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	
RPLLIB	Replacement library	Single values: *ONLY, *FIRST, *MODULE Other values: <i>Qualifier list</i>	Optional
	Qualifier 1: Replacement library	<i>Name</i>	

Keyword	Description	Choices	Notes
BNDSRVPGM	Bind service program	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Qualified object name</i>	Optional
	Qualifier 1: Bind service program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
SRVPGMLIB	Bound *SRVPGM library name	Single values: <b>*SAME</b> , <b>*LIBL</b> Other values: <i>Qualifier list</i>	Optional
	Qualifier 1: Bound *SRVPGM library name	<i>Name</i>	
BNDDIR	Binding directory	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Qualified object name</i>	Optional
	Qualifier 1: Binding directory	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	
ACTGRP	Activation group	<i>Name, *SAME</i>	Optional
OPTION	Creation options	Values (up to 6 repetitions): *GEN, *NOGEN, *DUPPROC, *NODUPPROC, *DUPVAR, *NODUPVAR, *WARN, *NOWARN, *TRIM, *NOTRIM, *RSLVREF, *UNRSLVREF	Optional
DETAIL	Listing detail	<b>*NONE</b> , *BASIC, *EXTENDED, *FULL	Optional

Top

---

## Program (PGM)

Specifies the bound program that is to be updated.

This is a required parameter.

### Qualifier 1: Program

*name* Specify the name of the bound program that is to be updated.

### Qualifier 2: Library

#### \*USRLIBL

Only the libraries in the user portion of the job's library list are searched.

#### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library where the bound program is located.

Top

---

## Module (MODULE)

Specifies the names of the existing \*MODULE objects that are to replace the modules of the same name in the bound program. If two or more modules of the bound program have the same name, the **Replacement library (RPLLIB)** parameter indicates which is to be replaced.

If the library of the module being replaced is different from the library of the replacing module, the module's library after the update will remain the library the module was in when the program was first



created. If the RPLLIB parameter is required to determine which module to replace, the value to be entered in the RPLLIB parameter for this module will not change due to the update. Up to 300 names can be specified.

This is a required parameter.

### Single values

#### **\*NONE**

No modules are specified.

### Qualifier 1: Module

**\*ALL** All modules of the same name, to which the user has authority, replace the modules of the bound program.

#### *generic-name*

Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound program.

*name* Specify the name of the module that replaces a module of the bound program.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

#### **\*USRLIBL**

Only the libraries in the user portion of the job's library list are searched.

*name* Specify the name of the library to be searched.

Specify the name of the module that replaces a module of the bound program.

Top

---

## Replacement library (RPLLIB)

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

#### **\*ONLY**

The bound program contains only one module of the specified name and it is replaced. If two or more modules of the bound program have the specified name, an exception is signaled and the bound program is not updated.

#### **\*FIRST**

The first module of the specified name in the module list of the bound program is replaced.

#### **\*MODULE**

The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

*name* Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

---

## Bind service program (BNDSRVPGM)

Specifies the service program to examine for exports if import requests to resolve external symbols cannot be met by the modules and service programs of the updated bound program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound program. Up to 300 names can be specified.

### Single values

#### \*NONE

No service programs are examined during symbol resolution.

### Qualifier 1: Bind service program

**\*ALL** All service programs are examined during symbol resolution.

#### *generic-name*

Specify the generic name of the service programs to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

*name* Specify the name of the service program to examine during symbol resolution.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library to be searched. QTEMP is not a valid library name for this parameter.

---

## Bound \*SRVPGM library name (SRVPGMLIB)

Specifies the library name used to resolve to currently bound service programs. A value other than \*SAME for this parameter can be specified if the program attribute ALWLIBUPD is \*YES.

#### \*SAME

Use the library name where the service program (\*SRVPGM) is currently bound from.

**\*LIBL** All libraries in the job's library list are searched until the first match is found for each bound \*SRVPGM. The first occurrence of a \*SRVPGM is used to resolve to currently bound service programs and \*LIBL is saved to be used at run time. If no match is found in the job's library list, the \*SRVPGM currently bound to the program is used. You must have \*USE authority to the \*SRVPGM objects in the library specified and \*EXECUTE authority to the library itself.

**Note:** The service programs that came from the implicit binding directories (system-supplied service programs) are not changed.

*name* Specify the name of the library to be used first to resolve to all currently bound service programs. If a bound \*SRVPGM does exist in the library specified on SRVPGMLIB parameter, that \*SRVPGM from that library is used instead of the currently bound \*SRVPGM and the library name specified on the SRVPGMLIB parameter is saved to be used at run time. If a \*SRVPGM does not exist in the library specified on the SRVPGMLIB parameter, the \*SRVPGM currently bound to the program is used. You must have \*USE authority to the \*SRVPGM objects in the library specified and \*EXECUTE authority to the library itself.

---

## Binding directory (BNDDIR)

Specifies the binding directory to examine for exports if import requests to resolve external symbols cannot be met by (1) the modules and service programs of the updated bound program or by (2) the service program specified on the BNDSRVPGM parameter. If a module or service program listed in the specified binding directory can resolve external symbols, it is added to the modules or service programs that are bound to the bound program. Up to 300 names can be specified.

### Single values

#### \*NONE

No binding directories are specified.

### Qualifier 1: Binding directory

*name* Specify the name of the binding directory to be used during symbol resolution.

### Qualifier 2: Library

\*LIBL All libraries in the library list for the current thread are searched until the first match is found.

#### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

#### \*USRLIBL

Only the libraries in the user portion of the job's library list are searched.

*name* Specify the name of the library to be searched.

---

## Activation group (ACTGRP)

Specifies the activation group to be used for the updated program.

#### \*SAME

The activation group is not changed. Specify this value if the program was given \*CALLER or \*NEW activation group at the time it was created.

*name* Specify the name of the activation group that is associated with this called program. If the program was given a named activation group at the time it was created, the name of that activation group can be changed to another named activation group.

**Note:** Changing the activation group name can affect the behavior of the program (or service program). Refer to the ILE Concepts book, SC41-5606 for detailed information on the behavior of named activation groups.

---

## Creation options (OPTION)

Specifies the options to be used when the bound program is updated.

You can specify up to 6 values for this parameter.

### Creating Program Objects

**\*GEN** An updated program object is created.

**\*NOGEN**  
An updated program object is not created.

### Duplicate Procedure Names

**\*DUPPROC**  
During symbol resolution, the procedures that are exported from the modules and service programs need not be unique. The first procedure of the specified module and service programs that matches the import request is exported.

**\*NODUPPROC**  
During symbol resolution, each procedure that is exported from the modules and service programs must be unique.

### Duplicate Variable Names

**\*DUPVAR**  
During symbol resolution, the variables that are exported from the modules and service programs need not be unique. The first variable of the specified modules and service programs that matches the import request is exported.

**\*NODUPVAR**  
During symbol resolution, each variable that is exported from the modules and service programs must be unique.

### Issuing Diagnostic Messages

**\*WARN**  
The appropriate diagnostic messages are signaled. Also, if you specify duplicate procedures or variables (\*DUPPROC or \*DUPVAR) and duplicates are found, a diagnostic message is issued indicating what duplicates were found.

**\*NOWARN**  
No information or diagnostic messages are issued.

### Trimming Marooned Module

A **marooned module** is a module of the bound program being updated. This module was originally bound into the bound program from a binding directory to resolve one or more imports. Imports are not resolved for this program update.

**\*NOTRIM**  
Marooned modules are not removed from the bound program.

**Note:** Programs may grow significantly over time when \*NOTRIM is specified.

**\*TRIM**  
Marooned modules are removed from the bound program.

**Note:** If marooned modules are removed from the bound program for this program update, the exports that the modules contain are not available for other program updates.

## Resolving References (Imports)

### \*RSLVREF

All imports must be resolved to exports for the bound program to be updated.

### \*UNRSLVREF

All imports do not need to resolve to exports for the bound program to be updated.

**Note:** If this command contains an import that does not resolve, an exception will be generated when the command is run.

Top

---

## Listing detail (DETAIL)

Specifies the level of detail of the binder listing to be printed. The printer file \*LIBL/QSYSPRT is used to create the listing.

### \*NONE

No binder listing is printed.

### \*BASIC

The brief summary table, the options passed to this command, and some processing statistics are printed.

### \*EXTENDED

The extended summary table and the binding information listing are printed, in addition to the information provided in the \*BASIC listing (the brief summary table, the options passed to this command, and some processing statistics).

**\*FULL** The cross-reference listing is printed, in addition to the information provided in the \*EXTENDED listing (the extended summary table, the binding information listing, the brief summary table, the options passed to this command, and some processing statistics).

Top

---

## Examples

```
UPDPGM PGM(STAR) MODULE(SKY/NOVA) RPLLIB(*FIRST)
```

This command replaces the first module named NOVA existing in the program object STAR with the module NOVA in the library SKY.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF223D

Not authorized to update &1 in &2 type \*&3.

#### CPF223E

Authority check for use adopted authority attribute failed.

#### CPF5CA7

SRVPGMLIB must be \*SAME when ALWLIBUPD is \*NO.

#### CPF5CE0

Program &1 not updated.

**CPF5CE2**

Unexpected error occurred during program or service program update.

**CPF5D1B**

Update of program &1 in library &2 not allowed.

[Top](#)

# Update Service Program (UPDSRVPGM)

Where allowed to run: All environments (\*ALL)  
 Threadsafes: No

Parameters  
 Examples  
 Error messages

The Update Service Program (UPDSRVPGM) command can be used to replace modules of an Integrated Language Environment (ILE) bound service program with other modules on the system, without requiring you to change or recompile the bound service program. Modules being replaced must be module objects (\*MODULE) on the system.

Other jobs running the bound service program can run while the service program is being updated with this command. The currently running service program is moved to library QRPLIB and an updated version of the service program will be inserted into the library of the service program. Current activations of the service program will continue running using the version of the service program in the QRPLIB library.

## Restrictions:

- You must have use (\*USE) and add (\*ADD) authorities to the library of the service program.
- You must have \*USE, object management (\*OBJMGT), and object existence (\*OBJEXIST) authorities to the service program.
- You must be the owner of the service program, or a member of a group who is the owner of the service program, or be a user with all object (\*ALLOBJ) special authority.
- You must have \*USE authority to the following:
  - \*MODULE objects specified on the **Module (MODULE)** parameter, and execute (\*EXECUTE) authority to the library that the module resides in.
  - \*SRVPGM objects specified on the **Bind service program (BNDSRVPGM)** parameter.
  - \*BNDDIR objects specified on the **Binding directory (BNDDIR)** parameter, and \*EXECUTE authority to the library, and all objects used to resolve external symbols for these \*BNDDIR objects, and their libraries.
  - Object operational (\*OBJOPR) and read (\*READ) authorities to the file specified for the **Export source file (SRCFILE)** parameter.

Top

## Parameters

Keyword	Description	Choices	Notes
SRVPGM	Service program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Service program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *USRLIBL, *CURLIB</i>	
MODULE	Module	Single values: *NONE Other values (up to 300 repetitions): <i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Module	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	
EXPORT	Export	<i>*CURRENT, *SRCFILE, *ALL</i>	Optional

Keyword	Description	Choices	Notes
SRCFILE	Export source file	<i>Qualified object name</i>	Optional
	Qualifier 1: Export source file	<i>Name, <u>QSRVSRC</u></i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Export source member	<i>Name, *SRVPGM</i>	Optional
RPLLIB	Replacement library	Single values: <b>*ONLY</b> , <b>*FIRST</b> , <b>*MODULE</b> Other values: <i>Qualifier list</i>	Optional
	Qualifier 1: Replacement library	<i>Name</i>	
BNDSRVPGM	Bind service program	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Qualified object name</i>	Optional
	Qualifier 1: Bind service program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
SRVPGMLIB	Bound *SRVPGM library name	Single values: <b>*SAME</b> , <b>*LIBL</b> Other values: <i>Qualifier list</i>	Optional
	Qualifier 1: Bound *SRVPGM library name	<i>Name</i>	
BNDDIR	Binding directory	Single values: <b>*NONE</b> Other values (up to 300 repetitions): <i>Qualified object name</i>	Optional
	Qualifier 1: Binding directory	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	
ACTGRP	Activation group	<i>Name, *SAME</i>	Optional
OPTION	Creation options	Values (up to 6 repetitions): <b>*GEN</b> , <b>*NOGEN</b> , <b>*DUPPROC</b> , <b>*NODUPPROC</b> , <b>*DUPVAR</b> , <b>*NODUPVAR</b> , <b>*WARN</b> , <b>*NOWARN</b> , <b>*TRIM</b> , <b>*NOTRIM</b> , <b>*RSLVREF</b> , <b>*UNRSLVREF</b>	Optional
DETAIL	Listing detail	<b>*NONE</b> , <b>*BASIC</b> , <b>*EXTENDED</b> , <b>*FULL</b>	Optional

Top

---

## Service program (SRVPGM)

Specifies the service program that is to be updated.

This is a required parameter.

### Qualifier 1: Service program

*name* Specify the name of the bound service program that is to be updated.

### Qualifier 2: Library

#### \*USRLIBL

Only the libraries in the user portion of the job's library list are searched.

#### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library where the bound service program is located.

Top



---

## Module (MODULE)

Specifies the existing \*MODULE objects that are to replace the modules of the same name in the bound program. Up to 300 names can be specified. If two or more modules of the bound program have the same name, the **Replacement library (RPLLIB)** parameter indicates which is to be replaced.

If the library of the module being replaced is different from the library of the replacing module, the module's library after the update will remain the library the module was in when the service program was first created. If the RPLLIB parameter is required to determine which module to replace, the value to be entered in the RPLLIB parameter for this module will not change due to the update.

This is a required parameter.

### Single values

#### \*NONE

No modules are specified.

