



System i  
Programming  
i5/OS commands  
Starting with STRS36PRC (Start S/36 Procedure)

*Version 6 Release 1*







System i

Programming

i5/OS commands

Starting with STRS36PRC (Start S/36 Procedure)

*Version 6 Release 1*

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices," on page 453.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-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, 2008. All rights reserved.**

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

---

# Contents

<b>Start S/36 Procedure (STRS36PRC)</b> . . . . .	<b>1</b>
Parameters . . . . .	1
Procedure (PRC) . . . . .	1
Current library (CURLIB) . . . . .	1
Procedure parameters (PARM) . . . . .	2
Examples . . . . .	2
Error messages . . . . .	2

<b>Start Save Synchronization (STRSAVSYNC)</b> . . . . .	<b>3</b>
Parameters . . . . .	3
Synchronization ID (SYNCID) . . . . .	3
Number of operations (NUMSYNC) . . . . .	3
Start save wait time (STRSAVWAIT) . . . . .	4
Examples . . . . .	4
Error messages . . . . .	4

<b>Start Subsystem (STRSBS)</b> . . . . .	<b>5</b>
Parameters . . . . .	6
Subsystem description (SBSD) . . . . .	6
Examples . . . . .	6
Error messages . . . . .	7

<b>Start Search Index (STRSCHIDX)</b> . . . . .	<b>9</b>
Parameters . . . . .	9
Search index (SCHIDX) . . . . .	9
Examples . . . . .	10
Error messages . . . . .	10

<b>Start Spool Reclaim (STRSPLRCL)</b> . . . . .	<b>11</b>
Parameters . . . . .	11
Output queue (OUTQ) . . . . .	11
ASP group (ASPGRP) . . . . .	13
Examples . . . . .	13
Error messages . . . . .	14

<b>Start Support Network (STRSPTN)</b> . . . . .	<b>15</b>
Parameters . . . . .	15
Account (ACCOUNT) . . . . .	15
User ID (SPTUSRID) . . . . .	15
Password (SPTPWD) . . . . .	16
Device description (FEADEV) . . . . .	16
Destination application (DESTAPP) . . . . .	16
Examples . . . . .	16
Error messages . . . . .	16

<b>Start Service Agent (STRSRVAGT)</b> . . . . .	<b>17</b>
Parameters . . . . .	17
Type (TYPE) . . . . .	17
Activation password (ACTPWD) . . . . .	18
Master password (MSTPWD) . . . . .	18
Examples . . . . .	18
Error messages . . . . .	18

<b>Start Service Job (STRSRVJOB)</b> . . . . .	<b>19</b>
Parameters . . . . .	19
Job name (JOB) . . . . .	19
Duplicate job option (DUPJOB OPT) . . . . .	20
Examples . . . . .	20
Error messages . . . . .	20

<b>Start System Service Tools (STRSST)</b> . . . . .	<b>21</b>
Parameters . . . . .	21
Examples . . . . .	21
Error messages . . . . .	21

<b>Start TCP/IP (STRTCP)</b> . . . . .	<b>23</b>
Parameters . . . . .	25
Start application servers (STRSVR) . . . . .	26
Start TCP/IP interfaces (STRIFC) . . . . .	26
Start point-to-point profiles (STRTPPPRF) . . . . .	26
Start IPv6 (STRIP6) . . . . .	26
Examples . . . . .	27
Error messages . . . . .	27

<b>Start TCP/IP Interface (STRTCPIFC)</b> . . . . .	<b>29</b>
Parameters . . . . .	29
Internet address (INTNETADR) . . . . .	29
Alias name (ALIASNAME) . . . . .	30
Line description (LIND) . . . . .	30
Examples . . . . .	31
Error messages . . . . .	32

<b>Start Point-to-Point TCP/IP (STRTCPPTP)</b> . . . . .	<b>33</b>
Parameters . . . . .	33
Configuration profile (CFGPRF) . . . . .	33
Script dialog output (OUTPUT) . . . . .	34
Restart (RESTART) . . . . .	34
Send inquiry message (SNDINQMSG) . . . . .	34
Autodelete configuration (AUTODLT CFG) . . . . .	35
Examples . . . . .	35
Error messages . . . . .	36

<b>Start TCP/IP Server (STRTCP SVR)</b> . . . . .	<b>37</b>
Parameters . . . . .	37
Server application (SERVER) . . . . .	38
Restart server (RESTART) . . . . .	41
HTTP server (HTTPSVR) . . . . .	42
DNS server (DNSSVR) . . . . .	44
TCM server (TCMSVR) . . . . .	44
ASFTOMCAT server (TOMCATSVR) . . . . .	45
SNTP service (NTPSRV) . . . . .	45
Instance (INSTANCE) . . . . .	46
Examples . . . . .	46
Error messages . . . . .	48

<b>Start TIE Session (STRTISSN)</b> . . . . .	<b>49</b>
---	-----------

Parameters . . . . .	49
User ID (SPTUSRID) . . . . .	49
Password (SPTPWD) . . . . .	49
Account (ACCOUNT) . . . . .	49
Examples . . . . .	50
Error messages . . . . .	50

**Start Trace (STRTRC) . . . . . 51**

Parameters . . . . .	52
Session ID (SSNID) . . . . .	53
Job name (JOB) . . . . .	53
Join trace (JOINTRC) . . . . .	55
Job types (JOBTYPE) . . . . .	55
Maximum storage to use (MAXSTG) . . . . .	56
Trace full (TRCFULL) . . . . .	56
Trace type (JOBTRCTYPE) . . . . .	56
Trace type (TRCTYPE) . . . . .	56
Trace filter (TRCFTR) . . . . .	58
Restart after next IPL (RESTR IPL) . . . . .	58
Watch for message (WCHMSG) . . . . .	59
Watched message queue (WCHMSGQ) . . . . .	59
Watched job (WCHJOB) . . . . .	60
Watch for LIC log entry (WCHLICLOG) . . . . .	61
Watch for PAL entry (WCHPAL) . . . . .	63
Length of time to watch (WCHTIMO) . . . . .	64
Trace program (TRCPGM) . . . . .	64
Time interval (TRCPGMITV) . . . . .	67
Run priority (RUNPTY) . . . . .	67
Examples . . . . .	67
Error messages . . . . .	69

**Start Trap Manager (STRTRPMGR) . . . . . 71**

Parameters . . . . .	71
Forward Trap (FWDTRP) . . . . .	71
Examples . . . . .	71
Error messages . . . . .	72

**Start Watch (STRWCH) . . . . . 73**

Parameters . . . . .	73
Session ID (SSNID) . . . . .	74
Watch program (WCHPGM) . . . . .	75
Call watch program (CALLWCHPGM) . . . . .	78
Watch for message (WCHMSG) . . . . .	78
Watched message queue (WCHMSGQ) . . . . .	79
Watched job (WCHJOB) . . . . .	80
Watch for LIC log entry (WCHLICLOG) . . . . .	81
Watch for PAL entry (WCHPAL) . . . . .	83
Run priority (RUNPTY) . . . . .	84
Examples . . . . .	84
Error messages . . . . .	85

**Subroutine (SUBR) . . . . . 87**

Parameters . . . . .	87
Subroutine (SUBR) . . . . .	87
Examples . . . . .	87
Error messages . . . . .	88

**Transfer Batch Job (TFRBCHJOB) . . . . . 89**

Parameters . . . . .	90
Job queue (JOBQ) . . . . .	90

Routing data (RTGDTA) . . . . .	90
Request data or command (RQSDTA) . . . . .	90
Examples . . . . .	91
Error messages . . . . .	91

**Transfer Control (TFRCTL) . . . . . 93**

Parameters . . . . .	93
Program (PGM) . . . . .	93
Parameter CL variable names (PARM) . . . . .	94
Examples . . . . .	94
Error messages . . . . .	94

**Transfer to Group Job (TFRGRPJOB) 95**

Parameters . . . . .	95
Group job (GRPJOB) . . . . .	95
Initial group program (INLGRPPGM) . . . . .	96
Special environment (SPCENV) . . . . .	96
Restore display (RSTDSP) . . . . .	96
Text 'description' (TEXT) . . . . .	97
Examples . . . . .	97
Error messages . . . . .	97

**Transfer Job (TFRJOB) . . . . . 99**

Parameters . . . . .	100
Job queue (JOBQ) . . . . .	100
Routing data (RTGDTA) . . . . .	100
Request data or command (RQSDTA) . . . . .	101
Examples . . . . .	101
Error messages . . . . .	101

**Transfer Pass-Through (TFRPASTHR) 103**

Parameters . . . . .	103
To job (TOJOB) . . . . .	103
Examples . . . . .	104
Error messages . . . . .	104

**Transfer Secondary Job (TFRSECJOB) . . . . . 105**

Parameters . . . . .	105
Examples . . . . .	105
Error messages . . . . .	105

**Trace TCP/IP Route (TRACEROUTE) 107**

Parameters . . . . .	107
Remote system (RMTSYS) . . . . .	108
Range of hops to probe (RANGE) . . . . .	108
Probes sent per hop (PROBES) . . . . .	108
Response wait time (WAITTIME) . . . . .	108
Packet length (in bytes) (PKTLEN) . . . . .	109
Output (OUTPUT) . . . . .	109
Data queue (DTAQ) . . . . .	109
Address version format (ADRVERFMT) . . . . .	109
Source IP address (LCLINTNETA) . . . . .	110
Base remote port (RMTIPORT) . . . . .	110
Lookup host names (NAMELOOKUP) . . . . .	110
Probing protocol (PROBEPCL) . . . . .	110
Allow fragmentation (FRAGMENT) . . . . .	111
Examples . . . . .	111
Error messages . . . . .	112

<b>Trace ASP Balance (TRCASPBAL)</b>	<b>113</b>
Parameters	113
Trace option setting (SET)	114
ASP number (ASP)	114
ASP device (ASPDEV)	114
Time limit (TIMLMT)	115
Examples	115
Error messages	115
<b>Trace Connection (TRCCNN)</b>	<b>117</b>
Parameters	117
Trace option setting (SET)	119
Trace type (TRCTYPE)	120
Trace full (TRCFULL)	120
Trace table name (TRCTBL)	120
Size (SIZE)	120
Trace data to be formatted (FMTDTA)	121
Coded character set identifier (CCSID)	121
Job name (JOB)	122
Spooled file number (SPLNBR)	122
Job system name (JOBSYSNAME)	122
Spooled file created (CRTDATE)	123
TCP/IP data (TCPDTA)	123
Watch for message (WCHMSG)	124
Watched message queue (WCHMSGQ)	125
Watched job (WCHJOB)	126
Watch for LIC log entry (WCHLICLOG)	126
Watch for PAL entry (WCHPAL)	128
Length of time to watch (WCHTIMO)	130
Trace program (TRCPGM)	130
Time interval (TRCPGMITV)	132
Run priority (RUNPTY)	132
Examples	133
Error messages	134
<b>Trace CPI Communications (TRCCPIC)</b>	<b>135</b>
Parameters	135
Trace option setting (SET)	136
Maximum storage to use (MAXSTG)	136
Trace full (TRCFULL)	136
User data length (DTALEN)	136
Output (OUTPUT)	137
Output file (OUTFILE)	137
Output member options (OUTMBR)	137
Examples	138
Error messages	138
<b>Trace ICF (TRCICF)</b>	<b>141</b>
Parameters	141
Trace option setting (SET)	141
Maximum storage to use (MAXSTG)	142
Trace full (TRCFULL)	142
User data length (DTALEN)	142
Output (OUTPUT)	142
Output file (OUTFILE)	143
Output member options (OUTMBR)	143
Examples	144
Error messages	144
<b>Trace Internal (TRCINT)</b>	<b>147</b>

Parameters	147
Trace option setting (SET)	151
Trace table name (TRCTBL)	151
Trace table size (SIZE)	151
Trace full (TRCFULL)	152
Trace type (TRCTYPE)	152
Job name (JOB)	155
Thread ID to include (SLTHHD)	156
Server type (SVRTYPE)	156
Task name (TASK)	157
Task number (TASKNBR)	157
Select trace points (SLTRCPNT)	158
Omit trace points (OMTTRCPNT)	158
Stop trace points (STOPTRCPNT)	159
Job types (JOBTYPE)	160
Job trace interval (JOBTRCITV)	160
TCP/IP data (TCPDTA)	160
Sockets data (SCKDTA)	162
Device (DEV)	163
Controller (CTL)	163
Line (LIN)	164
Network interface (NWI)	165
Network Server (NWS)	165
Hardware Resource (RSRCNAME)	166
Device (OUTDEV)	166
Task information (TASKINF)	166
Output (OUTPUT)	166
File to receive output (OUTFILE)	167
Output member options (OUTMBR)	167
Watch for message (WCHMSG)	167
Watched message queue (WCHMSGQ)	168
Watched job (WCHJOB)	169
Watch for LIC log entry (WCHLICLOG)	170
Watch for PAL entry (WCHPAL)	172
Length of time to watch (WCHTIMO)	173
Trace program (TRCPGM)	173
Time interval (TRCPGMITV)	176
Run priority (RUNPTY)	176
Examples	176
Error messages	177
<b>Trace Job (TRCJOB)</b>	<b>181</b>
Parameters	181
Trace option setting (SET)	182
Trace type (TRCTYPE)	182
Maximum storage to use (MAXSTG)	183
Trace full (TRCFULL)	183
Program to call before trace (EXITPGM)	183
Select procedures to trace (SLTPRC)	183
Thread ID to include (SLTHHD)	183
Output (OUTPUT)	184
File to receive output (OUTFILE)	184
Output member options (OUTMBR)	184
Examples	185
Error messages	185
<b>Trace REXX (TRCREX)</b>	<b>187</b>
Parameters	187
Trace option setting (SET)	187
Examples	188

Error messages . . . . . 188

**Trace TCP/IP Application (TRCTCPAPP) . . . . . 189**

Parameters . . . . . 190  
TCP/IP application (APP) . . . . . 192  
Trace option setting (SET) . . . . . 193  
Maximum storage for trace (MAXSTG) . . . . . 193  
Trace full action (TRCFULL) . . . . . 194  
Additional traces (ADLTRC) . . . . . 195  
Trace program (TRCPGM) . . . . . 196  
Trace title (TITLE) . . . . . 198  
User profile (USER) . . . . . 198  
Recipient mail address (MAILADR) . . . . . 199  
Recipient host name (HOST) . . . . . 199  
Remote network address (RMTNETADR) . . . . . 199  
Local network address (LCLNETADR) . . . . . 200  
Device description (DEVD) . . . . . 200  
Device type (DEVTYPE) . . . . . 201  
Trace point (TRCPNT) . . . . . 201  
Argument list (ARGLIST) . . . . . 201  
Virtual private network server (VPNSVR) . . . . . 202  
Certificate services type (CERTTYPE) . . . . . 202  
Domain name service (DNS) . . . . . 202  
PPP connection profile (PPPCNNPRF) . . . . . 203  
TCP/IP data to trace (TCPTRCDTA) . . . . . 203  
QOS trace type (QOSTRCTYPE) . . . . . 203  
HTTP server instance (HTTPSVR) . . . . . 203  
Instance (INSTANCE) . . . . . 203  
Trace level (TRCLVL) . . . . . 204  
Packet rules trace points (PKTTRCPNT) . . . . . 204  
Configuration object (CFGOBJ) . . . . . 204  
Type (CFGTYPE) . . . . . 204  
Watch for message (WCHMSG) . . . . . 205  
Watched message queue (WCHMSGQ) . . . . . 206  
Watched job (WCHJOB) . . . . . 206  
Watch for LIC log entry (WCHLICLOG) . . . . . 207  
Watch for PAL entry (WCHPAL) . . . . . 209  
Length of time to watch (WCHTIMO) . . . . . 210  
Time interval (TRCPGMITV) . . . . . 211  
Run priority (RUNPTY) . . . . . 211  
Job name (JOB) . . . . . 211  
Examples . . . . . 212  
Error messages . . . . . 215

**Trace TCP/IP Route (TRCTCPRTE) . . . . . 217**

Parameters . . . . . 217  
Remote system (RMTSYS) . . . . . 218  
Range of hops to probe (RANGE) . . . . . 218  
Probes sent per hop (PROBES) . . . . . 218  
Response wait time (WAITTIME) . . . . . 218  
Packet length (in bytes) (PKTLEN) . . . . . 219  
Output (OUTPUT) . . . . . 219  
Data queue (DTAQ) . . . . . 219  
Address version format (ADRVERFMT) . . . . . 219  
Source IP address (LCLINTNETA) . . . . . 220  
Base remote port (RMTPORT) . . . . . 220  
Lookup host names (NAMELOOKUP) . . . . . 220  
Probing protocol (PROBEPCL) . . . . . 220  
Allow fragmentation (FRAGMENT) . . . . . 221

Examples . . . . . 221  
Error messages . . . . . 222

**Translate Keystore File (TRNCKMKSF) 223**

Parameters . . . . . 223  
Keystore file (KEYSTORE) . . . . . 223  
Master key (MSTKEY) . . . . . 224  
Examples . . . . . 224  
Error messages . . . . . 224

**Remove Mounted FS (UNMOUNT) . . . 227**

Parameters . . . . . 227  
Type of file system (TYPE) . . . . . 227  
Directory mounted over (MNTOVRDIR) . . . . . 228  
Mounted file system (MFS) . . . . . 228  
Examples . . . . . 228  
Error messages . . . . . 229

**Update Data with Temp Program (UPDDTA) . . . . . 231**

Error messages for UPDDTA . . . . . 231  
Parameters . . . . . 231  
Data base file (FILE) . . . . . 231  
Member (MBR) . . . . . 231  
Examples . . . . . 232  
Error messages . . . . . 232

**Update Program (UPDPGM) . . . . . 233**

Parameters . . . . . 233  
Program (PGM) . . . . . 234  
Module (MODULE) . . . . . 234  
Replacement library (RPLLIB) . . . . . 235  
Bind service program (BNDSRVPGM) . . . . . 236  
Bound \*SRVPGM library name (SRVPGMLIB) . . . . . 237  
Binding directory (BNDDIR) . . . . . 237  
Activation group (ACTGRP) . . . . . 238  
Creation options (OPTION) . . . . . 238  
Listing detail (DETAIL) . . . . . 239  
Examples . . . . . 239  
Error messages . . . . . 240

**Update PTF Information (UPDPTFINF) 241**

Parameters . . . . . 241  
Examples . . . . . 241  
Error messages . . . . . 241

**Update Service Program (UPDSRVPGM) . . . . . 243**

Parameters . . . . . 243  
Service program (SRVPGM) . . . . . 244  
Module (MODULE) . . . . . 245  
Export (EXPORT) . . . . . 245  
Export source file (SRCFILE) . . . . . 246  
Export source member (SRCMBR) . . . . . 246  
Replacement library (RPLLIB) . . . . . 247  
Bind service program (BNDSRVPGM) . . . . . 247  
Bound \*SRVPGM library name (SRVPGMLIB) . . . . . 248  
Binding directory (BNDDIR) . . . . . 248  
Activation group (ACTGRP) . . . . . 249



Creation options (OPTION)	249
Listing detail (DETAIL)	250
Examples	251
Error messages	251

<b>Update System Information (UPDSYSINF)</b>	<b>253</b>
Parameters	253
Library (LIB)	253
Type of information (TYPE)	254
Examples	254
Error messages	254

<b>Verify APPC Connection (VFYAPPCNN)</b>	<b>257</b>
Parameters	257
Remote location (RMTLOCNAME)	257
Mode (MODE)	257
Remote user ID (RMTUSER)	258
Remote password (RMTPWD)	258
Message mode (MSGMODE)	258
Packet length (in bytes) (PKTLEN)	258
Number of iterations (NBRITER)	259
Number of packets (NBRPKT)	259
Echo (ECHO)	259
Wait time (in seconds) (WAITTIME)	259
Examples	260
Error messages	260

<b>Verify Communications (VFYCMN)</b>	<b>261</b>
Parameters	261
Verification type (VFYTYPE)	261
Remote control point (RCPNAME)	262
Network ID (NETID)	262
User ID (USERID)	262
Password (PASSWORD)	262
Examples	263
Error messages	264

<b>Verify Image Catalog (VFYIMGCLG)</b>	<b>265</b>
Parameters	265
Image catalog (IMGCLG)	265
Verify type (TYPE)	265
Sort image catalog (SORT)	266
Network file server share (NFSSHR)	266
Examples	267
Error messages	267

<b>Verify Link supporting LPDA-2 (VFYLNKLPDA)</b>	<b>269</b>
Parameters	269
Line (LINE)	269
Test (TEST)	269
Local DCE address (LCLDCEADR)	270
Remote DCE address (RMTDCEADR)	270
Output (OUTPUT)	270
Number of sequences (SEQCOUNT)	271
Remote DTE port (DTEPORT)	271
DTE retry (DTERTY)	271

DCE retry (DCERTY)	271
Link status after test (VRYLNKSTS)	272
Examples	272
Error messages	273

<b>Verify OptiConnect Connections (VFYOPCCNN)</b>	<b>275</b>
Parameters	275
Examples	275
Error messages	275

<b>Verify Optical (VFYOPT)</b>	<b>277</b>
Parameters	277
Device (DEV)	277
Examples	277
Error messages	277

<b>Verify Printer (VFYPRT)</b>	<b>279</b>
Parameters	279
Workstation printer device (DEV)	279
Times to print (TIMES)	280
Examples	280
Error messages	280

<b>Verify Service Agent (VFYSRVAGT)</b>	<b>281</b>
Parameters	281
Type (TYPE)	281
Error log identifier (ERRLOGID)	281
Examples	282
Error messages	282

<b>Verify Service Configuration (VFYSRVCFG)</b>	<b>283</b>
Parameters	283
Service (SERVICE)	283
Examples	283
Error messages	284

<b>Verify Tape (VFYTAP)</b>	<b>285</b>
Parameters	285
Device (DEV)	285
Resource name (RSRCNAME)	285
Examples	285
Error messages	286

<b>Verify TCP/IP Connection (VFYTCPCNN)</b>	<b>287</b>
Parameters	287
Remote system (RMTSYS)	287
Remote internet address (INTNETADR)	288
Address version format (ADRVERFMT)	288
Message mode (MSGMODE)	288
Packet length (in bytes) (PKTLEN)	289
Number of packets (NBRPKT)	289
Wait time (in seconds) (WAITTIME)	289
Local internet address (LCLINTNETA)	289
Type of service (TOS)	289
IP time to live (hop limit) (IPTTL)	290
Examples	290

Error messages . . . . . 292

**Vary Configuration (VRYCFG) . . . . . 293**

Parameters . . . . . 295  
Configuration object (CFGOBJ) . . . . . 295  
Type (CFGTYPE) . . . . . 296  
Status (STATUS) . . . . . 296  
Range (RANGE) . . . . . 297  
Vary on wait (VRYWAIT) . . . . . 297  
Asynchronous vary off (ASCVRYOFF) . . . . . 297  
Reset (RESET) . . . . . 297  
Resource name (RSRCNAME) . . . . . 298  
Forced vary off (FRCVRYOFF) . . . . . 298  
Submit multiple jobs (SBMMLTJOB) . . . . . 298  
Job description (JOB) . . . . . 299  
Generate path certificate (GENPTHCERT) . . . . . 299  
Reset system (RESESYS) . . . . . 299  
Examples . . . . . 300  
Error messages . . . . . 302

**Wait (WAIT) . . . . . 303**

Parameters . . . . . 303  
CL var for responding device (DEV) . . . . . 303  
Open file identifier (OPNID) . . . . . 304  
Examples . . . . . 304  
Error messages . . . . . 304

**When (WHEN) . . . . . 307**

Parameters . . . . . 307  
Condition (COND) . . . . . 307  
Command (THEN) . . . . . 308  
Examples . . . . . 308  
Error messages . . . . . 309

**Work with Active Jobs (WRKACTJOB) 311**

Parameters . . . . . 311  
Output (OUTPUT) . . . . . 311  
Reset status statistics (RESET) . . . . . 311  
Subsystem (SBS) . . . . . 312  
CPU percent limit (CPUPCTLMT) . . . . . 312  
Response time limit (RSPLMT) . . . . . 312  
Sequence (SEQ) . . . . . 312  
Job name (JOB) . . . . . 313  
Automatic refresh interval (INTERVAL) . . . . . 314  
Examples . . . . . 314  
Error messages . . . . . 314