**Note:** This value can be specified when the module is not changing, but you are updating the service program (the BNDSRVPGM parameter) or the binding directory (BNDDIR parameter) to examine for exports. The existing module is used.

### Qualifier 1: Module

**\*ALL** All modules of the same name to which the user has authority replace the modules of the bound service program.

#### *generic-name*

Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound service program.

*name* Specify the name of the module that replaces a module of the bound service program.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### \*CURLIB

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

#### \*USRLIBL

Only the libraries in the user portion of the job's library list are searched.

*name* Specify the name of the library to be searched.

Top

---

## Export (EXPORT)

Specifies the variables and procedures that are to be exported from the updated service program. This parameter also specifies whether new signatures identifying the sequence of exports in the service program are created.

#### \*CURRENT

The variables, procedures, and signatures currently exported from the service program continue to be exported. No new signatures are created.

**Note:** If a variable or procedure that is currently exported is not available for export after the update, an exception is signaled and the service program is not updated.

**\*SRCFILE**

The source file member identified by the **Source file (SRCFILE)** and **Source member (SRCMBR)** parameters contains EXPORT statements that identify the data and procedures to export from the service program. If the specified source file differs from the one used to create the service program, a new signature or set of signatures may be created.

**Note:** If a signature is lost, some current clients of the service program may not be able to use the service program without binding again.

**\*ALL** All variables and procedures exported from the modules of the service program are exported from the updated service program.

If the number or names of the variables and procedures exported before the service program update differs from those exported after the service program update, a new signature is created.

**Note:** If a new signature is created, all clients of the service program must be bound again before they can use the service program.

Top

---

## Export source file (SRCFILE)

Specifies the source file containing the specifications for exporting variables and procedures from this bound service program.

### Qualifier 1: Export source file

#### QSRVSRC

The source file name is QSRVSRC.

*name* Specify the name of the source file containing the specifications for exporting variables and procedures.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library to be searched.

Top

---

## Export source member (SRCMBR)

Specifies the member in the file specified for the **Export source file (SRCFILE)** parameter that contains the specifications for exporting variables and procedures from this bound service program.

#### \*SRVPGM

The source file member has the same name as the service program being updated.

*name* Specify the name of the member that contains the specifications for exporting.

Top

---

## Replacement library (RPLLIB)

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

### \*ONLY

The bound service program contains only one module of the specified name and it is replaced. If two or more modules of the bound service program have the specified name, an exception is signaled and the bound service program is not updated.

### \*FIRST

The first module of the specified name in the module list of the bound service program is replaced.

### \*MODULE

The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

*name* Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

Top

---

## Bind service program (BNDSRVPGM)

Specifies the service program to examine for exports if import requests to resolve external symbols cannot be met by the modules and service programs of the updated bound service program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound service program. Up to 300 names can be specified.

### Single values

#### \*NONE

No service programs, except those in the bound service program being updated, are examined during symbol resolution.

### Qualifier 1: Bind service program

**\*ALL** All service programs are examined during symbol resolution.

#### *generic-name*

Specify the generic name of the service programs to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

*name* Specify the name of the service program to examine during symbol resolution.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

*name* Specify the name of the library to be searched. QTEMP is not a valid library name for this parameter.

Top

---

## Bound \*SRVPGM library name (SRVPGMLIB)

Specifies the library name used to resolve to currently bound service programs. A value other than \*SAME for this parameter can be specified if the program attribute ALWLIBUPD is \*YES.

### \*SAME

Use the library name where the service program (\*SRVPGM) is currently bound from.

**\*LIBL** All libraries in the job's library list are searched until the first match is found for each bound \*SRVPGM. The first occurrence of a \*SRVPGM is used to resolve to currently bound service programs and \*LIBL is saved to be used at run time. If no match is found in the job's library list, the \*SRVPGM currently bound to the program is used. You must have \*USE authority to the \*SRVPGM objects in the library specified and \*EXECUTE authority to the library itself.

**Note:** The service programs that came from the implicit binding directories (system-supplied service programs) are not changed.

*name* Specify the name of the library to be used first to resolve to all currently bound service programs. If a bound \*SRVPGM does exist in the library specified on SRVPGMLIB parameter, that \*SRVPGM from that library is used instead of the currently bound \*SRVPGM and the library name specified on the SRVPGMLIB parameter is saved to be used at run time. If a \*SRVPGM does not exist in the library specified on the SRVPGMLIB parameter, the \*SRVPGM currently bound to the program is used. You must have \*USE authority to the \*SRVPGM objects in the library specified and \*EXECUTE authority to the library itself.

Top

---

## Binding directory (BNDDIR)

Specifies the binding directory to examine for exports if import requests to resolve external symbols cannot be met by (1) the modules and service programs of the updated bound program or by (2) the service program specified on the BNDSRVPGM parameter. If a module or service program listed in the specified binding directory can resolve external symbols, it is added to the modules or service programs that are bound to the bound service program. Up to 300 names can be specified.

### Single values

#### \*NONE

No binding directories, except those in the bound service program being updated, are examined during symbol resolution.

### Qualifier 1: Binding directory

*name* Specify the name of the binding directory to be used during symbol resolution.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

#### **\*CURLIB**

The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

#### **\*USRLIBL**

Only the libraries in the user portion of the job's library list are searched.

*name* Specify the name of the library to be searched.

Top

---

## Activation group (ACTGRP)

Specifies the activation group to be used for the updated service program.

### \*SAME

The activation group is not changed. Specify this value if the service program was given \*CALLER activation group at the time it was created.

*name* Specify the name of the activation group that is associated with this called service program. If the service program was given a named activation group at the time it was created, the name of that activation group can be changed to another named activation group.

**Note:** Changing the activation group name can affect the behavior of the program (or service program). Refer to the ILE Concepts book, SC41-5606 for detailed information on the behavior of named activation groups.

Top

---

## Creation options (OPTION)

Specifies the options to be used when the service program object is updated.

You can specify up to 6 values for this parameter.

### Creating Program Objects

\*GEN An updated program object is created.

### \*NOGEN

An updated program object is not created.

### Duplicate Procedure Names

#### \*DUPPROC

During symbol resolution, the procedures that are exported from the modules and service programs need not be unique. The first procedure of the specified modules and service programs that matches the import request is exported.

#### \*NODUPPROC

During symbol resolution, each procedure that is exported from the modules and service programs must be unique.

### Duplicate Variable Names

#### \*DUPVAR

During symbol resolution, the variables that are exported from the modules and service programs need not be unique. The first variable of the specified modules and programs that matches the import request is exported.

#### \*NODUPVAR

During symbol resolution, each variable that is exported from the modules and service programs must be unique.

### Issuing Diagnostic Messages

#### \*WARN

The appropriate diagnostic messages are signaled. Also, if you specify duplicate procedures or variables (\*DUPPROC or \*DUPVAR) and duplicates are found, a diagnostic message is issued indicating what duplicates were found.

#### **\*NOWARN**

No information or diagnostic messages are issued.

### **Trimming Marooned Modules**

A **marooned module** is a module of the bound program being updated. This module was originally bound into the bound program from a binding directory to resolve one or more imports. Imports are not resolved for this program update.

#### **\*NOTRIM**

Marooned modules are not removed from the bound program.

**Note:** Bound service programs may grow significantly over time when \*NOTRIM is specified.

#### **\*TRIM**

Marooned modules are removed from the bound program.

**Note:** If marooned modules are removed from the bound program during this program update, the exports that the modules contain are not available for other program updates.

### **Resolving References (Imports)**

#### **\*RSLVREF**

All imports must resolve to exports for the bound service program to be updated.

#### **\*UNRSLVREF**

All imports do not need to resolve to exports for the bound service program to be updated.

**Note:** If this command contains an import that does not resolve, an exception will be generated when the command is run.

Top

---

## **Listing detail (DETAIL)**

Specifies the level of detail of the binder listing to be printed. The printer file \*LIBL/QSYSPRT is used to create the listing.

#### **\*NONE**

No binder listing is printed.

#### **\*BASIC**

The brief summary table, the options passed to this command, and some processing statistics are printed.

#### **\*EXTENDED**

The extended summary table and the binding information listing are printed, in addition to the information provided in the \*BASIC listing (the brief summary table, the options passed to this command, and some processing statistics).

**\*FULL** The cross-reference listing is printed, in addition to the information provided in the \*EXTENDED listing (the extended summary table, the binding information listing, the brief summary table, the options passed to this command, and some processing statistics).

Top

---

## **Examples**

```
UPDSRVPGM  SRVPGM(WORKDOC)  MODULE(BIN/TASKTWO)
           RPLLIB(*MODULE)
```

This command replaces the module named TASKTWO in the service program object named WORKDOC with another module named TASKTWO in the library BIN, only if the module being replaced was originally from the library BIN.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF223D

Not authorized to update &1 in &2 type \*&3.

#### CPF223E

Authority check for use adopted authority attribute failed.

#### CPF5CA7

SRVPGMLIB must be \*SAME when ALWLIBUPD is \*NO.

#### CPF5CE1

Service program &1 not updated.

#### CPF5CE2

Unexpected error occurred during program or service program update.

#### CPF5D1C

Update of service program &1 in library &2 not allowed.

Top





---

# Update System Information (UPDSYSINF)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Update System Information (UPDSYSINF) command updates various system information that was gathered using the Retrieve System Information (RTVSYISINF) command. This command is used to restore full customization of your system.

The following types of information can be updated:

- Edit descriptions
- Network attributes
- Reply list entries
- Service attributes
- Service providers
- System values

**Note:** Service attributes and service providers will only be restored if the the source system was at V3R0M5 or later.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i>	Required, Positional 1
TYPE	Type of information	*ALL, *EDTD, *NETA, *RPYLE, *SRVATR, *SRVPVD, *SYSVAL	Optional, Positional 2

Top

---

## Library (LIB)

Specifies the library where the system information was placed by the Retrieve System Information (RTVSYISINF) command.

This is a required parameter.

The possible values are:

*library-name*

Specify the library name where the system information exists.

Top

---

## Type of information (TYPE)

Specifies the type of system information to be updated.

This is a required parameter.

The possible values are:

- \*ALL** Update all information saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*EDTD** Update all edit descriptions saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*NETA** Update all network attributes saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*RPYLE** Update all reply list entries saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*SRVATR** Update all service attributes saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*SRVPVD** Update all service providers saved on the source system by the Retrieve System Information (RTVSYISINF) command.
- \*SYSVAL** Update all system values saved on the source system by the Retrieve System Information (RTVSYISINF) command.

Top

---

## Examples

```
UPDSYISINF LIB(TEST) TYPE(*ALL)
```

This command will update all saved system information on the current system from the information in library TEST.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFA976

Error occurred updating system information for type &1.

Top

---

## Verify APPC Connection (VFYAPPCNN)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Verify APPC Connection (VFYAPPCNN) command, also known as APING, exchanges data packets between the local location and the specified remote location using Advanced Program-to-Program Communications, and measures the round-trip time of each data packet exchange iteration.

For this function to work, the remote location specified must be running the target portion of this function, APINGD (APING daemon).

Top

---

### Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	<i>Character value</i>	Required, Positional 1
MODE	Mode	<i>Communications name, *NETATR</i>	Optional, Positional 2
RMTUSER	Remote user ID	<i>Character value, *NONE, *CURRENT</i>	Optional
RMPWD	Remote password	<i>Character value, *NONE</i>	Optional
MSGMODE	Message mode	<i>*VERBOSE, *QUIET</i>	Optional
PKTLEN	Packet length (in bytes)	0-32763, <u>100</u>	Optional
NBRITER	Number of iterations	1-32767, <u>2</u>	Optional
NBRPKT	Number of packets	1-32767, <u>1</u>	Optional
ECHO	Echo	<i>*YES, *NO</i>	Optional
WAITTIME	Wait time (in seconds)	2-3600, <u>10</u> , *NOMAX, *NOWAIT	Optional

Top

---

### Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name. If only the remote location name is specified, the local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID).

Top

---

### Mode (MODE)

Specifies the name of the mode to be used for the APPC conversation.

The possible values are:

### **\*NETATR**

The mode in the network attributes is used.

### ***mode-name***

Specify a mode name. Specify BLANK for a mode name consisting of eight blank characters.

**Note:** SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

---

## **Remote user ID (RMTUSER)**

Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMTPWD(\*NONE) is not valid.

The possible values are:

### **\*NONE**

No user ID is sent. If security on the target system is configured to require a user ID, the command will fail.

### **\*CURRENT**

The user ID of the job (signed-on user) using this command is sent.

### ***remote-user-identifier***

Specify a user ID to use that exists on the target system. If a user ID is specified and password security is active on the target system, a password must be specified.

Top

---

## **Remote password (RMTPWD)**

Specifies the password sent to the target system.

The possible values are:

### **\*NONE**

The system does not send a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, the command will fail.

### ***password***

Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. The password may or may not be substituted across the communication line depending on whether the remote system supports password substitution.

Top

---

## **Message mode (MSGMODE)**

Specifies the amount of information displayed by the command.

The possible values are:

### **\*VERBOSE**

Display verification message after each iteration.

### **\*QUIET**

Display only initial and summary messages.

---

## Packet length (in bytes) (PKTLEN)

Specifies the length (in bytes) of the packets that are exchanged between the local and remote systems.

The possible values are:

**100** The packet length is 100 bytes.

### *packet-length*

Specify the length of the packet. Valid values range from 0 through 32763 bytes.

Top

---

## Number of iterations (NBRITER)

Specifies the number of iterations. For each iteration, the specified number of data packets are exchanged between the local and remote systems.

The possible values are:

**2** Two iterations are performed.

### *number-of-iterations*

Specify the number of iterations. Valid values range from 1 through 32767.