**Work with Alerts (WRKALR) . . . . . 317**

Parameters . . . . . 317  
Display option (DSPOPT) . . . . . 317  
Period (PERIOD) . . . . . 318  
Alert type (ALRTYPE) . . . . . 319  
Resource name (ALRRSC) . . . . . 320  
Alert resource type (ALRRSCTYPE) . . . . . 320  
User assigned (ASNUSE) . . . . . 320  
Group (GROUP) . . . . . 321  
Output (OUTPUT) . . . . . 321  
Detail (DETAIL) . . . . . 321  
Examples . . . . . 321  
Error messages . . . . . 322

**Work with Alert Descriptions (WRKALRD) . . . . . 323**

Parameters . . . . . 323  
Message identifier (MSGID) . . . . . 323  
Alert table (ALRTBL) . . . . . 323  
Examples . . . . . 324  
Error messages . . . . . 324

**Work with Alert Table (WRKALRTBL) 327**

Parameters . . . . . 327  
Alert table (ALRTBL) . . . . . 327  
Examples . . . . . 328  
Error messages . . . . . 329

**Work with APPN Status (WRKAPPNSTS) . . . . . 331**

Parameters . . . . . 331  
Option (OPTION) . . . . . 331  
Attached controller (CTL) . . . . . 331  
Remote network identifier (RMTNETID) . . . . . 332  
Remote control point (RMTLOCNAME) . . . . . 332  
Remote control point (RMTCPNAME) . . . . . 332  
Mode (MODE) . . . . . 333  
Transport connection ID (TCID) . . . . . 333  
Examples . . . . . 333  
Error messages . . . . . 334

**Work with ARM Jobs (WRKARMJOB) 335**

Parameters . . . . . 335  
Examples . . . . . 335  
Error messages . . . . . 336

**Work with ASP Jobs (WRKASPJOB) 337**

Parameters . . . . . 337  
ASP device (ASPDEV) . . . . . 337  
Examples . . . . . 337  
Error messages . . . . . 338

**Work with Authority (WRKAUT) . . . . . 339**

Parameters . . . . . 339  
Object (OBJ) . . . . . 339  
Symbolic link (SYMLNK) . . . . . 340  
Examples . . . . . 340  
Error messages . . . . . 340

**Work with Authorization Lists (WRKAUTL) . . . . . 343**

Parameters . . . . . 343  
Authorization list (AUTL) . . . . . 343  
Examples . . . . . 343  
Error messages . . . . . 344

**Work with Binding Directories (WRKBNDDIR) . . . . . 345**

Parameters . . . . . 345  
Binding directory (BNDDIR) . . . . . 345  
Examples . . . . . 346  
Error messages . . . . . 346

**Work with Binding Dir Entries  
(WRKBNDIRE) . . . . . 349**

Parameters . . . . . 349  
Binding directory (BNDDIR) . . . . . 349  
Examples . . . . . 350  
Error messages . . . . . 350

**Work with BOOTP table (WRKBPTBL) 351**

Parameters . . . . . 351  
Examples . . . . . 351  
Error messages . . . . . 351

**Work with Configuration Lists  
(WRKCFGL) . . . . . 353**

Parameters . . . . . 353  
Configuration list (CFGL) . . . . . 353  
Examples . . . . . 354  
Error messages . . . . . 354

**Work with Configuration Status  
(WRKCFGSTS) . . . . . 355**

Parameters . . . . . 355  
Type (CFGTYPE) . . . . . 355  
Configuration description (CFGD) . . . . . 356  
Output (OUTPUT) . . . . . 358  
Remote location (RMTLOCNAME) . . . . . 358  
Range (RANGE) . . . . . 358  
Status (STATUS) . . . . . 358  
Assistance level (ASTLVL) . . . . . 359  
Examples . . . . . 359  
Error messages . . . . . 360

**Work with Chart Formats  
(WRKCHTFMT) . . . . . 361**

Parameters . . . . . 361  
Chart format (CHTFMT) . . . . . 361  
Examples . . . . . 362  
Error messages . . . . . 362

**Work with Classes (WRKCLS) . . . . . 365**

Parameters . . . . . 365  
Class (CLS) . . . . . 365  
Examples . . . . . 366  
Error messages . . . . . 366

**Work with Commands (WRKCMD) . . . 369**

Parameters . . . . . 369  
Command (CMD) . . . . . 369  
Examples . . . . . 370  
Error messages . . . . . 370

**Work with Commitment Def  
(WRKCMTDFN) . . . . . 373**

Parameters . . . . . 373  
Job name (JOB) . . . . . 374  
Status (STATUS) . . . . . 374  
ASP group (ASP Group) . . . . . 375  
Logical unit of work ID (LUWID) . . . . . 375

Output (OUTPUT) . . . . . 376  
Duplicate job option (DUPJOB OPT) . . . . . 376  
Examples . . . . . 376  
Error messages . . . . . 377

**Work with Connection Lists  
(WRKCNNL) . . . . . 379**

Parameters . . . . . 379  
Connection list (CNNL) . . . . . 379  
Examples . . . . . 379  
Error messages . . . . . 379

**Work with Contact Information  
(WRKCNTINF) . . . . . 381**

Parameters . . . . . 381  
Examples . . . . . 381  
Error messages . . . . . 381

**Work with COS Descriptions  
(WRKCOSD) . . . . . 383**

Parameters . . . . . 383  
Class-of-service description (COSD) . . . . . 383  
Examples . . . . . 383  
Error messages . . . . . 383

**Work Comm Side Information  
(WRKCSI) . . . . . 385**

Parameters . . . . . 385  
Side information (CSI) . . . . . 385  
Examples . . . . . 386  
Error messages . . . . . 387

**Work with Ctl Descriptions  
(WRKCTLD) . . . . . 389**

Parameters . . . . . 389  
Controller description (CTLD) . . . . . 389  
Examples . . . . . 389  
Error messages . . . . . 390

**Work with DB Files using IDDU  
(WRKDBFIDD) . . . . . 391**

Parameters . . . . . 391  
Library (LIB) . . . . . 391  
Examples . . . . . 391  
Error messages . . . . . 391

**Work with DDM Files (WRKDDMF) 393**

Parameters . . . . . 393  
File (FILE) . . . . . 393  
Output (OUTPUT) . . . . . 394  
Examples . . . . . 395  
Error messages . . . . . 395

**Work with Device Descriptions  
(WRKDEVD) . . . . . 397**

Parameters . . . . . 397  
Device description (DEVD) . . . . . 397  
Remote location (RMTLOCNAME) . . . . . 399

Examples . . . . .	400
Error messages . . . . .	400

<b>Work with Device Tables (WRKDEVTBL) . . . . .</b>	<b>401</b>
Parameters . . . . .	401
Device table (DEVTBL) . . . . .	401
Text 'description' (TEXT) . . . . .	401
Examples . . . . .	402
Error messages . . . . .	402

<b>Work with Directory Entries (WRKDIRE) . . . . .</b>	<b>403</b>
Parameters . . . . .	403
User identifier (USRID) . . . . .	403
User profile (USER) . . . . .	404
Command character identifier (CMDCHRID) . . . . .	404
Examples . . . . .	404
Error messages . . . . .	405

<b>Work with Directory Locations (WRKDIRLOC) . . . . .</b>	<b>407</b>
Parameters . . . . .	407
Examples . . . . .	407
Error messages . . . . .	407

<b>Work with Dir Shadow Systems (WRKDIRSHD) . . . . .</b>	<b>409</b>
Parameters . . . . .	409
Type of shadow system (TYPE) . . . . .	409
Examples . . . . .	409
Error messages . . . . .	410

<b>Work with Documents (WRKDOC) . . . . .</b>	<b>411</b>
Parameters . . . . .	411
Document (DOC) . . . . .	411
Folder (FLR) . . . . .	411
Examples . . . . .	412
Error messages . . . . .	412

<b>Work with Document Libraries (WRKDOCLIB) . . . . .</b>	<b>413</b>
Parameters . . . . .	413
Examples . . . . .	413
Error messages . . . . .	413

<b>Work with Document Print Queue (WRKDOCPRTQ) . . . . .</b>	<b>415</b>
Parameters . . . . .	415
Examples . . . . .	415
Error messages . . . . .	415

<b>Work with DSNX/PC Queues (WRKDPCQ) . . . . .</b>	<b>417</b>
Parameters . . . . .	417
Distribution queue (PCNODE) . . . . .	417
Output (OUTPUT) . . . . .	417
Examples . . . . .	418

Error messages . . . . .	418
--------------------------	-----

<b>Work with Disk Status (WRKDSKSTS) 419</b>	
Parameters . . . . .	419
Output (OUTPUT) . . . . .	419
Reset status statistics (RESET) . . . . .	419
Examples . . . . .	419
Error messages . . . . .	420

<b>Work with Distribution Lists (WRKDSTL) . . . . .</b>	<b>421</b>
Parameters . . . . .	421
List identifier (LSTID) . . . . .	421
Command character identifier (CMDCHRID) . . . . .	422
Examples . . . . .	422
Error messages . . . . .	423

<b>Work with Distribution Queue (WRKDSTQ) . . . . .</b>	<b>425</b>
Parameters . . . . .	425
Distribution (QUEUE) . . . . .	425
Output (OUTPUT) . . . . .	426
Examples . . . . .	426
Error messages . . . . .	426

<b>Work with Data Areas (WRKDTAARA) 429</b>	
Parameters . . . . .	429
Data area (DTAARA) . . . . .	429
Examples . . . . .	430
Error messages . . . . .	430

<b>Work with Data Dictionaries (WRKDTADCT) . . . . .</b>	<b>433</b>
Parameters . . . . .	433
Examples . . . . .	433
Error messages . . . . .	433

<b>Work with Data Definitions (WRKDTADFN) . . . . .</b>	<b>435</b>
Parameters . . . . .	435
Data dictionary (DTADCT) . . . . .	435
Definition type (DFNTYPE) . . . . .	435
Examples . . . . .	436
Error messages . . . . .	436

<b>Work with Data Queues (WRKDTAQ) 437</b>	
Parameters . . . . .	437
Data queue (DTAQ) . . . . .	437
Examples . . . . .	438
Error messages . . . . .	438

<b>Work with Edit Descriptions (WRKEDTD) . . . . .</b>	<b>441</b>
Parameters . . . . .	441
Edit description (EDTD) . . . . .	441
Examples . . . . .	441
Error messages . . . . .	442

<b>Work with Environment Var (WRKENVVAR)</b>	<b>443</b>
Parameters	443
Level of the environment variable. (LEVEL)	443
Examples	443
Error messages	444
<b>Work with Files (WRKF)</b>	<b>445</b>
Parameters	445
File (FILE)	445
File attributes (FILEATR)	446
Examples	447
Error messages	447
<b>Work with Function Usage (WRKFCNUSG)</b>	<b>449</b>

Parameters	449
Function ID (FCNID)	449
Examples	449
Error messages	449

<b>Work with Folders (WRKFLR)</b>	<b>451</b>
Parameters	451
Folder (FLR)	451
Examples	451
Error messages	451

<b>Appendix. Notices</b>	<b>453</b>
Programming interface information	454
Trademarks	455
Terms and conditions	456
Code license and disclaimer information	456



---

## 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 Save Synchronization (STRSAVSYNC)

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

Parameters  
Examples  
Error messages

The Start Save Synchronization (STRSAVSYNC) command starts a synchronized checkpoint for more than one save while active operation. This allows you to save objects in one or more libraries and objects in one or more directories and have all the objects reach a checkpoint together.

### Restrictions:

- You must have save system (\*SAVSYS) special authority to run this command.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SYNCID	Synchronization ID	<i>Name</i>	Required, Positional 1
NUMSYNC	Number of operations	2-32, <u>2</u>	Optional
STRSAVWAIT	Start save wait time	1-99999, <u>600</u> , *NOMAX	Optional

Top

---

## Synchronization ID (SYNCID)

Specifies the name of the synchronized checkpoint. This name must also be specified for the **Synchronization ID (SYNCID)** parameter for each of the participating save operations.

*name* Specify the name of the synchronized checkpoint.

Top

---

## Number of operations (NUMSYNC)

Specifies the number of save while active operations that will participate in the synchronized checkpoint. All of the participating save operations must start within the amount of time specified for the **Start save wait time (STRSAVWAIT)** parameter.

2 Two save while active operations will participate in the synchronized checkpoint.

2-32 Specify the number of save while active operations that will participate in the synchronized checkpoint.