Top

---

## Number of packets (NBRPKT)

Specifies the number of packets that are sent by the local system for each iteration before giving the target system permission to send.

The possible values are:

**1** One packet is sent for each iteration.

### *number-of-packets*

Specify the number of packets that are sent for each iteration. Valid values range from 1 through 32767.

Top

---

## Echo (ECHO)

Whether the remote location should echo packets back to the local location.

The possible values are:

**\*YES** Packets are echoed back from the remote location to the local location.

**\*NO** Packets are sent from the local location to the remote location only; packets are not echoed back to the local location.

Top

---

## Wait time (in seconds) (WAITTIME)

Specifies the time in seconds to wait for the return (echo) before declaring the remote location to be unreachable.

The possible values are:

**10**      The system waits 10 seconds.

**\*NOMAX**  
The system waits forever.

**\*NOWAIT**  
The system returns immediately if there is a connection ready and available.

*time-to-wait-for-reply*  
Specify the length of time in seconds. Valid values range from 2 through 3600 seconds.

Top

---

## Examples

### Example 1: Verify an APPC Connection

```
VFYAPPCNN  RMTLOCNAME(RPCNET.CHICAGO)  NBRITER(3)  NBRPKT(4)
           PKTLEN(500)
```

This command exchanges four 500-byte packets in each of three iterations with remote location CHICAGO, network identifier RPCNET. The default mode used is taken from network attribute DFTMODE. Since the default MSGMODE(\*VERBOSE) was taken, each iteration will result in an informational message in the job log indicating the elapsed time for the iteration.

### Example 2: Using APING Alias Command

```
APING  RMTLOCNAME(RPCNET.CHICAGO)  NBRITER(3)  NBRPKT(4)  PKTLEN(500)
```

This command is equivalent to the command in Example 1 above.

### Example 3: Using APING with a Wait Time

```
APING  RMTLOCNAME(RPCNET.CHICAGO)  WAITTIME(20)
```

This command will verify the connection with remote location CHICAGO, network identifier RPCNET. The maximum time to wait for a response from the remote location is 20 seconds.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF91CC**  
Command did not complete successfully.

Top

---

## Verify Communications (VFYCMN)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Communications (VFYCMN) command shows the Select a Line to Test display, which can be used to verify that communications equipment is operating properly.

Depending on the user's system configuration, the following tests can be run:

- Link
- Local modem
- Remote modem
- Cable
- Communications input/output adapter
- Link Problem Determination Aid-2 (LPDA-2)

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
VFYTYPE	Verification type	*REMOTE, <u>*LOCAL</u>	Optional, Positional 1
RCPNAME	Remote control point	<i>Name</i>	Optional
NETID	Network ID	<i>Name</i> , <u>*NETATR</u>	Optional
USERID	User ID	<i>Name</i>	Optional
PASSWORD	Password	<i>Character value</i> , <u>X''</u> , *NONE	Optional

Top

---

### Verification type (VFYTYPE)

Specifies the type of verification. You may verify that local or remote communications hardware is operating correctly.

**Note:** You cannot do remote analysis if the SystemView System Manager/400 program is not installed.

This is a required parameter.

The possible values are:

\*LOCAL

Communications hardware is checked to verify that it is operating correctly on this AS/400.

### **\*REMOTE**

Communications hardware is checked to verify that it is operating correctly on another AS/400 system that is enrolled as a service requester.

Top

---

## **Remote control point (RCPNAME)**

Specifies the remote control point name for the service requester system where the remote verification is done.

**Note:** This parameter is valid only if \*REMOTE is specified for the **Verification type** prompt (VFYTYPE parameter).

Top

---

## **Network ID (NETID)**

Specifies the network identifier (ID) for the service requester system where the remote verification is done.

**Note:** This parameter is valid only if \*REMOTE is specified for the **Verification type** prompt (VFYTYPE parameter).

The possible values are:

### **\*NETATR**

The network ID of the service provider is used.

### *network-ID*

Specify the network ID.

Top

---

## **User ID (USERID)**

Specifies the user identifier (ID) used to access the remote system.

**Note:** This parameter is valid only if \*REMOTE is specified for the **Verification type** prompt (VFYTYPE parameter).

Top

---

## **Password (PASSWORD)**

Specifies the password used to access the remote system.

**Note:** This parameter is valid only if \*REMOTE is specified for the **Verification type** prompt (VFYTYPE parameter).

The possible values are:

### **\*NONE**

No password is needed to access the remote system because the remote system has a security level of 10.



*password*

Specify the password.

Top

---

## Examples

### Example 1: Show Select a Line to Test Panel

```
VFYCMN
```

This command shows the Select a Line to Test panel.

### Example 2: Checking a Remote System

```
VFYCMN VFYTYPE(*REMOTE)
```

This command shows the panel which prompts for the remaining values of the command. After you specify the appropriate values, remote analysis begins.

### Example 3: Accessing a Remote System Using a Password

```
VFYCMN VFYTYPE(*REMOTE) RCPNAME(RCH38377) USERID(JON)  
PASSWORD(ABC123)
```

This command shows the display which prompts for the remaining values of the command. After you specify the appropriate values beyond the ones specified on the command example, remote analysis begins.

### Example 4: Accessing a Remote System Without a Password

```
VFYCMN VFYTYPE(*REMOTE) RCPNAME(RCH38377) USERID(JON)
```

This command is similar to the preceding example except that the PASSWORD parameter is not specified. The same prompt display is shown, however, the system assumes that the remote system has a security level of 10, that is, it does not use passwords. Another prompt display appears after this command is specified. After the user specifies the appropriate values on this display, remote analysis begins.

### Example 5: Checking a Local System

```
VFYCMN VFYTYPE(*LOCAL)
```

This command begins analysis on the local device. The remaining parameters do not appear on the display.

Top

---

## Error messages

### \*ESCAPE Messages

```
CPF2B3C
```

Licensed program &1 not installed.

Top



---

## Verify Image Catalog (VFYIMGCLG)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Image Catalog (VFYIMGCLG) command is used to verify the images in an image catalog based on the value specified in the TYPE parameter. The user can optionally sort the images in install sequence based also on the TYPE parameter.

A status message will be issued upon successful completion of the command. If the VFYIMGCLG command fails, the Work with Image Catalog Entries (WRKIMGCLGE) command can be used to look at the images and the status of each. The VFYIMGCLG command is intended for verifying images for a complete software upgrade, installation of a PTF, or other types of installs.

### Restrictions:

- You must have security administrator (\*SECADM) and all object (\*ALLOBJ) special authorities to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
TYPE	Image catalog type	*UPGRADE, *PTF, *OTHER	Optional
SORT	Sort image catalog	*NO, *YES	Optional

Top

---

## Image catalog (IMGCLG)

Specifies the image catalog to be verified.

This is a required parameter.

*name* Specify the name of the image catalog.

Top

---

## Image catalog type (TYPE)

Specifies the type of image catalog to be verified.

### \*UPGRADE

The image catalog to be verified is for a complete software upgrade. The system will verify that the necessary images for a software upgrade exist and can be loaded into the virtual optical device. The following list shows the sort order and required images for verify type \*UPGRADE:

1. Licensed Internal Code for OS/400 (Required)

2. Operating System/400 (Required)
3. OS/400 - Library QGPL (Required)
4. OS/400 - Library QUSRSYS (Required)
5. Operating System/400 no-charge options
6. Operating System/400 no-charge (bonus) licensed programs and options
7. Operating System/400 Keyed set products
8. Single products
9. Secondary languages
10. Program temporary fixes

**\*PTF** The image catalog to be verified is for a PTF install. The system will verify all PTF volume sets are complete and can be loaded into the virtual optical device. All non-PTF volumes will be unloaded.

**\*OTHER**

The image catalog to be verified is not for a specific type of install. This option will load the images from the image catalog in the order they exist. There will be no verification or sorting of images.

Top

---

## Sort image catalog (SORT)

Specifies whether the images of this type should be sorted in the order required for a software upgrade or PTF install. If TYPE(\*OTHER) is specified, the images in the image catalog are not sorted.

**\*NO** The images in the image catalog are not sorted based on the value specified for the TYPE parameter.

**\*YES** The images in the image catalog are sorted based on the value specified for the TYPE parameter.

Top

---

## Examples

### Example 1: Verify Image Catalog for TYPE(\*UPGRADE)

```
VFYIMGCLG IMGCLG(MYCLG) TYPE(*UPGRADE) SORT(*YES)
```

This command verifies that the install image catalog MYCLG contains the necessary files for a software upgrade. If the necessary media files exist, the images will be sorted in the order required for a software upgrade.

### Example 2: Verify Image Catalog for TYPE(\*PTF)

```
VFYIMGCLG IMGCLG(MYCLG) TYPE(*PTF)
```

This command verifies that all required cumulative PTF volumes in image catalog MYCLG are available. No sorting of the images will occur.

Top

---

## Error messages

### \*ESCAPE Messages

**CPFBC20**

Verification for image catalog &1 failed.

**CPFBC45**

Image catalog &1 not found.

[Top](#)



---

## Verify Link supporting LPDA-2 (VFYLNKLPDA)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Verify Link Supporting LPDA-2 (VFYLNKLPDA) command allows you to run any of the LPDA-2 tests and to receive the results in a format you specify.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, or QSRVBAS user profiles have private authorities to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LINE	Line	<i>Name</i>	Required, Positional 1
TEST	Test	*DCELINSTS, *DCELINTST, *ANZLIN, *SNDRCV	Optional
LCLDCEADR	Local DCE address	X'01'-X'FB', *LCL	Optional
RMTDCEADR	Remote DCE address	X'01'-X'FB', *ANY	Optional
OUTPUT	Output	_, *PRINT	Optional
SEQCOUNT	Number of sequences	1-3, <u>1</u>	Optional
DTEPORT	Remote DTE port	<u>A</u> , B, C, D	Optional
DTERTY	DTE retry	*NO, *YES	Optional
DCERTY	DCE retry	*NO, *YES	Optional
VRYNKSTS	Link status after test	*SAME, *ON, *OFF	Optional

Top

---

### Line (LINE)

Specifies the name of the line (nonswitched \*SDLC) on the link to be tested.

This is a required parameter.

Top

---

### Test (TEST)

Specifies which of the four tests is to be run.

The possible values are:

#### \*DCELINSTS

Line status of the data circuit-terminating equipment (DCE) is returned.

#### \*DCELINTST

Line testing of the data circuit-terminating equipment (DCE) is done.

**\*ANZLIN**

The analyze line test is done. This test is for analog lines only.

**\*SNDRCV**

The send/receive test is done.

Top

---

## Local DCE address (LCLDCEADR)

Specifies the local data circuit-terminating equipment (DCE) address. By convention, bits 4-7 of this byte indicate the Link Segment Level (LSL) of the local DCE and remote DCE. Bits 0-3 are used to uniquely identify a local DCE among several on the same LSL. The address is set in the local DCE during configuration and should follow this convention.

**Note:** X'00' is not a valid address for a local DCE.

The possible values are:

**\*LCL** X'01', the address for the local DCE on LSL 1, is used.

*local-DCE-address*

Specify the local DCE address. Valid values range from X'01' through X'FB'.

Top

---

## Remote DCE address (RMTDCEADR)

Specifies the remote data circuit-terminating equipment (DCE) address.

This parameter must be specified if you are testing a multipoint link.

The possible values are:

**\*ANY** X'FD', the global remote DCE address is used. If the remote DCE is not idle, it will respond regardless of its previously configured address.

**Note:** Multipoint tributary DCEs will not respond to an address of \*ANY.

*remote-DCE-address*

Specify the remote DCE address. Valid values range from X'01' through X'FB'.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible values are:

**\*** The output is displayed (if requested by an interactive job) or printed with the job's spooled output (if requested by a batch job).

**\*PRINT**

The output is printed with the job's spooled output.



---

## Number of sequences (SEQCOUNT)

Specifies the number of sequences to transmit for the send/receive test. A sequence is a group of 16 blocks, with the block length dependent on the configuration of the DCE.

The possible values are:

1      1 sequence is sent during the test.

*number-of-sequences*

Specify the number of sequences to send. Valid values range from 1 through 3.

---

## Remote DTE port (DTEPORT)

Specifies the data terminal equipment (DTE) port of the remote DCE for which status will be returned. This parameter is valid only when working with line status and line testing of DCEs. This parameter is meaningful only for multiple port DCEs.

The possible values are:

A      Status will be returned for the A-port.

**B**      Status will be returned for the B-port.

**C**      Status will be returned for the C-port.

**D**      Status will be returned for the D-port.

---

## DTE retry (DTERTY)

Specifies that this command is a retry of a link operation from the system DTE to the local DCE due to a bad response or no response received from the local DCE.

The possible values are:

\*NO    This command is not a retry.

**\*YES**    This command is a retry.

---

## DCE retry (DCERTY)

Specifies whether the local DCE should retry the command to the remote DCE if a bad response or no response is received from the remote DCE.

The possible values are:

\*NO    The local DCE should not retry the command.

**\*YES**    The local DCE should retry the command.

---

## Link status after test (VRYLNKSTS)

Specifies the desired status of the link (varied on or varied off) after the test completes.

**Note:** After running a test on a manually switched link, the link should in most cases be left varied on to allow further information to be received on the same connection. If the switched link is varied off, the failing connection will be lost and no further analysis can be done.

The possible values are:

### \*SAME

The specified link is returned to the status it was in immediately prior to testing.

**\*ON** The link remains varied on.

**\*OFF** The link is varied off.

---

## Examples

### Example 1: Checking Line Status

```
VFYLNKLPDA  LINE(LINE1)  DTEPORT(B)
```

This command retrieves the DCE line status from synchronous data link control (SDLC) line, LINE1, and displays the status. The remote DCEs DTE line connection status of port B is returned if the user is verifying a multipoint DCE. An error message will be returned if the remote DTE has only a single port, for example, port A. The default VRYLNKSTS(\*SAME) causes the line named LINE1 to return to the status prior to the test.

### Example 2: Analyzing a Line

```
VFYLNKLPDA  LINE(LINE2)  TEST(*ANZLIN)  OUTPUT(*PRINT)
             LCLDCEADR(02)  VRYLNKSTS(*ON)
```

This command analyzes the SDLC line, LINE2. The second LSL is used; the lower four bits of the local DCE address (LCLDCEADR) are 2. The results are sent to a spooled file. After the test, LINE2 remains varied on to allow for more testing.

### Example 3: Testing Sending and Receiving Capabilities

```
VFYLNKLPDA  LINE(LINE3)  TEST(*SNDRCV)  SEQCOUNT(3)
             RMTDCEADR(21)  DCERTY(*YES)
```

This command tests the sending and receiving capabilities on the multipoint line, LINE3. Three sequences of 16 blocks are sent between the local (control) DCE and the remote (tributary) DCE with the address of X'21'. If the local DCE fails to receive a response on the first attempt, the local DCE will retry this command to the remote DCE.

---

## Error messages

### \*ESCAPE Messages

**CPF1BAF**  
Error occurred while processing VFYLNKLPDA command.

**CPF1BA9**  
Line &1 vary off failed.

**CPF1BCC**  
Test cannot be run at this time.

**CPF1BCD**  
DCE self test failed.

**CPF1BCE**  
Sense byte returned is not valid.

**CPF1BC1**  
Error detected while processing VFYLNKLPDA command.

**CPF1BC3**  
Test cannot run in switched network backup mode.

**CPF1BC4**  
Requested test is not supported.

**CPF1BC5**  
Required feature not installed.

**CPF1BC6**  
Required feature not operational.

**CPF1BC7**  
Test is not compatible with DCE configuration.

**CPF1BC8**  
DTEPORT parameter cannot be specified.

**CPF1BD1**  
Line description &1 is not \*SDLC.

**CPF1BD2**  
System Service Tools is active.

**CPF1BD4**  
Not authorized to line description &1.

**CPF1BD7**  
VFYLNKLPDA command does not support switched lines.

**CPF1B8A**  
Line &1 failed during test.

**CPF1B8B**  
No response received for test request.

**CPF1B8C**  
Test cannot be run on line &1.

**CPF1B8D**  
Error occurred while processing VFYLNKLPDA command.

**CPF1B8E**  
Test cannot be run at this time.

**CPF1B8F**  
Test request failed. Test already active on line.

**CPF1B80**

Line description &1 does not exist.

**CPF1B81**

Error occurred while getting configuration information.

**CPF1B83**

Line &1 is not in proper state for test.

**CPF1B89**

Test cannot be run on line &1.

**CPF1B9F**

Line &1 cannot be varied off at this time.

**CPF1B93**

Line &1 did not vary on.

[Top](#)

---

## Verify NetWare Aut Entry (VFYNTWAUTE)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify NetWare Authentication Entry (VFYNTWAUTE) command verifies the authentication entry for a server. The user name, password, and other data, are sent to the server, where they are used to sign on to the server. This command could be used, for example, to verify that the password is correct before submitting a batch job that uses the server. Use the Work with NetWare Authentication Entries (WRKNTWAUTE) command to view the server authentication entries in a user profile.

**Note:** This command cannot be used for entries that specify PASSWORD(\*STRNTWCNN).

Top

---

### Parameters

Keyword	Description	Choices	Notes
SVRTYPE	Server type	*NDS	Required, Positional 1
NDSTREE	NDS tree	<i>Character value</i>	Optional
SERVER	Server	<i>Character value, *ANY</i>	Optional
USRPRF	User profile	<i>Name, *CURRENT</i>	Optional

Top

---

### Server type (SVRTYPE)

Specifies the server type of the server authentication entry that is to be verified.

**\*NDS** The entry is for a NetWare Directory Services tree.

Top

---

### NDS tree (NDSTREE)

Specifies the NDS tree of the authentication entry to be verified.

*character-value*

Specify the name of the NDS tree.

Top

---

### Server (SERVER)

Specifies the server of the authentication entry to be verified.

**\*ANY** Use any server within the NDS tree.

*character-value*

Specify the name of the server.

Top

---

## User profile (USRPRF)

Specifies the user profile containing the authentication entry.

### \*CURRENT

Use the current user profile.

*name* Specify the name of the user profile. The user profile must be the current user profile, or the user must have \*USE and \*OBJMGT authority to the user profile, and \*SECADM special authority.

Top

---

## Examples

```
VFYNTWAUTE SVRTYPE(*NETWARE3) SERVER(SERVER03)
```

This command verifies the server authentication entry for NetWare 3.x server SERVER03 from the current user profile.

Top

---

## Error messages

### \*ESCAPE Messages

#### FPE021A

Verification of authentication entry failed.

Top

---

## Verify OptiConnect Connections (VFYOPCCNN)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

[Parameters](#)  
[Examples](#)  
[Error messages](#)

The Verify OptiConnect Connections (VFYOPCCNN) command verifies connections to all systems in the fiber optic network.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

VFYOPCCNN

This command verifies connections with all other systems that are connected to the requesting system through OptiConnect.

[Top](#)

---

### Error messages

None

[Top](#)





---

## Verify Optical (VFYOPT)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Optical (VFYOPT) command verifies whether a specified optical drive unit or a specified optical media library unit is operating.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use this command.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i>	Required, Positional 1

[Top](#)

---

### Device (DEV)

Specifies the name of the optical drive or the optical media library unit whose operation is being verified.

[Top](#)

---

### Examples

#### Example 1: Verifying an Optical Drive

```
VFYOPT  DEV(OPT1)
```

This command verifies whether the optical drive unit named OPT1 is operating.

#### Example 2: Verifying an Optical Media Library

```
VFYOPT  DEV(OPTMLB1)
```

This command verifies whether the optical media library unit named OPTMLB1 is operating.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF2C31

Optical unit description &1 not found.

##### CPF2C33

Device description &1 not an optical unit.



---

## Verify Printer (VFYPRT)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Verify Printer (VFYPRT) command runs the supported printers by causing them to print a test pattern a specified number of times. The following printers are supported:

**IPDS graphics capable:** 3812 IPDS and 4224

**SCS graphics capable:** 4214, 4234, 5224, and 5225

**SCS not graphics capable:**

- 3287
- 3812 SCS
- 4210
- 5219
- 5256
- 5262
- 5553 (DBCS only)
- 5583 (DBCS only)

**Restrictions:**

- VFYPRT does not support printers configured with \*YES specified on the AFP parameter. Some printers, such as the 3820, 3827, and 3835 can only be configured in this manner. This means that this command *can* exercise both *nonadvanced* function printers and advanced function printers, such as the 3812 and 3816 Printers, that have AFP(\*NO) specified in their device descriptions.
- The QPGMR, QSRV, and QSRVBAS user profiles have private authorities to use this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DEV	Workstation printer device	<i>Name</i>	Required, Positional 1
TIMES	Times to print	1-99, <u>1</u>	Optional

Top

---

## Workstation printer device (DEV)

Specifies the printer on which to run the test pattern. The device name must be the same as that specified in the device description for the printer.

This is a required parameter.

*name* Specify the name of the printer device description.

---

## Times to print (TIMES)

Specifies the number of times that the specified printer prints the test pattern.

- 1      The test pattern is printed one time.
- 1-99*    Specify the number of times to print the test pattern.

---

## Examples

```
VFYPRT  DEV(PRTR3)  TIMES(15)
```

This command causes printer PRTR3 to print a test pattern 15 times.

---

## Error messages

### \*ESCAPE Messages

#### CPF3943

Incorrect value specified for device parameter.

#### CPF9814

Device &1 not found.

#### CPF9825

Not authorized to device &1.

#### CPF9831

Cannot assign device &1.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

---

## Verify Service Agent (VFYSRVAGT)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Service Agent (VFYSRVAGT) command allows a user to verify a Service Agent operation. The operation to be verified is specified by the **Type (TYPE)** parameter.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TYPE	Type	*SRVCNN, *TSTPRB	Required, Positional 1
ERRLOGID	Error log identifier	Hexadecimal value, <u>00000000</u>	Optional

Top

---

### Type (TYPE)

Specifies the aspect of Service Agent to be verified.

This is a required parameter.

#### \*SRVCNN

The connection between the system or logical partition and IBM is to be verified. This connection may be used for inventory transmission whether or not Service Agent is activated for problem reporting.

#### \*TSTPRB

Service Agent is to create a Problem Log Entry with either a specified Error Log ID or a null (00000000) Error Log ID. The problem will then be reported via the normal problem reporting capabilities of Service Agent. This allows a test of the Service Agent problem reporting function. Messages for QSRV and QSYSOPR may be checked after a few minutes for Service Agent messages and the Problem Management Record (PMR).

Top

---

### Error log identifier (ERRLOGID)

Specifies the record identifier for the Product Activity Log record that is to be used to create a problem log entry for which a test problem will be sent.

**Note:** This parameter is valid only when TYPE(\*TSTPRB) is specified.

#### 00000000

Specifies that no Product Activity Log record will be used. The Problem log entry and test problem will have no Product Activity Log record identifier.

### *hexadecimal-value*

Specify the eight-character hexadecimal identifier of the desired Product Activity Log record. Identifiers may be found using the Work with Service Agent (WRKSRVAGT) command with TYPE(\*EVENT) specified, or by using the Start Service Tools (STRSST) command.

Top

---

## Examples

```
VFYSRVAGT TYPE(*TSTPRB) ERRLOGID(00000000)
```

This command will verify the operation of Service Agent by sending a test problem with no error log identifier.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9899

Error occurred during processing of command.

Top

---

## Verify Service Configuration (VFYSRVCFG)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Service Configuration (VFYSRVCFG) command allows a user to verify an existing AT&T Global Network Services (AGNS) or virtual private network (VPN) service configuration. A session will be started for the selected service using the current configuration. After the session becomes active the service configuration has been verified and the session will be ended.

### Restrictions:

- You must have use (\*USE) authority to the communications configuration objects (line, controller, and device descriptions).

Top

---

## Parameters

Keyword	Description	Choices	Notes
SERVICE	Service	*ECS, *SRVAGT	Required, Positional 1

Top

---

## Service (SERVICE)

Specifies the service to be verified.

This is a required parameter.

**\*ECS** The service configuration used by electronic customer support (ECS) and the hardware problem reporting function of iSeries Service Agent is to be verified.

**\*SRVAGT**  
The service configuration used by the inventory collection and transmission function of iSeries Service Agent is to be verified.

Top

---

## Examples

### Example 1: Verifying the ECS Service Configuration

```
VFYSRVCFG SERVICE (*ECS)
```

This command will verify that the service configuration used by electronic customer support (ECS) and the hardware problem reporting function of iSeries Service Agent can connect to IBM.

### Example 2: Verifying the iSeries Service Agent Service Configuration

```
VFYSRVCFG SERVICE (*SRVAGT)
```

This command will verify that the service configuration used by the inventory collection and transmission function of iSeries Service Agent can connect to IBM.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFB041

Parameter SERVICE required.

#### CPF9899

Error occurred during processing of command.

#### TCP8205

Required object &2/&1 type \*&3 not found.

#### TCP8211

Point-to-point profile &1 not found.

Top



---

## Verify Tape (VFYTAP)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Verify Tape (VFYTAP) command allows verification of tape unit operations for all tape units.

**Restriction:** This command is shipped with public \*EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i> , *RSRCNAME	Required, Positional 1
RSRCNAME	Resource name	<i>Name</i>	Optional, Positional 2

Top

---

### Device (DEV)

Specifies the name of the tape unit whose operation is being verified.

*device-name*

Specify the name of the tape unit whose operation is being verified.

\*RSRCNAME

The resource name of the tape unit whose operation is being verified is used.

Top

---

### Resource name (RSRCNAME)

Specifies the resource name of the tape unit whose operation is being verified.

Top

---

### Examples

```
VFYTAP DEV(TAP3)
```

This command verifies whether the tape unit named TAP3 is working.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF2B31

Tape unit description &1 not found.

#### CPF2B32

Resource &1 not found.

#### CPF2B33

Device description &1 not a tape unit.

#### CPF2B34

Resource &1 not a tape unit.

#### CPF2B35

Tape verification not available for '&1' type tape units.

#### CPF2B36

No device description was found for resource &1.

#### CPF2B37

Tape verification request not correct.

#### CPF2B39

Problem analysis did not complete due to an error.

Top

---

## Verify TCP/IP Connection (VFYTCPCNN)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Verify TCP/IP Connection (VFYTCPCNN) command, also known as PING, tests the connectivity between a system and the remote system specified by the remote system parameter.

### Notes:

- The VFYTCPCNN (PING) command cannot be used to verify IP over SNA connections.
- The local domain name is used by many applications including PING. PING appends the local domain to a host name if a domain is not specified or if a period (.) does not appear at the end of the specified host name.

Top

---

## Parameters

Keyword	Description	Choices	Notes
RMTSYS	Remote system	<i>Character value</i> , *INTNETADR	Required, Positional 1
INTNETADR	Remote internet address	<i>Character value</i>	Optional
ADRVERFMT	Address version format	* <u>CALC</u> , *IP4, *IP6	Optional
MSGMODE	Message mode	<i>Element list</i>	Optional
	Element 1: Response message detail	* <u>VERBOSE</u> , *QUIET	
	Element 2: Summary, if response errors	* <u>COMP</u> , *ESCAPE	
PKTLEN	Packet length (in bytes)	8-512, <u>256</u>	Optional
NBRPKT	Number of packets	1-999, <u>5</u>	Optional
WAITTIME	Wait time (in seconds)	1-120, <u>1</u>	Optional
LCLINTNETA	Local internet address	<i>Character value</i> , * <u>ANY</u>	Optional
TOS	Type of service	*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, * <u>NORMAL</u>	Optional
IPTTL	IP time to live (hop limit)	1-255, * <u>DFT</u>	Optional

Top

---

## Remote system (RMTSYS)

Specifies the remote system name of the host with which the Verify TCP/IP operation takes place. To be successful, the name must be valid, and the remote system must be able to communicate with the local system. You can assign names to an internet address by using either of the following:

- Work with Host Table menu, which is an option on the Configure TCP/IP menu.
- Remote name server to map a remote system name to an internet address.