Top

---

## Start save wait time (STRSAVWAIT)

Specifies the amount of time to wait for all of the participating save operations to be started. If the number of participating save operations specified for the **Number of operations (NUMSYNC)** parameter do not start within the specified time, any operations that do start within that time will be ended.

**600** The system waits up to 600 seconds for all of the participating save operations to begin.

### **\*NOMAX**

There is no maximum wait time.

### **1-99999**

Specify the number of seconds to wait for all of the participating save operations to begin.

Top

---

## Examples

```
STRSAVSYNC SYNCID(SYNCMYDATA) NUMSYNC(2)
SBMJOB CMD(SAVLIB LIB(MYLIB) DEV(TAP01)
          SAVACT(*SYNCLIB) SYNCID(SYNCMYDATA))
SBMJOB CMD(SAV DEV('/QSYS.LIB/TAP02.DEVD') OBJ('/MYDIR'))
          SAVACT(*SYNC) SYNCID(SYNCMYDATA))
```

This example shows a way to synchronize the checkpointed data when saving a library and a directory. The STRSAVSYNC command starts a synchronized checkpoint named SYNCMYDATA in which two save while active operations will participate. Then the two participating save operations are submitted for batch processing. The first submitted operation saves library MYLIB to device TAP01. The second submitted operation saves directory MYDIR to device TAP02. Each participating save operation specifies the appropriate SAVACT value for full synchronization of all the data that it saves, and also specifies the synchronization ID SYNCMYDATA.

Top

---

## Error messages

### \*ESCAPE Messages

#### **CPF37BB**

Synchronization ID &1 already started.

#### **CPF37C4**

Service &1 with protocol &2 not found.

#### **CPF37C6**

A communication error occurred.

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.
  - use (\*USE) authority to all auxiliary storage pool (ASP) device descriptions in the ASP group if the subsystem description specifies an ASP group.
2. If the subsystem description specifies an ASP group, the specified ASP group must be varied on and have a status of 'Available'.

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.

#### CPF101B

Subsystem &1 not started. ASP group &3 not available.

#### CPF101C

Not authorized to device &1.

#### 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.



---

## 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 Spool Reclaim (STRSPLRCL)

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

Parameters  
Examples  
Error messages

The Start Spool Reclaim (STRSPLRCL) command repairs output queues and spooled files that are left in unrecoverable states. If a writer job ends abnormally or an incomplete update has occurred, the output queue or spooled files associated with the writer job could remain in various statuses (such as WTR, PRT, PND and MSGW). The STRSPLRCL command does not wait for the spool reclaim function to complete. If \* is specified for the ASP group (ASPGRP) parameter and the current thread has an ASP group in its name space, two separate requests to reclaim output queues could be sent. If a matching output queue is found in \*SYSBAS, a request will be sent to the spool maintenance job in \*SYSBAS. If a matching output queue is found in the specified ASP group, a request will be sent to the spool maintenance job for that ASP group. When the reclaim function is complete, message CPC3309 is sent to the QHST and QSYSOPR message queues by each spool maintenance job.

If \*ALL/\*ALL is specified for the **Output queue (OUTQ)** parameter, and \*SYSBAS or \* is specified for the **ASP group (ASPGRP)** parameter, additional cleanup will be done on job queues and completed jobs. If a job is in OUTQ status but does not own any active spooled files, the job will be removed from the system.

### Restriction:

- You need spool control (\*SPLCTL) special authority to use this command. If a specific **ASP group** is specified for the ASPGRP parameter, the user must have use (\*USE) authority to all ASP device descriptions in the ASP group and the status of ASP group must be AVAILABLE.

Top

---

## Parameters

Keyword	Description	Choices	Notes
OUTQ	Output queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Output queue	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALL, *ALLUSR</i>	
ASPGRP	ASP group	<i>Name, *, *CURASPGRP, *SYSBAS</i>	Optional, Positional 2

Top

---

## Output queue (OUTQ)

Specifies the output queue to be reclaimed.

This is a required parameter.

### Qualifier 1: Output queue

**\*ALL** All output queues in the auxiliary storage pool (ASP) group defined by the ASPGRP parameter will be reclaimed.

*generic-name*

Specify the generic name of the output queues to be reclaimed.

*name* Specify the name of the output queue to be reclaimed.

**Qualifier 2: Library**

**\*LIBL** All libraries in the library list for the current thread are 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.

**\*ALL** All libraries in the auxiliary storage pool (ASP) group defined by the ASPGRP parameter are searched.

**\*ALLUSR**

All user libraries in the auxiliary storage pools (ASPs) defined by the **ASP group (ASPGRP)** parameter are searched.

User libraries are all libraries with names that do not begin with the letter Q except for the following:

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPGLIB	
#DFULIB	#SDALIB	

Although the following libraries with names that begin with the letter Q are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are also considered user libraries:

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVRxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMQMDATA	QUSRADSM	QUSRPOSSA	
QMQMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, 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 QUSRVRxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

**\*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.

*name* Specify the name of the library in which the output queue is located.

Top

---

## ASP group (ASPGRP)

Specifies the auxiliary storage pool (ASP) group that contains the libraries that have output queues to be reclaimed.

\*  
- Output queues which are found in the ASPs that are currently part of the thread's library name space are reclaimed. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

### \*SYSBAS

Output queues which are found in the system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) are reclaimed.

### \*CURASPGRP

Output queues which are found in the primary and secondary ASPs in the thread's ASP group are reclaimed. If no ASP group is associated with the thread, an error will be issued.

*name* Specify the name of an ASP group. Output queues which are found in the primary and secondary ASPs of the specified ASP group are reclaimed.

Top

---

## Examples

### Example 1: Repair a Single Output Queue

```
STRSPLRCL  OUTQ(QUSRSYS/PRT01)
```

This command will reclaim output queue PRT01 and all spooled files that reside on the output queue.

### Example 2: Repair Several Output Queues in a Specific Library

```
STRSPLRCL  OUTQ(QUSRSYS/PRT*)
```

This command will reclaim output queues in library QUSRSYS that have 'PRT' as the first three characters. The spooled files that reside on the output queues will be reclaimed as well.

### Example 3: Repair Output Queues in the Current User's ASP Group

```
STRSPLRCL  OUTQ(*ALL/PRT01) ASPGRP(*CURASPGRP)
```

This command will reclaim output queues named PRT01 in all libraries that are in the current user's ASP group. The spooled files that reside on the selected output queues will be reclaimed as well.

### Example 4: Repair All Output Queues and Spooled Files in System and Basic User ASPs

```
STRSPLRCL  OUTQ(*ALL/*ALL) ASPGRP(*SYSBAS)
```

This command will reclaim all output queues in all libraries that are found in the system auxiliary storage pool (ASP 1) and all defined basic user ASPs (ASPs 2-32). The spooled files that reside on the output queues will be reclaimed as well.

---

## Error messages

### \*ESCAPE Messages

**CPF338D**

\*SPLCTL required to reclaim output queues.

**CPF338F**

Value for ASPGRP not valid with special value for library.

**CPF339A**

\*CURASPGRP specified and thread has no ASP group.

**CPF339B**

Output queue &1 in library &2 not found.

**CPF9825**

Not authorized to device &1.

**CPFB8ED**

Device description &1 not correct for operation.

---

## 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
FEADEV	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.

---

## Device description (FEADEV)

Specifies the Front End Application (FEA) device description.

This is a required parameter.

---

## Destination application (DESTAPP)

Specifies the name for the destination application. The destination application is supplied by the network.

This is a required parameter.

---

## 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.

---

## Error messages

None

---

# 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 service information collection and transmission 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.

---

## 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.

---

## Examples

```
STRSRVAGT TYPE(*SBSJOB)
```

This command starts the Service Agent monitoring jobs that run in the QSYSWRK subsystem.

---

## Error messages

### \*ESCAPE Messages

#### CPF9899

Error occurred during processing of command.



---

## 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 system.

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 and Devices

**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.
- CPFB305**  
Work with service tools user IDs not allowed when signed on to SST using a default password.
- CPFB306**  
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 system. 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) or Change TCP/IP Interface (CHGTCPIFC) commands, or the System i 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:

- Bootstrap Protocol (BOOTP)
- DataLink File Manager (DLFM)
- Debug Server (DBG)
- Directory Services (DIRSRV)
- Distributed Data Management (DDM)
- Domain Name Server (DNS)
- Domino
- Dynamic Host Configuration Protocol (DHCP)
- Extended Dynamic Remote SQL (EDRSQL)
- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP)
- IBM Host On-Demand (HOD)
- IBM Online Help and Eclipse Information Center (IBMHELP)
- Internet Daemon (INETD)
- Line Printer Daemon (LPD)
- Management Central (MGTC)
- NetServer (NETSVR)
- Network Station Login Daemon (NSLD)
- On Demand (ONDMD)
- On-Demand Platform Authentication (ODPA)
- Open Shortest Path First (OSPF) Routing Protocol
- Post Office Protocol (POP)
- Quality of Service (QoS)
- Remote Execution (REXEC)
- Router Daemon (ROUTED)
- Service and Support Proxy (SRVSPTPRX)
- Simple Mail Transfer Protocol (SMTP)

- Simple Network Management Protocol (SNMP) agent
- Simple Network Time Protocol (NTP)
- Triggered Cache Manager (TCM)
- Trivial File Transfer Protocol (TFTP)
- Virtual Private Network (VPN)
- Virtual terminal support (TELNET)
- WebFacing (WEBFACING)

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 (CHGSMNMPA) command specifying AUTOSTART(\*YES)
- For the RouteD server use the Change RouteD Attributes (CHGRDTDA) 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 (CHGTFTPA) 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 TCP/IP 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 (CHGFTPA) 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 (CHGHTTPA) 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 DBG server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change DIRSVR server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change IBMHELP server attributes, including the AUTOSTART parameter, using System i Navigator.

- Change SRVSPTPRX server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change NSLD daemon attributes, including the AUTOSTART parameter, using System i Navigator.
- Change INETD daemon attributes, including the AUTOSTART parameter, using System i Navigator.
- Change MGTC daemon attributes, including the AUTOSTART parameter, using System i Navigator.
- Change ONDMD server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change OMPROUTED server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change NETSVR attributes, including the AUTOSTART parameter, using System i Navigator.
- Change DLFM server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change EDRSQL server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change HOD server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change ODPA server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change NTP server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change QoS server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change TCM server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change DOMINO server attributes, including the AUTOSTART parameter, using System i Navigator.
- Change WEBFACING server attributes, including the AUTOSTART parameter, using System i Navigator.

When the STRTCP command is issued the QTCPCTL system job will handle all requests for activating and deactivating TCP/IP interfaces.

**Note:** The STRTCP 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 (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	*YES, *NO	Optional
STRIFC	Start TCP/IP interfaces	*YES, *NO	Optional
STRPTPPRF	Start point-to-point profiles	*YES, *NO	Optional
STRIP6	Start IPv6	*YES, *NO	Optional

Top

---

## Start application servers (STRSVR)

Specifies whether or not TCP/IP application servers are started by the 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 (STRTCP SVR) 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 STRTCP command is issued the QTCPWRK system 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 (STRPTPRF)

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

---

## Start IPv6 (STRIP6)

Specifies whether or not to activate the IPv6 portion of the TCP/IP protocol stack.

**\*YES** As part of the running of the STRTCP command, the IPv6 portion of the TCP/IP protocol stack will be started. If STRIFC(\*YES) is specified, then an attempt will be made to start all of the IPv6 lines and IPv6 interfaces with a configuration attribute of AUTOSTART(\*YES).

**\*NO** The IPv6 protocol will not be started.

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) STRIP6(*YES)
```

This will start TCP/IP processing (both IPv4 and IPv6), even if the system is in restricted state. TCP/IP application servers and IP interfaces will not be started.

### Example 5: Starting TCP/IP without starting IPv6 processing

```
STRTCP STRSVR(*YES) STRIFC(*YES) STRTPPRF(*YES) STRIP6(*NO)
```

This command initializes and activates TCP/IP processing (IPv4 only), starts the TCP/IP interfaces (IPv4 only) that have the AUTOSTART parameter set to \*YES. Starts the TCP/IP servers that have the AUTOSTART parameter set to \*YES. Starts the Point to Point profiles that have the AUTOSTART parameter set to \*YES.

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 (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, *IP6SAC	Optional, Positional 1
ALIASNAME	Alias name	Simple name	Optional
LIND	Line description	Name, *ADRALIAS	Optional

Top

---

### Internet address (INTNETADR)

Specifies the internet address of the interface that had previously been added to the TCP/IP configuration with the Add TCP/IP Interface (ADDTCPIFC) command.

**Note:** Either the INTNETADR or the ALIASNAME parameter must be specified for the command, but not both of them.

#### \*AUTOSTART

Start the TCP/IP interfaces that have the AUTOSTART value set to \*YES by using the Add TCP/IP Interface (ADDTCPIFC) or the Change TCP/IP Interface (CHGTCPIFC) commands.

#### **\*IP4DHCP**

Specify that Dynamic Host Configuration Protocol should start for the associated line description. A line description name must be specified for the LIND parameter.

#### **\*IP6SAC**

Specify that IPv6 stateless address auto-configuration should start for the associated line description. A line description name must be specified for the LIND parameter.

#### ***character-value***

Specifies the internet address with the interface to be started.

An IPv4 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.

An IPv6 internet address is specified in the form *x::x::x::x::x::x*, where *x* is a hexadecimal number ranging from 0 through X'FFFF'. "::" may be used once in the IPv6 address to indicate one or more groups of 16 bits of zeros. The "::" may be used to compress leading, imbedded, or trailing zeros in the address.

An IPv6 internet address must be a unicast address and must not contain an imbedded IPv4 address (compatibility or mapped). If the IPv6 address is a link-local unicast address, a line description name must be specified for the LIND parameter.

If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

---

### **Alias name (ALIASNAME)**

Specifies the name of the interface to be started. The alias name specified may be for an IPv4 or IPv6 interface.

**Note:** Either the INTNETADR or the ALIASNAME parameter must be specified for the command, but not both of them.

#### ***simple-name***

Specify the name of the interface to be started.

Top

---

### **Line description (LIND)**

Specifies the name of the line description associated with the interface being started.

#### **\*ADRALIAS**

The line description is determined from either the INTNETADR value or the ALIASNAME value.

**name** Specify the name of the line description associated with the interface being started. The line description must be specified if the INTNETADR value is \*IP4DHCP, \*IP6SAC, or an IPv6 link-local unicast address.

Top

---

## 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.

### Example 4: Starting an Interface with the Alias Name

```
STRTCPIFC  ALIASNAME(TEST_NETWORK)
```

This command causes the TCP/IP protocol stack to activate the interface associated with the alias name TEST\_NETWORK.

### Example 5: Starting an Ethernet IPv6 Interface

```
STRTCPIFC  INTNETADR('1234:5678:9ABC:DEF0:1111:2222:3333:4444')
```

This command causes the TCP/IP protocol stack to start the IPv6 interface associated with address 1234:5678:9ABC:DEF0:1111:2222:3333:4444.

### Example 6: Starting an Ethernet IPv6 Link-Local Interface

```
STRTCPIFC  INTNETADR('FE80::1234') LIND(ETHLINE)
```

This command causes the TCP/IP protocol stack to start the IPv6 link-local interface associated with address FE80::1234 and line description ETHLINE.

### Example 7: Starting IPv6 Stateless Address Auto-Configuration On An Ethernet Line

```
STRTCPIFC  INTNETADR(*IP6SAC) LIND(ETHLINE2)
```

This command causes the TCP/IP protocol stack to start IPv6 stateless address auto-configuration for line ETHLINE2.

Top

---

## 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 the local system and establish a point-to-point TCP/IP session.
2. Dial mode (\*DIAL) sessions are used to have the local system 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 System i 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 the local system will be contacting a remote system (\*DIAL session), or if a remote system will be contacting the local system (\*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 the local system 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 the local system will be contacting a remote system (\*DIAL session), or if a remote system will be contacting the local system (\*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 local system 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 the local system.

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. Thread &10 in 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 the operating system 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 in normal operating mode. Servers can not be started when the system is in restricted state.

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 Add TCP/IP Server (ADDTCP) 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: <u>*ALL</u> , *AUTOSTART Other values (up to 300 repetitions): <i>Character value</i>	Optional, Positional 1
RESTART	Restart server	<u>*NONE</u> , *HTTP, *DNS, *DHCP, *OMPROUTED, *QOS	Optional
HTTPSVR	HTTP server	Single values: <u>*ALL</u> Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Name</i> , *ADMIN	
	Element 2: Instance startup values	<i>Character value</i> , <u>*NONE</u>	
DNSSVR	DNS server	Single values: <u>*ALL</u> Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Character value</i>	
	Element 2: Instance startup values	<i>Character value</i> , <u>*NONE</u>	
TCMSVR	TCM server	Single values: <u>*NONE</u> 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> , <u>*NONE</u>	

Keyword	Description	Choices	Notes
TOMCATSVR	ASFTOMCAT server	Single values: <b>*NONE</b> Other values: <i>Element list</i>	Optional
	Element 1: Server instance name	<i>Character value</i> , <b>*ALL</b>	
NTPSRV	SNTP service	<b>*ALL</b> , <b>*CFGFILE</b> , <b>*CLIENT</b> , <b>*SERVER</b>	Optional
INSTANCE	Instance	Single values: <b>*DFT</b> , <b>*ALL</b> , <b>*AUTOSTART</b> Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Character value</i>	
	Element 2: Instance startup values	<i>Character value</i> , <b>*NONE</b>	

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)

#### \*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.

**\*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. See the parameter documentation for the INSTANCE (INSTANCE) parameter for information on starting server instances. 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 (CHGFTP) 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.

**\*IBMHELP**

The IBM Online Help and Eclipse Information Center (IBMHELP) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*IBMHELP) will not start any additional servers.

**\*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 ENDTCPSVR command specifying SERVER(\*NTP) will end both client and server services. To change which SNTP services are active, run the 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.

#### **\*OMPROUTED**

The OMPROUTE Daemon (OMPROUTED) server is started. The OMPROUTE Daemon handles the Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) server jobs. Specifying SERVER(\*OMPROUTED) and INSTANCE(\*OSPF) will start only the OSPF server job. Specifying SERVER(\*OMPROUTED) and INSTANCE(\*RIP) will start only the RIP server job. Subsequent use of the STRTCPSVR SERVER(\*OMPROUTED) command results in a diagnostic message if the server job 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.

#### \*SRVSPTPRX

The Service and Support Proxy (SRVSPTPRX) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(\*SRVSPTPRX) will result in a diagnostic message if the SRVSPTPRX server has already been started.

#### \*SSHD

The Secure Shell (SSH) Daemon server is started. Subsequent use of the STRTCPSVR SERVER(\*SSHD) command results in a diagnostic message if the 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 (CHGTFTP) 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:

- \*DNS
- \*DHCP
- \*HTTP
- \*OMPROUTED
- \*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.

#### **\*OMPROUTED**

The OMPROUTE Daemon (OMPROUTE) server job that is already running will reprocess its configuration or index file and initialize itself with any changes that are found. The INSTANCE parameter can be used to only reprocess the OSPF files (\*OSPF) or only the RIP files (\*RIP). If the OMPROUTE 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 system 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/HTTPA/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.

**-M [no value is provided]**

Display a list of all modules compiled into the server plus those that are defined by the LoadModule directive in the configuration. The output will be generated in a spooled file for the server job.

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

---

## Instance (INSTANCE)

Specifies the server instance to be started as well as any additional startup values to be used by the server to control the server instance. This parameter may be specified if the SERVER parameter is \*DIRSRV, \*DNS, \*HTTP, \*OMPROUTED, or \*TCM. It may also be specified for user-defined servers which support multiple server instances.

If multiple server instances are supported, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

### Single values

**\*DFT** The default server or servers will be started. If the SERVER parameter is \*DIRSRV, the QUSRDIR server instance will be started. If the SERVER parameter is \*HTTP or \*DNS, all defined server instances will be started. If the SERVER parameter is \*OMPROUTED, both the \*OSPF and \*RIP server jobs are started. If the SERVER parameter is \*TCM, no defined server instances will be started.

**\*ALL** All defined instances of the server will be started.

### **\*AUTOSTART**

Server instances that are defined as being automatically started will be started. This feature may not be supported by all servers.

### Element 1: Server instance

#### *character-value*

Specify the name of the server instance to be started. Up to 32 characters may be specified.

### 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. Up to 300 characters may be specified.

**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

---

## 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)
STRTCPSVR  SERVER(*HTTP)  INSTANCE(HTTP1)
```

These commands start 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')
STRTCPSVR  SERVER(*HTTP)  INSTANCE(HTTP1 '-p 81 -sslport 443')
```

These commands start 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)
STRTCPSVR  SERVER(*DNS)  INSTANCE(DNS1)
```

These commands start 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')
STRTCPSVR  SERVER(*DNS)  INSTANCE(DNS1 '-d 5')
```

These commands start the TCP/IP DNS application server instance named DNS1, and specifies that the server instance should turn on debug level 5.

**Example 9: Starting Directory Services Instances**

```
STRTCPSVR  SERVER(*DIRSRV) INSTANCE(*AUTOSTART)
```

This command starts all instances of the Directory Services server that have been configured to be automatically started.

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

```
STRTIESSN 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)  
**Threadsafe:** 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 i5/OS through System i 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 or you specify \*YES in the **Join trace (JOINTRC)** parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Trace any user function of i5/OS through System i 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.
- Only one STRTRC command across the system is allowed to specify JOINTRC(\*YES).
- At least one generic job name must be specified if \*YES has been specified for the **Restart after next IPL (RESTR IPL)** parameter.
- Watch-related parameters cannot be specified if \*YES is specified for the RESTR IPL parameter.
- Only one STRTRC command across the system can be specified to restart after the next IPL.
- When the **Watched job (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. Job control (\*JOBCTL) special authority is also required if a generic user name is specified for the WCHJOB parameter.
- If you specify a generic user name in the WCHJOB parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Watch any job function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_WATCH\_ANY\_JOB, can also be used to change the list of users that are allowed to start and end watch operations.
- You must have operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the user exit program if specified in **Trace program (TRCPGM)** parameter, and execute (\*EXECUTE) authority to the library where the program is located.
- You must have use (\*USE) authority to the message queues specified in **Watched message queue (WCHMSGQ)** parameter, and use (\*USE) authority to the library where the message queue is located.

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	Generic name, name, *ALL	
	Qualifier 2: User	Generic name, name, *ALL	
	Qualifier 3: Number	000001-999999, *ALL	
	Element 2: Thread ID to include	Single values: *ALL, *SELECT Other values (up to 20 repetitions): <i>Character value</i> , *INITIAL	
	Element 3: Subsystem	Generic name, name, *ALLSBS	
JOINTRC	Join trace	*NO, *YES	Optional
JOBTYP	Job types	*ALL, *ACTIVE, *NEW	Optional
MAXSTG	Maximum storage to use	1024-4000000, 10000	Optional
TRCFULL	Trace full	*STOPTRC, *WRAP	Optional
JOBTRCTYP	Trace type	Values (up to 2 repetitions): *ALL, *FLOW, *DATA, *TRCTYP	Optional
TRCTYP	Trace type	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Component	<i>Character value</i> , *ASPMGT, *CKMAPI, *CLUSTER, *CMTCTL, *DBDDM, *DBDRDA, *DBHSSVR, *DBQRY, *DBSQL, *DEVCFG, *DIRSRV, *EDRSQ, *ENVVAR, *FILESVR, *FLIGHTRCD, *HPT, *HTTP, *IFS, *IPC, *IPDRV, *LOCKSPACE, *LWI, *NETSVR, *NFS, *NPS, *NWSCFG, *NWSSTG, *OBJLCK, *OPASSIST, *OPENSSL, *POSIXMISC, *POSIXPROC, *PRTTFMSRV, *PTHREAD, *QNTC, *QSHELL, *RAS, *RCDLCK, *REGFAC, *REXEC, *ROUTING, *SAVRST, *SECURITY, *SERVICE, *SIGNAL, *SMTPAPI, *SOCKETS, *SPCLCK, *SPOOL, *SVRVRY, *TCPIPCFG, *THREADMGT, *WRKMGT, *WSFM	
	Element 2: Trace level	*ERROR, *INFO, *VERBOSE	
TRCFTR	Trace filter	Name, *NONE	Optional
RESTRIP	Restart after next IPL	*NO, *YES	Optional
WCHMSG	Watch for message	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Message identifier	Name	
	Element 2: Comparison data	<i>Character value</i> , *NONE	
	Element 3: Compare against	*MSGDTA, *FROMPGM, *TOPGM	
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	Name	
	Qualifier 2: Library	Name, *LIBL	

Keyword	Description	Choices	Notes
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>Generic name, name</i>	
	Qualifier 3: Number	000001-999999, * <u>ALL</u>	
WCHLICLOG	Watch for LIC log entry	Single values: * <u>NONE</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Major code	<i>Character value, *ALL</i>	
	Element 2: Minor code	<i>Character value, *ALL</i>	
	Element 3: Comparison data	<i>Character value, *<u>NONE</u></i>	
	Element 4: Compare against	* <u>ALL</u> , *TDENBR, *TASKNAME, *SVRTYPE, *JOBNAME, *JOBUSR, *JOBNBR, *THDID, *EXCPID, *MODNAME, *MODRUNAME, *MODEPNAME, *MODOFFSET, *MODTSP	
WCHPAL	Watch for PAL entry	Single values: * <u>NONE</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: System reference code	<i>Character value, *ALL</i>	
	Element 2: Comparison data	<i>Character value, *<u>NONE</u></i>	
	Element 3: Compare against	* <u>RSCNAME</u> , *RSCTYPE, *RSCMODEL	
WCHTIMO	Length of time to watch	1-43200, * <u>NOMAX</u>	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, *<u>LIBL</u></i>	
TRCPGMITV	Time interval	1-9999, * <u>NONE</u>	Optional
RUNPTY	Run priority	1-99, <u>25</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. If the specified jobs are in the specified subsystem, then those jobs will be part of the trace collection and data will be collected for those jobs.