Host name resolution will depend on the value specified for the **Address version format (ADRVERFMT)** parameter.

**\*INTNETADR**

The remote system is identified by the value specified for the **Remote internet address (INTNETADR)** parameter.

*character-value*

Specify the remote system name to be verified.

Top

---

## Remote internet address (INTNETADR)

Specifies the remote internet address. Either a valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

*character-value*

Specify the internet address of the remote system. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

---

## Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

**\*CALC**

The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. VFYTCPCNN (PING) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution is used in determining the IP address.

**\*IP4** Use the IP Version 4 host name resolution method.

**\*IP6** Use the IP Version 6 host name resolution method.

Top

---

## Message mode (MSGMODE)

Specifies the amount of information to be displayed.

### Element 1: Response message detail

**\*VERBOSE**

Display messages as each PING response arrives.

**\*QUIET**

Display only the initial PING (VFYTCPCNN) message and the summary messages.

### Element 2: Summary, if response errors

**\*COMP**

If the PING (CFYTCPCNN) request is successful, the summary message returned is a completion message.

### \*ESCAPE

A monitorable escape message is returned. This is useful if you have written a program to issue the PING request and wish to monitor the PING request for errors. See the error messages section of the PING (VFYTCPCNN) command help for a list of possible escape messages.

Top

---

## Packet length (in bytes) (PKTLEN)

Specifies the length (in bytes) of the packets that are sent to the remote system.

256 The packet length is 256 bytes.

*8-512* Specify the number of bytes in each packet.

Top

---

## Number of packets (NBRPKT)

Specifies the number of packets that are sent to the remote system.

5 Five packets are sent.

*1-999* Specify the number of packets that are sent to the remote system.

Top

---

## Wait time (in seconds) (WAITTIME)

Specifies the number of seconds to wait for the return (echo) packet before declaring this packet transfer a failure.

1 The system waits 1 second.

*1-120* Specify the number of second to wait.

Top

---

## Local internet address (LCLINTNETA)

Specifies the local internet address of the interface that the outbound packets are to use. Any valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

\*ANY Use any interface's local internet address.

*character-value*

Specify the local internet address.

Top

---

## Type of service (TOS)

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

**Note:** If you issue VFYTCPCNN using a local internet address (LCLINTNETA) to specify an interface to be used for outbound packets, you must also specify a type of service (TOS) that matches that interface.

**Note:** This parameter is not used if IP Version 6 address resolution is used for verifying connectivity to a remote system.

**\*NORMAL**

Normal service is used for delivery of data.

**\*MINDELAY**

Minimize delay means that prompt delivery is important for data on this connection.

**\*MAXTHRPUT**

Maximize throughput means that a high data rate is important for data on this connection.

**\*MAXRLB**

Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

**\*MINCOST**

Minimize monetary cost means that lower cost is important for data on this connection.

Top

---

## IP time to live (hop limit) (IPTTL)

Specifies the IP datagram (packet) time-to-live value. The datagram is valid only for the number of router hops specified by this parameter. The time-to-live value acts as a "hop counter". The counter is decremented each time the datagram passes through a router or gateway. Limiting the validity of the datagram by the number of hops helps to prevent internet routing loops.

**Note:** IP Version 6 refers to this parameter as the **hop limit**.

**\*DFT** Use the default time-to-live value.

The default time-to-live value for multicast addresses is 1. The default time-to-live value for all other addresses is specified by the IPTTL parameter of the Change TCP/IP Attributes (CHGTCPA) command.

**1-255** Specify an IP datagram (packet) time-to-live value.

Top

---

## Examples

### Example 1: Verify TCP/IP Connection with a Specified Host Name

```
VFYTCPCNN  RMTSYS(IPHOST)  PKTLEN(100)  NBRPKT(10)
           WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system (known to the TCP/IP configuration as IPHOST) over a TCP/IP link. Each packet transfer must take place within 15 seconds or it fails.

### Example 2: Verify TCP/IP Connection with an IP Address

```
VFYTCPCNN  RMTSYS(*INTNETADR)  INTNETADR('128.1.1.10')
           PKTLEN(100)  NBRPKT(10)  WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system over a TCP/IP interface. The user represents the RMTSYS with its internet address 128.1.1.10, rather than with an assigned system name. Each packet transfer that takes more than 15 seconds fails.

**Example 3: Verify TCP/IP Connection with Host Name and Using a Specific Local Interface Address**

```
VFYTCPCNN  RMTSYS(IPHOST)  MSGMODE(*QUIET)
           LCLINTNETA('9.2.2.3')
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system over a specific TCP/IP interface that has the local address 9.2.2.3.

Because MSGMODE(\*QUIET) is specified, only the primary output messages are displayed. The interface parameter is useful on multi-homed hosts to verify network connectivity through a specific physical interface.

**Example 4: Verify TCP/IP Connection with an IP Version 6 Address**

```
VFYTCPCNN  RMTSYS(*INTNETADR)
           INTNETADR('1:2:3:4:5:6:7:8')
```

This command attempts to verify the TCP/IP connection of a remote system that has the local address of **1:2:3:4:5:6:7:8**.

**Example 5: Verify TCP/IP Connection with a Specified IP Version 6 Defined Host Name**

```
VFYTCPCNN  RMTSYS(IPV6HOST)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

The default "Address version format" is \*CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (\*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to verify its connection over a TCP/IP link.

**Example 6: Verify TCP/IP Connection and Explicitly Use IP Version 6 Host Name Resolution**

```
VFYTCPCNN  RMTSYS(IPV6HOST)  ADRVERFMT(*IP6)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

This example differs from example 5 in that only a valid IP version 6 resolved address, for IPV6HOST, will be used when attempting to verify its connection over a TCP/IP link.

[Top](#)

---

## Error messages

None

### \*ESCAPE Messages

#### TCP3210

Connection verification statistics: &1 of &2 successful (&3 %).

#### TCP3219

Address &1 does not match address version format &2.

[Top](#)





---

## Vary Configuration (VRYCFG)

**Where allowed to run:** All environments (\*ALL)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Vary Configuration (VRYCFG) command varies on or off one or more configuration objects, with the capability of also varying on the downline attached configuration objects. The VRYCFG command also optionally resets the input/output processor (IOP) associated with the specified object.

The configuration objects that can be varied on or off are network server, network interfaces, lines, controllers, and devices. This command applies to all network interfaces, lines, controllers and devices on the system.

For the configuration object type of media library resource, this command can be used to reset the drives within a tape media library device or change the allocation of drives within a tape media library device or an optical media library device. To determine the current allocation of drive resources, use the Work with Media Library Status (WRKMLBSTS) command.

External LAN TCP/IP interfaces attached to network server objects of type \*WINDOWSNT are automatically started by default and can optionally be controlled with the **Start TCP/IP interfaces** prompt (STRTCPIFC parameter).

Downline attached objects can be varied on or off along with the specified object by specifying the value \*NET for the **Range** prompt (RANGE parameter). Downline attached objects of a network interface description are all the lines attached to the network interface, all the controllers attached to the lines, and all the devices attached to the controllers. Downline attached objects of a line are all the attached controllers and all the devices attached to the controllers. Downline attached objects of a controller are all the attached devices. Devices do not have downline attachments. The **Range** prompt (RANGE parameter) has no affect when varying devices.

Varying on network interfaces, and lines synchronously or asynchronously can be controlled by the **Vary on wait** prompt (VRYWAIT parameter). This applies only to Token-Ring, Ethernet, X.25, or switched SDLC, IDLC, BSC, and Async line descriptions. The value specified for the **Vary on wait** prompt (VRYWAIT parameter) determines how long the system will wait until either the object goes to varied on before completing the vary on command, or until the timer expires.

The Vary Configuration (VRYCFG) command can also be used to reset input/output processors. An IOP can be a communications controller, a local work station controller, or a magnetic media controller. An IOP reset is valid only when the following are being varied on:

- Network Interface Descriptions
- Lines (except twinaxial data link control (TDLC) lines)
- Local work station controllers
- Tapes
- Diskettes

A line cannot be varied on:

- Until the Network Interface Description is varied on, in the case of IDLC lines.
- Until a dial connection has been completed, in the case of switched lines.

A controller cannot be varied on:

- If the line to which it is attached is varied off, in the case of nonswitched lines.
- Until a dial connection has been completed, in the case of switched lines.

A device cannot be varied on:

- If the controller to which it is attached is varied offline. In the case of diskette devices and some of the tape devices, they are not attached to a controller, so this restriction does not apply.

A network server cannot be varied off:

- Until all attached devices and controllers are varied off. Varying off the server also varies off the attached line descriptions.
- If any AS/400 clients have files open on the server

**Note:** Use the Work with Network Service Status (WRKNWSSTS) command (available from Work with Configuration Status display) to determine the status of network server sessions with other clients.

A network interface description cannot be varied off:

- Until all attached lines, controllers and devices are varied off.

A line cannot be varied off:

- Until all the attached controllers and devices are varied off.

A controller cannot be varied off:

- If it is being used, or is allocated for use.
- Until all the attached devices are varied off.

A device cannot be varied off:

- If it is being used, or is allocated for use.

When the **Range** prompt (RANGE parameter) is used:

- For devices: The value \*NET to vary on or off downline attached objects has no effect. Devices do not have downline attached objects.
- For switched lines: The value \*NET, only when varying on, has no effect. The value \*NET, when varying off, will vary off the line and its downline attached objects.
- For Network Interface Descriptions: When varying on, the value \*NET varies on all nonswitched attachments, and when varying off, \*NET varies off all nonswitched attachments.
- The default value is \*NET

When the **Vary on wait** prompt (VRYWAIT parameter) is used:

- The time to vary on a line or network interface is the time it takes to put tasks in place to manage the line, the time to activate the communications I/O processor (IOP), including download of the IOP program, the time to establish communications with the data circuit-terminating equipment (DCE), and so on.
- Line or network interface vary on time does not include telephone dialing time; however, a powered off modem may prevent vary on completion and cause the wait time to expire. If the timer expires, an informational message will be sent to the QSYSOPR message queue. This will be followed by the vary on completion message.
- If the **Vary on wait** prompt (VRYWAIT parameter) is specified on the VRYCFG command for a line description that is not Token-Ring, Ethernet, X.25, or switched SDLC, IDLC, BSC, or Async, the parameter is accepted but ignored.

Top

## Parameters

Keyword	Description	Choices	Notes
CFGOBJ	Configuration object	Single values: *ANYNW, *APPN, *PRVCFGTYPE Other values (up to 256 repetitions): <i>Generic name, name</i>	Required, Positional 1
CFGTYPE	Type	*NWS, *NWI, *LIN, *CTL, *DEV, *MLBRSC	Required, Positional 2
STATUS	Status	*ON, *OFF, *RESET, *ALLOCATE, *UNPROTECTED, *DEALLOCATE	Required, Positional 3
RANGE	Range	*NET, *OBJ	Optional
VRYPWAIT	Vary on wait	15-180, *CFGOBJ, *NOWAIT	Optional
ASCVRYOFF	Asynchronous vary off	*NO, *YES	Optional
RESET	Reset	*NO, *YES	Optional
RSRCNAME	Resource name	Single values: *ALL Other values (up to 16 repetitions): <i>Name</i>	Optional
RESETCFGF	Reset configuration file	*NO, *YES	Optional
FRCVRYOFF	Forced vary off	*NO, *YES, *LOCK	Optional
STRTCPIFC	Start TCP/IP interfaces	*NO, *YES	Optional
SBMLTJOB	Submit multiple jobs	*NO, *YES	Optional
JOB	Job description	<i>Qualified object name</i>	Optional
	Qualifier 1: Job description	<i>Name</i> , <u>QBATCH</u>	
	Qualifier 2: Library	<i>Name</i> , <u>LIBL</u>	

Top

## Configuration object (CFGOBJ)

Specifies the names of the configuration objects to be varied on or off. Up to 256 names can be specified. Names can be up to 10 characters in length.

### \*ANYNW

All controller descriptions that specify a link type of \*ANYNW will be varied on or off. This value is only valid if CFGTYPE is \*CTL.

### \*APPN

All objects that use Advanced Peer-to-Peer Networking (APPN) will be varied on or off. This value is only valid if CFGTYPE is \*CTL or \*DEV.

### \*PRVCFGTYPE

Process all objects that were processed the last time this command was run in this job for the specified configuration object type.

### *generic-description-name*

Specify a generic description name.

**Note:** A generic name is specified as a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

You can enter multiple values for this parameter.

Top

---

## Type (CFGTYPE)

Specifies the type of object to be varied.

The possible values are:

- \***NWS** The network server is varied on or off.
- \***NWI** A network interface description is varied on or off.
- \***LIN** A line description is varied on or off.
- \***CTL** A controller description is varied on or off.
- \***DEV** A device description is varied on or off.
- \***MLBRSC**  
The status for drives within a media library is changed.

Top

---

## Status (STATUS)

Specifies whether to vary the object on or off.

- \***ON** The object is varied on.
- \***OFF** The object is varied off.
- \***RESET**  
The drive resources of the tape media library device are reset.  
**Note:** The drive resources must be specified on the RSRCTYPE parameter.  
The media library device must be varied on before this value can be specified.
- \***ALLOCATE**  
For tape, the drive resources of the tape media library device are allocated for use only by this system. If the library device is shared by multiple systems, other systems cannot use these drives while this device description is varied on. For optical, the drive resources of the optical media library device are allocated for use. The drive resources are only available for use by this media library device.  
**Note:** The drive resources must be specified on the RSRCTYPE parameter.
- \***UNPROTECTED**  
The drive resources of the tape media library device can be used by all systems that share this library device.  
**Note:** This value is not recommended. When the drive resources are in unprotected mode, each system can access the resource at the same time. Unpredictable results can occur.
- \***DEALLOCATE**  
For tape, the drive resources of the tape media library device are deallocated for this system. If the tape media library is shared by multiple systems, the drives cannot be used by this system, but can be used by other systems. For optical, the drive resources of the optical media library resource are deallocated for the media library resource. The drives are not available for use by another optical media library device.  
**Note:** The drive resources must be specified on the RSRCTYPE parameter.

Top

---

## Range (RANGE)

Specifies what configuration elements are varied.

**\*NET** All downline attached configuration elements are varied.

**\*OBJ** Only the specified object is varied.

Top

---

## Vary on wait (VRYWAIT)

Specifies whether the line is varied on asynchronously or synchronously. For synchronous vary on, this parameter specifies how long the system waits for the vary on to complete.

### **\*CFGOBJ**

The system uses the value specified on the **Vary on wait** prompt (VRYWAIT parameter) in the network interface or line description.

### **\*NOWAIT**

Do not wait for vary on completion. The network interface or line will vary on asynchronously.

### *wait-time*

Specify a value from 15 to 180 seconds in 1 second intervals. The system will wait until either the timer expires or until the line or network interface goes to varied on, before completing the VRYCFG command.

Top

---

## Asynchronous vary off (ASCVRYOFF)

Specifies whether the object is varied off synchronously or asynchronously.

**\*NO** The object is varied off synchronously.

**\*YES** The object is varied off asynchronously.

Top

---

## Reset (RESET)

Specifies if a reset is to be done for the IOP associated with the object.

The possible values are:

**\*NO** The associated IOP is not reset.

**\*YES** The associated IOP is reset.

Top

---

## Resource name (RSRCNAME)

Specifies the resource name of the drive within the media library device to be reset or reallocated.

Top

---

## Reset configuration file (RESETCFGF)

Specifies whether to reset the configuration file. This parameter is ignored if the network server description that is being varied on is of type \*WINDOWSNT. This parameter is valid only when CFGTYPE is \*NWS.

The possible values are:

- \*NO** The configuration file is not reset.
- \*YES** The configuration file is reset.

Top

---

## Forced vary off (FRCVRYOFF)

Specifies whether inquiry messages for active jobs will be issued. This parameter is not allowed when STATUS(\*ON) is specified.

**Note :** For any value other than \*NO, the ability to reject the vary request through a user exit program is revoked. See the QIBM\_QDC\_VRYEXIT point documentation of the iSeries Information Center for additional information on vary configuration exit point processing.

The possible values are:

- \*NO** Inquiry messages for active jobs will be issued.
- \*YES** Inquiry messages for active jobs will not be issued and the jobs will be ended.
- \*LOCK**

For devices other than APPC and Intra, an attempt will be made to get a lock on the device description no matter what its current status might be. If the lock is successfully obtained, it will be transferred to the system job assigned to hold the device description lock when the device is in a varied off state. If the device is in a state other than varied off, an attempt to vary off the device description will also be made.

Top

---

## Start TCP/IP interfaces (STRTCPIFC)

Specifies whether or not to start the TCP/IP interfaces associated with the external LAN ports 1 and 2 of a network server description of type \*WINDOWSNT. This parameter is valid only when CFGTYPE is \*NWS. This parameter is ignored if the network server description that is being varied on is not of type \*WINDOWSNT.

The possible values are:

- \*YES** The external LAN TCP/IP interfaces associated with ports 1 and 2 are started.
- \*NO** The external LAN TCP/IP interfaces associated with ports 1 and 2 are not started.

Top

---

## Submit multiple jobs (SBMMLTJOB)

Specifies whether or not to submit multiple batch jobs to vary the specified network server or auxiliary storage pool (ASP) device description objects. This parameter is valid only when CFGTYPE is \*NWS or \*DEV.

The possible values are:

- \*NO** All specified network server or auxiliary storage pool (ASP) device descriptions will be varied synchronously in the job under which the original VRYCFG command was issued.
- \*YES** A new VRYCFG command will be submitted in batch for each network server or auxiliary storage pool (ASP) description specified.

Top

---

## Job description (JOBDB)

Specifies the JOBDB describing the job to which multiple VRYCFG commands will be submitted in batch. This parameter is valid only when CFGTYPE is \*NWS or \*DEV and SBMMLTJOB is \*YES.

The possible values are:

### QBATCH

The job description, QBATCH, is used as the job description of the submitted job. Note that QBATCH as shipped with the system specifies job queue QBATCH which is configured to allow a maximum of 1 job to run at a time.

### *job-description-name*

Specify the name (library-name/job-description-name) of the job description used for the submitted job. In order to run several varies in parallel a job description could be created to pass jobs to the job queue QSYS/QUSRNOMAX which is shipped with no maximum on the number of active jobs:

```
CRTJOBDB JOBDB(QSYS/QUSRNOMAX) JOBQ(QSYS/QUSRNOMAX)
```

Other considerations are that the subsystem that allocates the job queue should not have the queue in a held state and the system should not be in a restricted state.

The possible library values are:

- \*LIBL** All libraries in the library list for the current thread are searched until the first match is found.

### *library-name*

Specify the name of the library where the job description name is located.

Top

---

## Examples

### Example 1: Varying On the Network Interface and Downline Attachments

```
VRYCFG CFGOBJ(NWI1) CFGTYPE(*NWI) STATUS(*ON)
```

This command varies on the network interface and all downline attachments.

### Example 2: Varying Off the Line and Attached Downline Objects

```
VRYCFG CFGOBJ(LINE1) CFGTYPE(*LIN) STATUS(*OFF)
```

This command varies off the line and all attached downline objects. The RANGE parameter took the default value of \*NET.

### Example 3: Varying on the Controller

```
VRYCFG CFGOBJ(CONTROLLER1) CFGTYPE(*CTL) STATUS(*ON)
RANGE(*OBJ)
```

This command varies on only the controller.

### Example 4: Varying on the Device

```
VRYCFG CFGOBJ(DEVICE1) CFGTYPE(*DEV)
STATUS(*ON) RANGE(*NET)
```

This command varies on only the device. Note the RANGE parameter value has no effect on devices.

### Example 5: Varying on the Line and Resetting the IOP

```
VRYCFG CFGOBJ(LINE1) CFGTYPE(*LIN) STATUS(*ON)
RANGE(*OBJ) RESET(*YES)
```

This command varies on only the line and resets the associated IOP.

### Example 6: Using Line Description Value for Wait Time

```
VRYCFG CFGOBJ(LINE1) OBJTYPE(*LIN) STATUS(*ON)
RANGE(*OBJ) VRYWAIT(*CFGOBJ)
```

This command varies on only the line and uses the vary wait time value specified in the line description for LINE1.

### Example 7: Using 80 Seconds as Vary Wait Time

```
VRYCFG CFGOBJ(LINE1) CFGTYPE(*LIN) STATUS(*ON)
RANGE(*OBJ) VRYWAIT(80)
```

This command varies on only the line using 80 seconds as the vary wait time value.

### Example 8: Varying on a Network Server

```
VRYCFG CFGOBJ(SERVER1) CFGTYPE(*NWS) STATUS(*ON)
```

This command varies on the network server description named SERVER1 and its attached line descriptions. The vary on wait value specified in the network server description is used. Note that the RANGE and RESET parameters are ignored for network servers if they are specified.

### Example 9: Resetting Drives Within a Media Library

```
VRYCFG CFGOBJ(MYLIBRARY) CFGTYPE(*MLBRSC) STATUS(*RESET)
RSRCNAME(TAP01 TAP02)
```

This command resets the drives TAP01 and TAP02 within the media library device MYLIBRARY. The device MYLIBRARY must be varied on to perform this action.

### Example 10: Deallocating Drives Within a Media Library

```
VRYCFG CFGOBJ(MYLIBRARY) CFGTYPE(*MLBRSC)
STATUS(*DEALLOCATE) RSRCNAME(OPT02)
```

This command deallocates drive OPT02 within the media library device MYLIBRARY. The device MYLIBRARY must be varied on to perform this action.

### Example 11: Varying On Multiple Network Servers in Parallel



```
VRYCFG  CFGOBJ(IPCS*) CFGTYPE(*NWS) STATUS(*ON)
        SBMMLTJOB(*YES) JOB(*LIBL/QBATCH)
```

This command submits a separate batch job to perform the vary on for each network server which has a name that begins with IPCS. The number of jobs that run in parallel depends on the configuration of the subsystem being used.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF26AF**

Status of drive resources in device description &1 not changed.

#### **CPF26B6**

Initialization program has ended with a hard error.

#### **CPF26B7**

Initialization program ended with soft error.

#### **CPF262E**

Error occurred during vary on at IPL processing.

#### **CPF262F**

QDCTRF stopped due to failure.

#### **CPF2640**

Vary command not processed.

#### **CPF2659**

Vary command may not have completed.

#### **CPDB1B9**

Status of auxiliary storage pool (ASP) device &1 not changed. Reason code is &2.

#### **CPD2609**

Device &25 configuration not valid. Reason code &1.

#### **CPDB1B1**

Device &25 configuration not valid. Reason code &1.

#### **CPDB157**

Device &25 configuration not valid. Reason code &1.

Top



---

## Wait (WAIT)

### Where allowed to run:

- Batch program (\*BPGM)
- Interactive program (\*IPGM)

Threadsafe: No

Parameters  
Examples  
Error messages

The Wait (WAIT) command accepts input from any display device from which user data is requested by one or more previous Receive File (RCVF), Send File (SNDF), or Send/Receive File (SNDRCVF) commands that do not wait to receive the input data. Those commands had \*NO specified in the WAIT parameter or, in the case of SNDF, had the INVITE DDS keyword option specified in the record format sent to the display, and specified a particular device file to receive and transfer the data to the CL procedure. Only one input request per device can be outstanding at any given time. If there are multiple outstanding input requests, the user data of the first device to respond to the specified device file is sent to the CL procedure. If the data is received within the wait interval, the Wait operation ends and the next command in the program is processed. Otherwise an escape message is sent to the CL procedure.

The program waits the number of seconds specified for the WAITRCD keyword of the Create Display File (CRTDSPF), Change Display File (CHGDSPF), or Override with Display File (OVRDSPF) commands for a device to respond to an input request.

### Restrictions:

- This command is valid only for display files within a CL procedure. It cannot be used with database files.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DEV	CL var for responding device	CL variable name, <u>*NONE</u>	Optional, Positional 1
OPNID	Open file identifier	Simple name, <u>*NONE</u>	Optional

Top

---

## CL var for responding device (DEV)

Specifies the name of the CL variable that receives the name of the display device that responds with user data for the CL procedure.

### \*NONE

No CL variable name is specified; the name of the responding device is not needed.

*name* Specify the name of the CL variable that receives the name of the responding device. A device name cannot be specified.

Top

---

## Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL procedure. A CL variable cannot be specified for this parameter value.

### \*NONE

No open file identifier is provided. This command will use the file associated with the DCLF command that had \*NONE specified for the OPNID parameter. Only one file can be declared in a CL procedure with \*NONE as the open file identifier.

### *simple-name*

Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL procedure.

Top

---

## Examples

### Example 1: Receiving User Data

```
DCLF  FILE(MSCREEN)
:
RCVF  DEV(DEV1)  WAIT(*NO)
:
RCVF  DEV(DEV2)  WAIT(*NO)
:
WAIT  DEV(&DEVNAM)
```

In this example, the device file MSCREEN is used to receive user data. The RCVF commands specify that the procedure does not wait for the data. The WAIT command causes the procedure to wait for the display device file MSCREEN to pass input data to it from one of its devices. The name of the responding display device is placed in the CL variable &DEVNAM. The received data is placed in the CL variables associated with the record format of the declared file.

### Example 2: Receiving Data Using Open File Identifier