### Single values

**\*** Only the job that issues the STRTRC (Start Trace) command is to be traced.

## Other values (up to 8 repetitions)

### Element 1: Job name

#### Qualifier 1: Job name

**\*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.

#### *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.

*name* Specify the name of the job to be traced. Up to eight job names may be specified.

#### Qualifier 2: User

**\*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.

#### *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.

*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

#### Single values

**\*ALL** All threads within the specified job name and user name are 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.

#### Other values (up to 20 repetitions)

#### **\*INITIAL**

Only the initial thread within the specified job name and user name is traced.

### *thread-identifier*

Specify the identifiers of up to twenty threads whose information is to be included. This is the thread ID as shown by the Work with Job (WRKJOB) command.

### Element 3: Subsystem

#### \*ALLSBS

The specified jobs will be part of the trace collection regardless of which subsystem the jobs are in.

#### *generic-name*

Specify the generic name of the subsystem to be included. 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.

*name* Specify the name of the subsystem to be included.

Top

---

## Join trace (JOINTRC)

Specifies if the specified jobs will set their trace characteristics in a related job or thread, meaning that the related job or thread will be added to the trace collection. Examples of related threads are jobs submitted because of spawn API being called or pre-start jobs doing work for a parent job.

**Note:** Only one STRTRC command can be specified to set the trace characteristics of a job in a related job or thread. If another STRTRC command was previously specified with JOINTRC(\*YES), an error message is sent and the second STRTRC command fails.

**Note:** The trace characteristics will be set only to one level. This is, if a related job or thread (that already adopted the trace characteristics of the original job) generates a second job or thread, the trace characteristics will not be set to this job.

\*NO The trace characteristics of the specified jobs will not be set in a related job or thread.

\*YES The trace characteristics of the specified jobs will be set in a related job or thread.

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.

Top

---

## 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.

Top

---

## 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

Specify the component to trace. The component can be either a user-defined component name, which is typically defined by an application provider, or a pre-defined special value, which is defined by the operating system. Special value can be selected from the following table:

TRACE DESCRIPTION	SPECIAL VALUE
ASP Management	*ASPMGT
Cryptographic Key Management APIs	*CKMAPI
Cluster Resource Services	*CLUSTER
Commitment Control	*CMTCTL
Distributed Data Management	*DBDDM
Distributed Relational Database Architecture	*DBDRDA
Database Host Server	*DBHSVR
Database Query Engine	*DBQRY
Database SQL	*DBSQL
Device Configuration	*DEVCFG
Directory Services	*DIRSRV
Extended Dynamic Remote SQL (EDRS)	*EDRSQL
Environment Variable APIs	*ENVVAR
File Server	*FILESVR
Flight Recorder	*FLIGHTRCD
Host Print Transform	*HPT
HTTP Server powered by Apache	*HTTP
Integrated File System	*IFS
Interprocess Communication APIs	*IPC
Internet Print Protocol Driver	*IPPDRV
Lock Space Management	*LOCKSPACE
LightWeight Infrastructure	*LWI
System i5 NetServer	*NETSVR
Network File System	*NFS
Network Print Server	*NPS
Network Server Configuration	*NWSCFG
Network Server Storage Spaces	*NWSSTG
Object Locks	*OBJLCK
Operational Assistant	*OPASSIST
OpenSSL APIs	*OPENSSL
POSIX Miscellaneous APIs	*POSIXMISC
POSIX Process-Related APIs	*POSIXPROC
Print Transform Services	*PRTTFMSRV
POSIX Thread APIs	*PTHREAD
QNTC File System	*QNTC
Qshell	*QSHELL
Remote Access Services	*RAS
Record Locks	*RCDLCK
Registration Facility	*REGFAC
REXEC Server	*REXEC
OSPF and RIP protocols server	*ROUTING
Save/Restore	*SAVRST
Software Security	*SECURITY
Service Facility	*SERVICE
Signal APIs	*SIGNAL
SMTP APIs	*SMTPAPI
Sockets APIs	*SOCKETS
Space Location Locks	*SPCLCK
Spool functions	*SPOOL
Integrated Server Vary Processing	*SVRVRY
TCP/IP Configuration	*TCP/IPCFG
Thread Management	*THREADMGT
Work Management	*WRKMGT
Workstation Function Manager	*WSFM

## Element 2: Trace level

Specify 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

---

## Restart after next IPL (RESTR IPL)

Specifies if the trace will be restarted after the next initial program load (IPL) of the operating system.

**Note:** Only one STRTRC command can be specified to restart after the next IPL. If another STRTRC command was previously specified with RESTR IPL(\*YES), an error message is sent and this STRTRC command fails. This parameter can be specified only if at least one generic job name is specified in the **Job name (JOB)** parameter. You must run the End Trace (ENDTRC) command for the trace session which specified RESTR IPL(\*YES) to clear the storage used to hold the STRTRC command string to be restarted after an IPL. This parameter cannot be specified if any watch-related parameters have been specified.

**Note:** If the IPL is associated with an install or upgrade of the operating system, the trace will not be restarted.

**\*NO** The trace will not be restarted after an IPL.

**\*YES** The trace will be restarted after the next IPL. The trace being started by this STRTRC automatically ends if an IPL of the operating system is performed. The trace started after the next IPL will be a new trace session with new trace data. You can use the ENDTRC command to end the trace.



---

## 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, the "From program" or the "To program" includes the specified text, the watched for condition is true. If the message data, the "From program" or the "To program" 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, the "From program" or the "To program" of the watched for message. This text is case sensitive and can be quoted in order to specify imbedded or trailing blanks.

### Element 3: Compare against

Specify which part of the message the comparison data specified for element 2 is to be compared against.

#### \*MSGDATA

The comparison data will be compared against the message replacement data.

#### \*FROMPGM

The comparison data will be compared against the name of the program sending the message, or the name of the ILE program that contains the procedure sending the message.

#### \*TOPGM

The comparison data will be compared against the name of the program the message was sent to, or the name of the ILE program that contains the procedure the message was sent to.

---

## 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 message queue in library QSYS).

#### 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

##### *generic-name*

Specify the generic name of the user 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 user name specifies all jobs with the specified job name and with user names that begin with the generic prefix.

*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 or a generic user 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 field specified for element 4, the watch condition is considered to be true. This text is case sensitive. If \*ALL is specified in the LIC log compare against field, the LIC log fields which will be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile 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.

**Element 4: Compare against**

Specify which part of the LIC log the comparison data specified for element 3 is to be compared against.

**\*ALL** The LIC log comparison data will be compared against all the fields described below.

**\*TDENBR**

The LIC log comparison data will be compared against the number of the task dispatching element (TDE) which requested the LIC log entry.

**\*TASKNAME**

The LIC log comparison data will be compared against the name of the task which requested the LIC log entry. Task name is blank (hex 40s) if the LIC log entry is not requested by a task.

**\*SVRTYPE**

The LIC log comparison data will be compared against the type of server that requested the LIC log entry. Server type is blank (hex 40s) if the LIC log entry is not requested by a server.

**\*JOBNAME**

The LIC log comparison data will be compared against the name of the job which requested the LIC log entry. LIC job name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBUSR**

The LIC log comparison data will be compared against the user name of the job which requested the LIC log entry. LIC user name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBNBR**

The LIC log comparison data will be compared against the job number (000001-999999) to further qualify the job name and user name of the job which requested the LIC log entry. LIC job number is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*THDID**

The LIC log comparison data will be compared against the thread which requested the LIC log entry. Thread identifier is binary zeros if the LIC log entry is not requested by a thread.

**\*EXCPID**

The LIC log comparison data will be compared against the exception that caused the LIC log entry to be requested. This is a 2-byte hexadecimal field formed by concatenating to the high-order 1-byte exception group number a low-order 1-byte exception subtype number. Exception identifier is binary zeros if the LIC log entry is not requested as a result of an exception.

**\*MODNAME**

The LIC log comparison data will be compared against the LIC module name which requested the LIC log entry. If the module name is greater than 64 characters, the LIC module name is truncated to 64 characters.

**\*MODRUNAME**

The LIC log comparison data will be compared against the LIC module replacement unit name. LIC module RU name is always in upper case EBCDIC.

**\*MODEPNAME**

The LIC log comparison data will be compared against the name of the entry point which requested the LIC log entry. If the entry point name is greater than 128 characters, the LIC module entry point name is truncated to 128 characters.

**\*MODOFFSET**

The LIC log comparison data will be compared against the byte offset into the LIC module text which requested the LIC log entry.

**\*MODTSP**

The LIC log comparison data will be compared against the timestamp of when the LIC module was compiled. The format for this field is the system time-stamp format.

Top

---

## Watch for PAL entry (WCHPAL)

Specifies up to five Product Activity Log (PAL) entries which are to be watched for. When the watched for PAL occurs, the trace exit program is called; if no trace exit program is defined, the trace stops.

### Single values

**\*NONE**

No PAL entries will be watched for.

### Other values (up to 5 repetitions)

#### Element 1: System reference code

**\*ALL** Any system reference code will be considered to be a match.

#### *character-value*

Specify the system reference code (SRC) to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the eight-digit code. A question mark is a wildcard character that will match any digit in that position. Up to seven wildcard characters can be specified. You can also specify a generic SRC that 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 SRC specifies all PAL entries with system reference codes that begin with the generic prefix.

#### Element 2: Comparison data

Specify comparison data to be used if a PAL entry matching the specified system reference code occurs. If the field specified in element 3 matches the specified text, the watched for condition is true. If the field specified in element 3 does not match the specified text, the watch function just continues.

### \*NONE

No comparison data is specified. If a PAL entry matching the specified system reference code occurs, the watched for condition is true.

### *character-value*

Specify the text string used to compare against the field specified in element 3 of the watched for PAL entry. This text is case sensitive.

You can specify question mark (?) and asterisk (\*) wildcard characters in the text string. A question mark is a single-character wildcard and will match any character in the same position. For example, '??123' will match any value that is five characters long and ends with '123'. Multiple question mark wildcard characters can be specified for the comparison data value.

An asterisk is a multiple-character wildcard character. You can specify a single asterisk wildcard character at the end of the comparison data value. For example, 'ABC\*' will match any value that begins with the letters 'ABC'.

### **Element 3: Compare against**

Specify which part of the PAL entry the comparison data specified for element 2 is to be compared against.

### \*RSCNAME

The comparison data will be compared against the name of the physical device that has the entry in the log. A resource name is assigned at first by the system, but may have been changed to a new value by a user.

### \*RSCTYPE

The comparison data will be compared against the number or word used to identify a product.

### \*RSCMODEL

The comparison data will be compared against the numbers or letters used to identify the feature level of a product with a given type.

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 or a Product Activity Log (PAL) 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 history log.

### \*NOMAX

There is no time limit for watching for a particular message or LIC log entry or PAL 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, or a match of a Product Activity Log (PAL) entry specified for the WCHPAL 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.

**\*PAL** A match on a PAL and any associated comparison data specified on the WCHPAL parameter occurred.

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).

Allowed values for the "Comparison data" parameter when \*PAL is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch information
4 4	CHAR(8)	System reference code
12 C	CHAR(10)	Device name
22 16	CHAR(4)	Device type
26 1A	CHAR(4)	Model
30 1E	CHAR(15)	Serial number
45 2D	CHAR(10)	Resource name
55 37	CHAR(8)	Log identifier
63 3F	CHAR(8)	PAL timestamp
71 47	CHAR(4)	Reference code
75 4B	CHAR(8)	Secondary code
83 53	CHAR(8)	Table identifier
91 5B	CHAR(1)	Reserved
92 5C	BINARY(4)	Sequence
96 60	BINARY(4)	Offset to comparison data
100 64	BINARY(4)	Length of comparison data
104 68	CHAR(10)	PAL compare against
* *	CHAR(*)	PAL comparison data

For more information on the trace exit program interface, refer to the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

## Single values

### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry or Product Activity Log (PAL) 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

---

## Run priority (RUNPTY)

Specifies the priority of the job where the watch session work will be run.

**25** A job priority of 25 will be used.

**1-99** Specify the run priority of the job. For more information on job run priority, refer to the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

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/QTVTNET*) MAXSTG(80000)
```

This command starts a new trace on all jobs on the system that have a job name that begins with QTVTELNET 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.

### Example 9: Start a Trace Specifying a Subsystem

```
STRTRC  SSNID(MYTRACE)
        JOB((123456/MYUSER/MYJOB (*ALL) *ALLSBS)
           (*ALL/*ALL/QPA* (*ALL) QINTER))
```

This command starts a new trace on the job with job name MYJOB, job user name of MYUSER, and job number of 123456 regardless of which subsystem the job is in. It also traces all jobs that have a job name that begins with QPA in the QINTER subsystem.

### Example 10: Start a Trace Specifying to Set the Trace Characteristics in a Related Job or Thread

```
STRTRC  SSNID(MYTRACE)
        JOB((123456/QUSER/QZRCRVSD))
        JOINTRC(*YES)
```

This command starts a trace on the job 123456/QUSER/QZRCRVSD. If this job does a spawn or generates some work to be done in one of the prestart jobs named QZRCRVSD, the child job or thread will inherit the trace characteristics of 123456/QUSER/QZRCRVSD.

Top

---

## Error messages

### \*ESCAPE Messages

#### 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.

#### CPF39F2

Cannot allocate library &1

#### CPF98A2

Not authorized to &1 command or API.

#### CPF39D3

Unable to start/end the trace.

#### CPF39D4

Another trace session is already active that specified RESTRIPL(\*YES).

#### CPF39D5

Another trace session is already active that specified JOINTRC(\*YES).

**CPF39D6**

Not authorized to set the trace characteristics of the specified jobs in a related job or thread.

**CPF9899**

Error occurred during processing of command.

[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 i5/OS 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.

---

## Error messages

### \*ESCAPE Messages

#### CPFA80D

Trap manager job already active.

#### CPFA802

Trap manager not started.

## Start Watch (STRWCH)

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

Parameters  
 Examples  
 Error messages

The Start Watch (STRWCH) command starts the watch for event function, which notifies the user by calling a user specified program when the specified event (a message, a LIC log entry or PAL entry) occurs. PAL stands for Product Activity Log which shows errors that have occurred (such as in disk and tape units, communications, and work stations).

Up to 10000 watch sessions can be active at a time but active watch session identifiers must be unique across the system.

The watch session continues until ended with the End Watch (ENDWCH) command or with the End Watch (QSCEWCH) API. A watch 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 watch function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_SERVICE\_WATCH, can also be used to change the list of users that are allowed to start and end watch operations.
- When the **Watched job (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. Job control (\*JOBCTL) special authority is also required if a generic user name is specified for the WCHJOB parameter.
- If you specify \*ALL for the watched job name, or a generic user name, you must have all object (\*ALLOBJ) special authority, or be authorized to the Watch any job function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_WATCH\_ANY\_JOB, can also be used to change the list of users that are allowed to start and end watch operations.
- You must have operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the watch program to be called, and execute (\*EXECUTE) authority to the library where the program is located.
- You must have use (\*USE) authority to the message queues specified in **Watched message queue (WCHMSGQ)** parameter, and use (\*USE) authority to the library where the message queue is located.

Top

## Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	Name, *GEN	Required, Positional 1
WCHPGM	Watch program	Qualified object name	Required, Positional 2
	Qualifier 1: Watch program	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
CALLWCHPGM	Call watch program	Single values: *WCHEVT Other values (up to 2 repetitions): *STRWCH, *ENDWCH	Optional

Keyword	Description	Choices	Notes
WCHMSG	Watch for message	Single values: <b>*NONE</b> 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, *NONE</i>	
	Element 3: Compare against	<b>*MSGDTA</b> , <b>*FROMPGM</b> , <b>*TOPGM</b>	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional
	Element 1: Message queue	Single values: <b>*SYSOPR</b> , <b>*JOBLOG</b> , <b>*HSTLOG</b> Other values: <i>Qualified object name</i>	
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
WCHJOB	Watched job	Single values: <b>*</b> 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, *ALL</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</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
	Element 1: Major code	<i>Character value, *ALL</i>	
	Element 2: Minor code	<i>Character value, *ALL</i>	
	Element 3: Comparison data	<i>Character value, *NONE</i>	
	Element 4: Compare against	<b>*ALL</b> , <b>*TDENBR</b> , <b>*TASKNAME</b> , <b>*SVRTYPE</b> , <b>*JOBNAME</b> , <b>*JOBUSR</b> , <b>*JOBNBR</b> , <b>*THDID</b> , <b>*EXCPID</b> , <b>*MODNAME</b> , <b>*MODRUNAME</b> , <b>*MODEPNAME</b> , <b>*MODOFFSET</b> , <b>*MODTSP</b>	
WCHPAL	Watch for PAL entry	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: System reference code	<i>Character value, *ALL</i>	
	Element 2: Comparison data	<i>Character value, *NONE</i>	
	Element 3: Compare against	<b>*RSCNAME</b> , <b>*RSCTYPE</b> , <b>*RSCMODEL</b>	
RUNPTY	Run priority	1-99, <b>25</b>	Optional

Top

## Session ID (SSNID)

Specifies a session identifier for this watch. This identifier must be unique across all active watches on the system.

This is a required parameter.

**\*GEN** The system will generate a unique session identifier for this watch.

**name** Specify the session identifier for this watch. You cannot specify a session identifier that starts with "QSC".

Top



---

## Watch program (WCHPGM)

Specifies the user exit program to be called to notify that a specified watch event occurred. The exit program will be called once for each message id, LIC log entry and PAL entry specified on this command. That is, if a message is watched on a message queue and in a job log, and the message is sent to both locations, the exit program will be called twice.

This is a required parameter.

The watch program will be called:

- After a match of a message identifier and any associated comparison data specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry and any associated comparison data specified for the WCHLICLOG parameter, or a match of a Product Activity Log (PAL) entry and any associated comparison data specified for the WCHPAL parameter occurs.
- Whenever it was defined in the Call watch program (CALLWCHPGM) parameter.

There are three input parameters and one output parameter associated with the watch program. The four parameters are required:

1	Watch option setting	Input	Char(10)
2	Session Id	Input	Char(10)
3	Error detected	Output	Char(10)
4	Event data	Input	Char(*)

Allowed values for the "Watch option setting" parameter are:

### \*MSGID

A match on a message id and any associated comparison data specified on WCHMSG parameter occurred.

### \*LICLOG

A match on a LIC log and any associated comparison data specified on the WCHLICLOG parameter occurred.

**\*PAL** A match on a Product Activity Log (PAL) and any associated comparison data specified on the WCHPAL parameter occurred.

### \*STRWCH

The watch program will be called before starting watching for any event.

### \*ENDWCH

The watch program will be called when the watch session is ending. Possible reasons might be:

- End Watch (ENDWCH) command or End Watch (QSCEWCH) API was issued.
- One or more of the watch for event jobs ended abnormally or ended by user action.

**Note:** A watch session might end because an error was detected on the watch exit program. In this case, the watch program will not be called at \*ENDWCH time.

The "Session ID" parameter contains the name of the watch session that is calling the user exit program.

Allowed values for the "Error detected" parameter are:

### \*ERROR

Error detected by watch exit program. The watch session that was passed in Session ID parameter will be ended. If the watch session to be ended originally specified multiple message ids or LIC log entries or PAL entries, all of them will no longer be watched. CPI3999 message will be sent to the job log and the history log to indicate an error in the exit program caused the watch session to be ended.

## <blanks>

No error detected by watch exit program.

## Notes:

- Any value other than "ERROR" or <blanks> will be considered an error and the watch session that was passed in Session ID parameter will be ended. If the watch session to be ended originally specified multiple message ids or LIC log entries or PAL entries, all of them will no longer be watched. CPI3999 message will be sent to the job log and the history log to indicate an error in the exit program caused the watch session to be ended.

Allowed values for the "Event data" parameter when \*MSGID is specified for the "Watch option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch information
4 4	CHAR(7)	Message ID
11 B	CHAR(1)	Reserved
12 C	CHAR(10)	Message queue name
22 16	CHAR(10)	Message queue library
32 20	CHAR(10)	Job name
42 2A	CHAR(10)	User name
52 34	CHAR(6)	Job number
58 3A	CHAR(4)	Reserved
62 3E	CHAR(256)	Sending program name
318 13E	CHAR(10)	Sending module name
328 148	BINARY(4)	Offset to sending procedure name
332 14C	BINARY(4)	Length of sending procedure name
336 150	CHAR(10)	Receiving program name
346 15A	CHAR(10)	Receiving module name
356 164	BINARY(4)	Offset to receiving procedure name
360 168	BINARY(4)	Length of receiving procedure name
364 16C	BINARY(4)	Message severity
368 170	CHAR(10)	Message type
378 17A	CHAR(8)	Message timestamp
386 182	CHAR(4)	Message key
390 186	CHAR(10)	Message file name
400 190	CHAR(10)	Message file library
410 19A	CHAR(2)	Reserved
412 19C	BINARY(4)	Offset to comparison data
416 1A0	BINARY(4)	Length of comparison data
420 1A4	CHAR(10)	Compare against
430 1AE	CHAR(2)	Reserved
432 1B0	BINARY(4)	Comparison data CCSID
436 1B4	BINARY(4)	Offset where comparison data was found
440 1B8	BINARY(4)	Offset to replacement data
444 1BC	BINARY(4)	Length of replacement data
448 1C0	BINARY(4)	Replacement data CCSID
452 1C4	CHAR(10)	Sending user profile
462 1CE	CHAR(10)	Target job name
472 1D8	CHAR(10)	Target job user name
482 1E2	CHAR(6)	Target job number
* *	CHAR(*)	Sending procedure name
* *	CHAR(*)	Receiving procedure name
* *	CHAR(*)	Message comparison data
* *	CHAR(*)	Message replacement data

Allowed values for the "Event data" parameter when \*LICLOG is specified for the "Watch option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch 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	CHAR(8)	LIC Log timestamp
28	1C	CHAR(8)	TDE number
36	24	CHAR(16)	Task name
52	34	CHAR(30)	Server type
82	52	CHAR(2)	Exception ID
84	54	CHAR(10)	LIC job name
94	5E	CHAR(10)	LIC user name
104	68	CHAR(6)	LIC job number
110	6E	CHAR(4)	Reserved
114	72	CHAR(8)	Thread ID
122	7A	CHAR(8)	LIC module compile timestamp
130	82	CHAR(8)	LIC module offset
138	8A	CHAR(8)	LIC module RU name
146	92	CHAR(48)	LIC module name
194	DA	CHAR(128)	LIC module entry point name
322	142	CHAR(1)	LIC log compare against specified
323	143	CHAR(1)	Reserved
324	144	BINARY(4)	Offset to comparison data
328	148	BINARY(4)	Length of comparison data
332	14C	CHAR(10)	LIC log compare against
*	*	CHAR(*)	LIC log comparison data

Allowed values for the "Event data" parameter when \*PAL is specified for the "Watch option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec	Hex	
0	0	BINARY(4) Length of watch information
4	4	CHAR(8) System reference code
12	C	CHAR(10) Device name
22	16	CHAR(4) Device type
26	1A	CHAR(4) Model
30	1E	CHAR(15) Serial number
45	2D	CHAR(10) Resource name
55	37	CHAR(8) Log identifier
63	3F	CHAR(8) PAL timestamp
71	47	CHAR(4) Reference code
75	4B	CHAR(8) Secondary code
83	53	CHAR(8) Table identifier
91	5B	CHAR(1) Reserved
92	5C	BINARY(4) Sequence
96	60	BINARY(4) Offset to comparison data
100	64	BINARY(4) Length of comparison data
104	68	CHAR(10) PAL compare against
*	*	CHAR(*) PAL comparison data

Allowed values for the "Event data" parameter when \*STRWCH or \*ENDWCH is specified for the "Watch option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec	Hex	
0	0	BINARY(4) Length of watch information (always 4 at this time)

For more information on the watch exit program interface, refer to the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

### Qualifier 1: Watch program

*name* Specify the name of the watch exit program.

### 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 used to locate the program. 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 user exit program is located.

Top

---

## Call watch program (CALLWCHPGM)

Specifies the times at which the program specified for the **Watch program (WCHPGM)** parameter will be called. The program will always be called when the watched for event occurs. You can specify additional times for the program to be called.

### Single values

\*WCHEVT

The watch program will be called only when the watched for event occurs.

### Other values (up to 2 repetitions)

\*STRWCH

The watch program will also be called before starting watching for any event.

\*ENDWCH

The watch program will also be called when the watch session is ending. Possible reasons might be:

- End Watch (ENDWCH) command or End Watch (QSCWCH) API was issued.
- One or more of the watch for event jobs ended abnormally or ended by user action.

**Note:** A watch session might end because an error was detected on the watch exit program. In this case, the watch program will not be called at \*ENDWCH time.

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 **Watched message queue (WCHMSGQ)** parameter. When the watched for message is added to the specified message queue or log, the watch exit program is called.

### 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, the "From program" or the "To program" includes the specified text, the watched for condition is true. If the message data, the "From program" or the "To program" does not contain the specified text, the watch function just 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, the "From program" or the "To program" of the watched for message. This text is case sensitive and can be quoted in order to specify imbedded or trailing blanks.

## Element 3: Compare against

Specify which part of the message the comparison data specified for element 2 is to be compared against.

### \*MSGDATA

The comparison data will be compared against the message replacement data.

### \*FROMPGM

The comparison data will be compared against the name of the program sending the message, or the name of the ILE program that contains the procedure sending the message.

### \*TOPGM

The comparison data will be compared against the name of the program the message was sent to, or the name of the ILE program that contains the procedure the message was sent to.

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 message queue in library QSYS).

#### 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 **Watch for message (WCHMSG)** parameter. The specified job will only be watched if **\*JOBLOG** is specified on the **Watched message queue (WCHMSGQ)** parameter. Up to five job names may be specified.

### Single values

**\*** Only the job log of the job that issued this watch 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.

**\*ALL** All jobs with the specified job user name are watched. **\*ALL** for the job name is considered to be a generic job specification because it will watch all jobs that meet the job user name qualifier that you specified.

*name* Specify the name of the job to be watched.

#### Qualifier 2: User

##### *generic-name*

Specify the generic name of the user 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 user name specifies all jobs with the specified job name and with user names that begin with the generic prefix.

**\*ALL** All jobs with the specified job name are watched. **\*ALL** for the job user name is considered to be a generic job specification because it will watch all jobs that meet the job name qualifier that you specified.

*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. **\*ALL** for the job number is considered to be a generic job specification because it will watch 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 or a generic user 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.

### 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.

#### \*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 field specified for element 4, the watch condition is considered to be true. This text is case sensitive. If **\*ALL** is specified in the LIC log compare against field, the LIC log fields which will be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile 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.

#### **Element 4: Compare against**

Specify which part of the LIC log the comparison data specified for element 3 is to be compared against.

**\*ALL** The LIC log comparison data will be compared against all the fields described below.

#### **\*TDENBR**

The LIC log comparison data will be compared against the number of the task dispatching element (TDE) which requested the LIC log entry.

#### **\*TASKNAME**

The LIC log comparison data will be compared against the name of the task which requested the LIC log entry. Task name is blank (hex 40s) if the LIC log entry is not requested by a task.

#### **\*SVRTYPE**

The LIC log comparison data will be compared against the type of server that requested the LIC log entry. Server type is blank (hex 40s) if the LIC log entry is not requested by a server.

#### **\*JOBNAME**

The LIC log comparison data will be compared against the name of the job which requested the LIC log entry. LIC job name is blank (hex 40s) if the LIC log entry is not requested by a job.

#### **\*JOBUSR**

The LIC log comparison data will be compared against the user name of the job which requested the LIC log entry. LIC user name is blank (hex 40s) if the LIC log entry is not requested by a job.

#### **\*JOBNBR**

The LIC log comparison data will be compared against the job number (000001-999999) to further qualify the job name and user name of the job which requested the LIC log entry. LIC job number is blank (hex 40s) if the LIC log entry is not requested by a job.

#### **\*THDID**

The LIC log comparison data will be compared against the thread which requested the LIC log entry. Thread identifier is binary zeros if the LIC log entry is not requested by a thread.

#### **\*EXCPID**

The LIC log comparison data will be compared against the exception that caused the LIC log entry to be requested. This is a 2-byte hexadecimal field formed by concatenating to the high-order 1-byte exception group number a low-order 1-byte exception subtype number. Exception identifier is binary zeros if the LIC log entry is not requested as a result of an exception.

#### **\*MODNAME**

The LIC log comparison data will be compared against the LIC module name which requested the LIC log entry. If the module name is greater than 64 characters, the LIC module name is truncated to 64 characters.

#### **\*MODRUNAME**

The LIC log comparison data will be compared against the LIC module replacement unit name. LIC module RU name is always in upper case EBCDIC.

#### **\*MODEPNAME**

The LIC log comparison data will be compared against the name of the entry point which requested the LIC log entry. If the entry point name is greater than 128 characters, the LIC module entry point name is truncated to 128 characters.



#### **\*MODOFFSET**

The LIC log comparison data will be compared against the byte offset into the LIC module text which requested the LIC log entry.

#### **\*MODTSP**

The LIC log comparison data will be compared against the timestamp of when the LIC module was compiled. The format for this field is the system time-stamp format.

Top

---

## **Watch for PAL entry (WCHPAL)**

Specifies up to five Product Activity Log (PAL) entries which are to be watched for. When the watched for PAL occurs, the watch exit program is called.

### **Single values**

#### **\*NONE**

No PAL entries will be watched for.

### **Other values (up to 5 repetitions)**

#### **Element 1: System reference code**

**\*ALL** Any system reference code will be considered to be a match.

#### *character-value*

Specify the system reference code (SRC) to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the eight-digit code. A question mark is a wildcard character that will match any digit in that position. Up to seven wildcard characters can be specified. You can also specify a generic SRC that 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 SRC specifies all PAL entries with system reference codes that begin with the generic prefix.

#### **Element 2: Comparison data**

Specify comparison data to be used if a PAL entry matching the specified system reference code occurs. If the field specified in element 3 matches the specified text, the watched for condition is true. If the field specified in element 3 does not match the specified text, the watch function just continues.

#### **\*NONE**

No comparison data is specified. If a PAL entry matching the specified system reference code occurs, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the field specified in element 3 of the watched for PAL entry. This text is case sensitive.

You can specify question mark (?) and asterisk (\*) wildcard characters in the text string. A question mark is a single-character wildcard and will match any character in the same position. For example, '??123' will match any value that is five characters long and ends with '123'. Multiple question mark wildcard characters can be specified for the comparison data value.

An asterisk is a multiple-character wildcard character. You can specify a single asterisk wildcard character at the end of the comparison data value. For example, 'ABC\*' will match any value that begins with the letters 'ABC'.

### Element 3: Compare against

Specify which part of the PAL entry the comparison data specified for element 2 is to be compared against.

#### \*RSCNAME

The comparison data will be compared against the name of the physical device that has the entry in the log. A resource name is assigned at first by the system, but may have been changed to a new value by a user.

#### \*RSCTYPE

The comparison data will be compared against the number or word used to identify a product.

#### \*RSCMODEL

The comparison data will be compared against the numbers or letters used to identify the feature level of a product with a given type.

Top

---

## Run priority (RUNPTY)

Specifies the priority of the job where the watch session work will be run.

25 A job priority of 25 will be used.

**1-99** Specify the run priority of the job. For more information on job run priority, refer to the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

Top

---

## Examples

### Example 1: Start a Watch on Your Job