```
DCLF  FILE(DF1)  RCFMT(FMT1)  OPNID(DSPF1)
:
RCVF  DEV(DEV1)  OPNID(DSPF1)  WAIT(*NO)
:
WAIT  DEV(*NONE)  OPNID(DSPF1)
```

In this example, the RCVF command specifies to use the display file associated with open file identifier DSPF1, namely DF1. The procedure does not wait for user data. When the WAIT command is issued with the same open file identifier, the data received is placed in the CL variables declared for record format FMT1 of display file DF1. The name of the responding device is not returned into a CL variable.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF0859

File override caused I/O buffer size to be exceeded.

#### CPF0882

No corresponding RCVF or SNDRCVF command for WAIT command.

#### CPF0886

Record contains a data field that is not valid.

**CPF0888**

Command not run because job being ended.

**CPF0889**

No data available for input request within specified time.

**CPF4101**

File &2 in library &3 not found or inline data file missing.

**CPF5068**

Program device &4 not found in file &2 in library &3.

**CPF5070**

File &2 in library &3 has no program devices acquired.

Top



---

## When (WHEN)

### Where allowed to run:

- Batch program (\*BPGM)
- Interactive program (\*IPGM)

Threadsafe: Yes

Parameters  
Examples  
Error messages

The When (WHEN) command evaluates a logical expression and conditionally processes CL procedure commands according to the evaluation of the expression. If the logical expression is true (a logical 1), the command (or the group of commands in a Do group) specified in the THEN parameter is processed, and all subsequent When and Otherwise commands in the Select command group are not processed. If the result of the logical expression is false (a logical 0), control passes to the next sequential When or Otherwise command in the Select group.

When an IF, DO, DOWHILE, DOUNTIL, or DOFOR command is specified on the THEN parameter, the entire group of commands is bypassed if the result of the logical expression is false. Control passes to the next When, Otherwise, or End Select command.

When the command or Do group specified by the THEN parameter is completed, control passes to the next command following the End Select command and processing continues from that command.

### Restrictions:

- This command is valid only within a CL procedure.
- This command is valid only within a SELECT-ENDSELECT command group.

Top

---

## Parameters

Keyword	Description	Choices	Notes
COND	Condition	<i>Logical value</i>	Required, Positional 1
THEN	Command	<i>Command string</i>	Optional, Positional 2

Top

---

## Condition (COND)

Specifies the logical expression that is evaluated to determine a condition in the program and what is done next. Refer to "Logical Expressions" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for a description of logical expressions. Note that variables, constants, and the %SUBSTRING, %SWITCH, and %BINARY built-in functions can be used within the expression.

This is a required parameter.

### *logical-value*

Specify the name of a CL logical variable or a logical expression.

---

## Command (THEN)

Specifies the command or group of commands (in a Do group or If command) that are processed if the result of evaluating the logical expression is true. After the command or Do group is processed, control is passed to the next command *after* the ENDSELECT command associated with this WHEN command. If the command specified in this parameter is a DO, DOWHILE, DOUNTIL, or DOFOR command, all commands within the Do group are considered to be the command specified by the parameter.

If no command is specified on the THEN parameter (a null THEN), control is passed to the next command *after* the ENDSELECT command associated with this WHEN command.

If a DO command is specified, only the DO command (not the commands specified within the Do group) is within the parentheses. For example:

```
WHEN COND(&A *EQ &B) THEN(DO)
  CMD1
  CMD2
  ...
  ENDDO
```

If the logical expression evaluates to true and no command is specified on the THEN parameter (a null THEN) control is passed to the next command *after* the ENDSELECT command associated with this WHEN command.

Any CL command can be specified on the THEN parameter, except the following commands:

- ELSE
- PGM, ENDPGM
- ENDDO
- MONMSG
- DCL, DCLF
- WHEN, OTHERWISE, ENDSELECT

Top

---

## Examples

```
DCL VAR(&NAME) TYPE(*CHAR) LEN(10)
DCL VAR(&INT) TYPE(*INT) LEN(4)
:
SELECT
  WHEN COND(&NAME *EQ *CMD) THEN(DO)
  : (group of CL commands)
  ENDDO
  WHEN COND(&INT *EQ 1 & &NAME *EQ *PGM) THEN(DO)
  : (group of CL commands)
  ENDDO
ENDSELECT
```

The WHEN specifies the command to run if its condition is evaluated to true. The WHEN commands in a SELECT group are evaluated in the order they are encountered. If a WHEN condition is not met, processing continues with the next command following the ENDSELECT command.

Top



---

## Error messages

None

[Top](#)



---

## Work with Active Jobs (WRKACTJOB)

Where allowed to run: All environments (\*ALL)  
Threadsafe: No

Parameters  
Examples  
Error messages

The Work with Active Jobs (WRKACTJOB) command allows you to work with performance and status information for the active jobs in the system. The sequence of jobs can be changed with the **Sequence (SEQ)** parameter or through operations on the display. Other parameters allow the selection of jobs to be shown on the display. The selection parameters can also be changed by operations on the display.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*, *PRINT _	Optional, Positional 1
RESET	Reset status statistics	* <u>NO</u> , *YES	Optional, Positional 2
SBS	Subsystem	Single values: * <u>ALL</u> Other values (up to 25 repetitions): <i>Name</i>	Optional
CPUPCTLMT	CPU percent limit	0.1-99.9, * <u>NONE</u>	Optional
RSPLMT	Response time limit	0.1-999.9, * <u>NONE</u>	Optional
SEQ	Sequence	* <u>SBS</u> , *AUXIO, *CPU, *CPUPCT, *FUNCTION, *INT, *JOB, *NUMBER, *POOL, *PTY, *RSP, *STS, *THREADS, *TYPE, *USER	Optional
JOB	Job name	<i>Qualifier list</i>	Optional
	Qualifier 1: Job name	<i>Generic name, name, *<u>ALL</u>, *SYS, *SBS</i>	
INTERVAL	Automatic refresh interval	5-999, * <u>PRV</u>	Optional

Top

---

### Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

\*  
\_     The output is displayed for interactive jobs or printed with the job's spooled output for non-interactive jobs.

#### \*PRINT

The output is printed with the job's spooled output.

Top

---

### Reset status statistics (RESET)

Specifies whether the active job statistics are reset.

**\*NO** The active job statistics are not reset. The measurement time interval is extended if a previous WRKACTJOB command has run in the current job. All active jobs are displayed.

**\*YES** The active job statistics are reset. A measurement time interval of zero is used. All active jobs are displayed.

Top

---

## Subsystem (SBS)

Specifies the names of the subsystems (or all subsystems) whose active jobs are displayed.

### Single values

**\*ALL** All active jobs in the system are displayed. System jobs that are not associated with any subsystem are also displayed.

### Other values (up to 25 repetitions)

*name* Specify the name of the subsystem to be displayed. All active jobs in this subsystem (including the monitor) are displayed.

Top

---

## CPU percent limit (CPUPCTLMT)

Specifies the minimum processing time percent value that a job must have before it is included on the display.

### **\*NONE**

There is no minimum processing time limit that a job must have to be displayed.

### **0.1-99.9**

Specify the minimum processing time percent limit that a job must have to be included on the display.

Top

---

## Response time limit (RSPLMT)

Specifies the minimum response time limit that a job must have before it is included on the display.

### **\*NONE**

There is no minimum response time limit that a job must have to be displayed.

### **0.1-999.9**

Specify the minimum response time limit that a job must have to be included on the display.

Top

---

## Sequence (SEQ)

Specifies the sequence of the active jobs that are displayed.

**\*SBS** The jobs are ordered on the basis of the subsystem in which they are running. Jobs that run in a subsystem (auto-start jobs, interactive jobs, batch jobs, readers, and writers) are put in alphabetical order by job name, and are indented under the subsystem with which they are associated. Subsystem monitor jobs (with the jobs in the subsystem grouped under each monitor

job) are put in alphabetical order and presented before system jobs. The system jobs are put in alphabetical order by job name, and are presented after the subsystem monitors and jobs in the subsystems.

**\*AUXIO**

Jobs are ordered by the number of auxiliary storage input/output (I/O) operations that have occurred during the measurement time interval. The largest values are presented first.

**\*CPU** Jobs are ordered by the amount of processing time they have used since the job started. The largest values are presented first.

**\*CPUPCT**

Jobs are ordered by the percent of processing unit resource they have used during the measurement interval. The largest values are presented first.

**\*FUNCTION**

Jobs are put in alphabetical order by the contents of the function field.

**\*INT** Jobs are ordered by the number of operator interactions that have occurred during the measurement interval. The largest values are presented first. Non-interactive jobs are shown last and have a blank interaction field.

**\*JOB** Jobs are put in alphabetical order by job name.

**\*NUMBER**

Jobs are ordered by job number. The largest values are presented first.

**\*POOL**

Jobs are ordered by the system pool in which they are running. The lowest values are presented first.

**\*PTY** Jobs are ordered by priority of running. The highest priority values (0) are presented first.

**\*RSP** Jobs are ordered by the average response time during the measurement interval. The largest values are presented first. Non-interactive jobs are shown last and have a blank interaction field.

**\*STS** Jobs are put in alphabetical order by the contents of the status field.

**\*THREADS**

Jobs are ordered by the number of active threads. The jobs with the largest number of active threads are presented first.

**\*TYPE** Jobs are put in alphabetical order by job type and job name within the same type.

**\*USER**

Jobs are put in alphabetical order by user name.

Top

---

## Job name (JOB)

Specifies the name of the active jobs to be displayed. Only active jobs within selected subsystems (based on the SBS parameter) are displayed. Subsystem monitor names only appear when \*ALL or \*SBS is specified. System jobs only appear when \*ALL or \*SYS is specified.

### Qualifier 1: Job name

**\*ALL** All the active jobs are displayed.

**\*SYS** All active system jobs are displayed. If a value other than the default is specified in the SBS parameter when using this value, an error message is issued.

**\*SBS** All active subsystem monitors are displayed.

*generic-name*

Specify all active jobs, that meet the criteria, that are to be displayed. System jobs and subsystem monitors are not displayed using this parameter.

*name* Specify the active job that is to be displayed. System jobs and subsystem monitors are not displayed using this parameter.

Top

---

## Automatic refresh interval (INTERVAL)

Specifies the interval (in seconds) to wait during the automatic refresh option. The default time is 300 seconds (5 minutes). Valid values range from 5 to 999 seconds. If this value is changed by the user, the value is saved and used as the default value. When automatic refresh is started the screen is refreshed automatically based on the time specified.

\*PRV The interval of time used in the previous invocation. Until an interval is specified, 300 seconds is used.

**5-999** Specify the delay time (in seconds) for automatic refresh.

Top

---

## Examples

### Example 1: Resetting Active Job Statistics

```
WRKACTJOB  RESET(*YES)  CPUPCTLMT(2)
```

This command allows the user to work with a display with no jobs appearing; the active job statistics are reset and no job has used any processing unit time since the reset point. When the display appears, the F5 key may be pressed; this causes a display of all jobs that have exceeded 2 percent of the processing unit utilization since the reset point.

### Example 2: Working With Jobs in a Subsystem

```
WRKACTJOB  SBS(QINTER)  SEQ(*INT)
```

This command allows the user to work with all jobs in the QINTER subsystem. The sequence of the jobs is by the number of operator interactions, with the job with the most interactions appearing first.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1093

Override of file device type not valid.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

#### CPF9847

Error occurred while closing file &1 in library &2.

**CPF9850**

Override of printer file &1 not allowed.

**CPF9851**

Overflow value for file &1 in &2 too small.

**CPF9871**

Error occurred while processing.

[Top](#)





---

## Work with Alerts (WRKALR)

Where allowed to run: All environments (\*ALL)  
Threatsafe: No

Parameters  
Examples  
Error messages

The Work with Alerts (WRKALR) command shows alerts that are created by your system or received from another system as part of alert focal point services.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DSPOPT	Display option	<u>*ALL</u> , *RCV, *LOCAL, *HELD	Optional, Positional 1
PERIOD	Period	<i>Element list</i>	Optional
	Element 1: Start time and date	<i>Element list</i>	
	Element 1: Start time	<i>Time</i> , <u>*AVAIL</u>	
	Element 2: Start date	<i>Date</i> , <u>*BEGIN</u> , *CURRENT	
	Element 2: End time and date	<i>Element list</i>	
	Element 1: End time	<i>Time</i> , <u>*AVAIL</u>	
Element 2: End date	<i>Date</i> , <u>*END</u>		
ALRTYPE	Alert type	Single values: <u>*ALL</u> Other values (up to 5 repetitions): <i>Character value</i> , *PERM, *TEMP, *PERF, *PAFF, *IMPEND, *UNKNOWN	Optional
ALRRSC	Alert resource	Single values: <u>*ALL</u> Other values (up to 50 repetitions): <i>Name</i>	Optional
ALRRSCTYPE	Alert resource type	Single values: <u>*ALL</u> Other values (up to 50 repetitions): <i>Character value</i>	Optional
ASNUSER	User assigned	Single values: <u>*ALL</u> Other values (up to 50 repetitions): <i>Character value</i> , *NONE	Optional
GROUP	Group	Single values: <u>*ALL</u> Other values (up to 50 repetitions): <i>Name</i> , *NONE, *DEFAULT	Optional
OUTPUT	Output	<u>*</u> , *PRINT	Optional
DETAIL	Detail	<u>*BASIC</u> , *EXTENDED, *FULL	Optional

Top

---

### Display option (DSPOPT)

Specifies whether alerts received from other systems or alerts created locally are shown. Alerts that cannot be sent to the system focal point and are marked as held are shown.

\*ALL All alerts that are received and locally created are shown.

\*RCV Only alerts received from other systems are shown.

### \*LOCAL

Only locally created alerts are shown.

### \*HELD

All alerts that cannot be sent to the system's focal point and are marked as held are shown.

**Note:** There is a distinction between held alerts that are sent or forwarded by this system, and held alerts that are received by another system. DSPOPT(\*HELD) shows only held alerts that could not be sent or forwarded by this system.

Top

---

## Period (PERIOD)

Specifies the period of time for which the logged alerts are shown.

### Element 1: Start time and date

#### Element 1: Start time

One of the following is used to specify the starting time at which, or after which, the alert must have been logged. Any alerts logged before the specified time and date are not shown.

#### \*AVAIL

The logged alerts that are available for the specified start date are shown.

*time* Specify the start time for the specified start date to indicate which logged alerts are shown. The time can be entered as 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

#### Element 2: Start date

One of the following is used to specify the start date on which, or after which, the alerts must have been logged. Any alerts logged before the specified date are not shown.

#### \*BEGIN

The logged alerts from the beginning of the log are shown. If \*BEGIN is specified, then any time value other than \*AVAIL for start time is ignored.

#### \*CURRENT

The logged alerts for the current day that occur between the specified start and end times (if specified) are shown.

*date* Specify the start date for which logged alerts are shown. The date must be specified in the job-date format.

## Element 2: End time and date

### Element 1: End time

One of the following is used to specify the end time before which the alerts must have been logged:

#### \*AVAIL

The logged alerts that are available for the specified end date are shown.

*time* Specify the end time for the specified end date to indicate which logged alerts are shown. The time is entered as 4 or 6 digits (hhmm or hhmmss).

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

### Element 2: End date

One of the following is used to specify the end date before which, or on which, the alerts must have been logged.

\*END The last day on which alerts were logged is the last day for which the logged alerts are shown. If \*END is specified, any time value other than \*AVAIL for end time is ignored.

*date* Specify the end date for the last day for which logged alerts are shown. The date must be specified in the job-date format.

Top

---

## Alert type (ALRTYPE)

Specifies which types of alerts are shown. The alert type indicates the severity of the alert.

### Single values

\*ALL All types of alerts are shown.

### Other values (up to 5 repetitions)

#### \*TEMP

All alerts that report a temporary problem are shown.

#### \*PERM

All alerts that report a permanent problem are shown.

\*PERF All alerts that report a performance problem are shown.

#### \*IMPEND

All alerts that report an impending problem are shown.

**\*UNKNOWN**

All alerts that report a problem with an unknown severity are shown.

**\*PAFF** All alerts that report a problem with a permanently impaired resource are shown.

***character-value***

Specify the code point for the alert type. Code points are specified with two (2) hexadecimal digits.

Top

---

## Resource name (ALRRSC)

Specifies the name of resources that are reporting problems.

**Single values**

**\*ALL** Alerts associated with all failing resources are shown.

**Other values (up to 50 repetitions)**

***name*** Specify an alert resource name. Alerts that are reporting problems associated with that alert resource name are shown.

Top

---

## Alert resource type (ALRRSCTYPE)

Specifies the types of resources that are reporting problems. Each alert resource name has an alert resource type associated with that resource.

**Single values**

**\*ALL** Alerts for all alert resource types are shown.

**Other values (up to 50 repetitions)**

***character-value***

Specify an alert resource type. Alerts that are reporting problems associated with the assigned alert resource type are shown.

Top

---

## User assigned (ASNUSER)

Specifies the user to which the alerts being shown are assigned. This value is taken from the value on the ASNUSER parameter in the Add Alert Action Entry (ADDALRACNE) command.

**Single values**

**\*ALL** All alerts are shown.

**Other values (up to 50 repetitions)**

**\*NONE**

The alerts not assigned to a user are shown.

***character-value***

Specify the name of the user to which the alerts being shown are assigned.

---

## Group (GROUP)

Specifies the group to which the alerts being shown are assigned. This value is taken from the value on the GROUP parameter in the Add Alert Selection Entry (ADDALRSLTE) command.

### Single values

**\*ALL** All alerts are shown.

### Other values (up to 50 repetitions)

**\*DEFAULT**  
The alerts assigned to the default group are shown.

**\*NONE**  
The alerts not assigned to a group are shown.

*name* Specify the name of the group to which the alerts being shown are assigned.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

**\*** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

**\*PRINT**  
The output is printed with the job's spooled output.

Top

---

## Detail (DETAIL)

Specifies the level of detail for a printed listing, if \*PRINT was specified on the **Output** prompt (OUTPUT parameter).

**\*BASIC**  
A list of the basic alert information is printed. This information includes the alert resource and type, the date and time of occurrence, the problem identification, the alert description, and the probable cause.

**\*EXTENDED**  
An extended list of alert information is printed. This information includes all of the information provided by the \*BASIC value, plus all recommended actions and the main details of the alert.

**\*FULL** Full alert information is printed. This information includes all of the information provided by the \*BASIC value, plus all recommended actions and all the details of the alert.

Top

---

## Examples

```
WRKALR  DSPOPT(*LOCAL)  ALRTYPE(*TEMP *PERM)  ALRRSCTYPE(DKT)
```

This command allows the user to work with all locally created alerts in the alert database that are both temporary and permanent. The alerts shown are reporting problems about diskettes.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9807

One or more libraries in library list deleted.

#### CPF9808

Cannot allocate one or more libraries on library list.

#### CPF9812

File &1 in library &2 not found.

#### CPF9822

Not authorized to file &1 in library &2.

#### CPF9845

Error occurred while opening file &1.

#### CPF9846

Error while processing file &1 in library &2.

#### CPF9847

Error occurred while closing file &1 in library &2.

Top

---

## Work with Alert Descriptions (WRKALRD)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Work With Alert Descriptions (WRKALRD) command allows you to view, add, change, and remove alert descriptions.

Top

---

### Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	Name, <u>*FIRST</u>	Optional, Positional 1
ALRTBL	Alert table	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Alert table	Name, <u>QCPFMSG</u>	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	

Top

---

### Message identifier (MSGID)

Specifies the message ID to work with using the WRKALRD display.

#### \*FIRST

The first alert description found in the given alert table is shown on the WRKALRD display.

*name* Specify the message identifier to work with.

Top

---

### Alert table (ALRTBL)

Specifies the alert table to work with.

#### Qualifier 1: Alert table

##### QCPFMSG

The alert table named QCPFMSG is used.

*name* Specify the name of the alert table that is used.

#### Qualifier 2: Library

\*LIBL All libraries in the job's library list are searched until the first match is found.

##### \*CURLIB

The current library is searched for the alert table. If no library is specified as the current library for the job, the QGPL library is used.

*name* Specify the name of the library where the alert table is located.

---

## Examples

```
WRKALRD  MSGID(USR1234)  ALRTBL(USER/USRMSG)
```

This command shows the Work with Alert Descriptions panel, starting with message identifier USR1234 from alert table USRMSG in library USER.

---

## Error messages

### \*ESCAPE Messages

**CPF2499**

Message identifier &1 not allowed.

**CPF7D41**

Error occurred while logging order assistance request.

**CPF7D42**

Error occurred while performing database operation.

**CPF9802**

Not authorized to object &2 in &3.

**CPF9803**

Cannot allocate object &2 in library &3.

**CPF9807**

One or more libraries in library list deleted.

**CPF9808**

Cannot allocate one or more libraries on library list.

**CPF9810**

Library &1 not found.

**CPF9811**

Program &1 in library &2 not found.

**CPF9812**

File &1 in library &2 not found.

**CPF9814**

Device &1 not found.

**CPF9820**

Not authorized to use library &1.

**CPF9821**

Not authorized to program &1 in library &2.

**CPF9822**

Not authorized to file &1 in library &2.

**CPF9825**

Not authorized to device &1.

**CPF9830**

Cannot assign library &1.



**CPF9831**

Cannot assign device &1.

**CPF9871**

Error occurred while processing.

[Top](#)



---

## Work with Alert Table (WRKALRTBL)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Work with Alert Tables (WRKALRTBL) command shows a list of alert tables and allows you to change and delete specified alert tables, work with alert descriptions contained in specified alert tables, and create new alert tables. More information on the alerts is in the Alerts Support book, SC41-5413.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the alert tables to which you have some authority will be shown on the display.
- To perform operations on the alert tables, you must have \*USE authority to the command used by the operation, and the appropriate authority to the alert tables on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
ALRTBL	Alert table	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Alert table	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALL, *ALLUSR</i>	

Top

---

## Alert table (ALRTBL)

Specifies the alert tables with which you want to work. A specific alert table name or a generic alert table name can be specified. Either type of name can be optionally qualified by a library name.

This is a required parameter.

### Qualifier 1: Alert table

**\*ALL** All alert tables in the libraries identified in the library qualifier are searched. You can display only those alert tables for which you have some authority.

#### *generic-name*

Specify the generic name of the alert tables to be shown. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, all alert tables that have names with the same prefix as the generic alert table name are shown.

*name* Specify the name of the alert table to be shown.

### Qualifier 2: Library

**\*LIBL** All libraries in the library list for the current thread are searched. All objects in these libraries with the specified object name are shown.

### \*CURLIB

The current library for the thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

### \*USRLIBL

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

### \*ALLUSR

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB    #DSULIB    #SEULIB
#COBLIB    #RPGLIB
#DFULIB    #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

```
QDSNX      QRCLxxxxx  QUSRIJS    QUSRVxRxMx
QGPL       QSRVAGT    QUSRINFSKR
QGPL38     QSYS2      QUSRNOTES
QMGTC      QSYS2xxxxx QUSROND
QMGTC2     QS36F      QUSRPOSGS
QMPGDATA   QUSER38    QUSRPOSSA
QMOMDATA   QUSRADSM   QUSRPYMSVR
QMOMPROC   QUSRBRM    QUSRDRARS
QPFRDATA   QUSRDIRCL  QUSRSYS
QRCL       QUSRDIRDB  QUSRVI
```

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

\*ALL All libraries in the system, including QSYS, are searched.

*name* Specify the name of the library to be searched.

Top

---

## Examples

```
WRKALRTBL ALRTBL(ALRTBLLIB/AL*)
```

This command shows a list of all alert tables in library ALRTBLLIB whose names begin with 'AL'. From the list shown, you can change, delete, or work with the alert descriptions in any or all of the alert tables shown. You can also create a new alert table.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9809

Library &1 cannot be accessed.

#### CPF9810

Library &1 not found.

**CPF9820**

Not authorized to use library &1.

Top



---

## Work with APPN Status (WRKAPPNSTS)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** No

Parameters  
Examples  
Error messages

The Work with APPN Status (WRKAPPNSTS) command allows you to display and work with information about the status of APPN and HPR network sessions, and RTP connections on your local system. The controller (CTL) parameter and either the RMTLOCNAME parameter or the RMTCPNAME parameter (if specified), are used to select the information to display. Note that RMTLOCNAME and RMTCPNAME cannot both be specified.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OPTION	Option	<u>*SELECT</u> , *LOC, *RTP	Optional, Positional 1
CTL	Attached controller	Generic name, name, <u>*ALL</u>	Optional
RMTNETID	Remote network identifier	Communications name, <u>*ALL</u> , *NETATR	Optional
RMTLOCNAME	Remote location	Generic name, name, <u>*ALL</u>	Optional
RMTCPNAME	Remote control point	Generic name, name, <u>*ALL</u>	Optional
MODE	Mode	Generic name, name, <u>*ALL</u> , *NETATR	Optional
TCID	Transport connection ID	Character value, <u>*ALL</u>	Optional

Top

---

### Option (OPTION)

Specifies the type of information that you can work with.

The possible values are:

\*SELECT

A list of options is shown that allows a user to select the information with which to work.

\*LOC The Work with APPN Locations panel is displayed.

\*RTP The Work with RTP Connections panel is displayed.

Top

---

### Attached controller (CTL)

Specifies the controller name for which status is shown. Only sessions using the specified controller are listed on the Work with APPN Status display.

The possible values are:

\*ALL All controllers with active APPN sessions are shown.

*generic\*-controller-name*

Specify the generic name of the controller.

*controller-name*

Specify the name of the controller.

Top

---

## Remote network identifier (RMTNETID)

Specifies the name of the remote network in which the remote control point or remote location reside.

The possible values are:

**\*ALL** All remote locations and all remote control points with active APPN sessions are shown. If \*ALL is specified for RMTNETID, then RMTCPNAME and RMTLOCNAME must be \*ALL.

**\*NETATR**

The LCLNETID value specified in the system network attributes is used.

*remote-network-identifier*

Specify the remote network identifier.

Top

---

## Remote control point (RMTLOCNAME)

Specifies the remote location name of active APPN sessions for which status is shown. Only sessions with the specified remote location name are listed on the Work with APPN Status display.

The possible values are:

**\*ALL** All remote locations with active APPN sessions are shown.

*generic\*-remote-location-name*

Specify the generic name of the remote location.

*remote-location-name*

Specify the full name of a remote location.

Top

---

## Remote control point (RMTCPNAME)

Specifies the remote control point name of active APPN sessions for which status is shown. Only sessions with the specified remote control point name are listed on the Work with APPN Status display.

For the OPTION(\*RTP) view, RMTCPNAME is used to specify the control point name for the RTP connection partner. For the OPTION(\*LOC) view, RMTCPNAME is used to specify the control point name for the attached controller.

The possible values are:

**\*ALL** All remote control points with active APPN sessions are shown.

*generic\*-remote-control-point-name*

Specify the generic name of the remote control point.



### *remote-control-point-name*

Specify the full name of a remote control point.

Top

---

## Mode (MODE)

Specifies the name of the mode by which to subset all list entries.

The possible values are:

**\*ALL** All active sessions are shown.

### **\*NETATR**

The DFTMODE value specified in the system network attributes is used.

### *generic\*-mode-point-name*

Specify the generic name of the mode.

### *mode-name*

Specify the full name of a mode.

Top

---

## Transport connection ID (TCID)

Specifies the transport connection identifier (TCID) of an RTP connection. Only sessions running over an RTP connection with the specified TCID are listed on the Work with Sessions for RTP Connections panel. This parameter is valid only when OPTION(\*RTP) is specified.

The possible values are:

**\*ALL** All TCIDs with active sessions are shown.

### *transport-connection-identifier*

Specify the TCID to be shown. When the TCID parameter is not equal to \*ALL, both the CTL and RMTCPNAME parameters must be \*ALL

Top

---

## Examples

### Example 1: Working with RTP Connections

```
WRKAPPNSTS OPTION(*RTP) TCID(*ALL)
```

This command enables the user to display all active RTP connections.

### Example 2: Working with APPN Locations

```
WRKAPPNSTS OPTION(*LOC) RMTNETID(ROCV) RMTCPNAME(ROCAS*)
```

For the specified remote control point name, this command allows the user to display all APPN location pairs that have active APPN sessions.

Top

---

## Error messages

None

[Top](#)

---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY8809  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing  
AFP  
AS/400  
CICS  
COBOL/400  
C/400  
DataPropagator  
DB2  
IBM  
Infoprint  
InfoWindow  
iSeries  
LPDA  
OfficeVision

OS/400  
Print Services Facility  
RPG/400  
SystemView  
System/36  
TCS  
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

## Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

**Personal Use:** You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

---

## Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.





Printed in USA