```
STRWCH  SSNID(OWN_JOB) WCHPGM(MYLIB/MYPGM)
        WCHMSG((CPF0001)) WCHMSGQ((*JOBLOG))
```

This command starts the watch session named "OWN\_JOB" to start watching for CPF0001 message to occur on the job that called the STRWCH command. When CPF0001 message is sent to the current job log, MYPGM program in MYLIB library will be called to be notified of the event.

### Example 2: Start a Watch for a Message Specifying a Run Priority

```
STRWCH  SSNID(*GEN) WCHPGM(MYLIB/EXTPGM)
        WCHMSG((CPF1804))
        WCHMSGQ((*SYSOPR) (*JOBLOG))
        WCHJOB((*ALL/MYUSER/MYJOBNAME))
        RUNPTY(10)
```

This command starts a watch session to call MYLIB/EXTPGM user exit program when CPF1804 message is found on the System Operator message queue or within the \*ALL/MYUSER/MYJOBNAME job log. A unique watch session identifier will be generated. The session identifier will be returned in the message data of CPC3901 completion message sent after the watch session starts successfully. The job where the user exit program will be called will run with a run priority of 10.

### Example 3: Start a Watch for a Message Specifying Comparison Data

```
STRWCH  SSNID(FRMPGM) WCHPGM(MYLIB/EXTPGM)
        WCHMSG((CPC3922 QSCSWCH *FROMPGM))
        WCHMSGQ((*HSTLOG))
```

This command starts a watch session to call MYLIB/EXTPGM user exit program when CPC3922 message is sent to message queue QHST in library QSYS by QSCSWCH program.

### Example 4: Start a Watch for a LIC Log Entry

```
STRWCH  SSNID(LICLOGSSN) WCHPGM(*LIBL/EXTPGM)
        WCHLICLOG(('99??' 9932 MYJOBNAME))
```

This command starts LICLOGSSN to watch for a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 generated on the system. Also, the LIC log information should contain the text "MYJOBNAME". The first match of EXTPGM program found in the library list will be called notifying that the event occurred.

### Example 5: Start a Watch for a PAL Entry and Call Exit Program at Start and End Times

```
STRWCH  SSNID(PALSSN)
        WCHPGM(USRLIB/USRPGM)
        CALLWCHPGM(*STRWCH *ENDWCH)
        WCHPAL((B600512? MYRSC *RSCNAME))
```

This command starts PALSSN to watch for a Product Activity Log (PAL) entry that has a system reference code starting with B600512 generated on the system. Also, the PAL resource name contains the text "MYRSC". The program USRLIB/USRPGM will be called notifying that the event occurred, but it will also be called before starting watching for any event and when the watch session is ending.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF0006

Errors occurred in command.

#### CPF2401

Not authorized to library &1.

#### CPF2403

Message queue &1 in &2 not found.

#### CPF2408

Not authorized to message queue &1.

#### CPF39E3

Session ID &1 already exists.

#### CPF39E4

Specify Watch for message, LIC log entry or PAL entry.

**CPF39E5**  
No active jobs found, watch session not started.

**CPF39E6**  
The user does not have the required authority.

**CPF39E7**  
Invalid session identifier.

**CPF39E8**  
Not enough authority to watch operations.

**CPF39E9**  
\*JOBCTL special authority required.

**CPF3958**  
Not authorized to use program &1 in library &2.

**CPF39D0**  
Watch for event function cannot start. Reason code &1.

**CPF39D1**  
Limit exceeded for jobs watching for events.

**CPF3C20**  
Error found by program &1.

**CPF9811**  
Program &1 in library &2 not found.

Top

---

## Subroutine (SUBR)

### Where allowed to run:

- Batch program (\*BPGM)
- Interactive program (\*IPGM)

Threadsafe: Yes

Parameters  
Examples  
Error messages

The Subroutine (SUBR) command is used in a CL program or ILE CL procedure, along with the End Subroutine (ENDSUBR) command, to delimit the group of commands that define a **subroutine**. The SUBR command defines the start of a subroutine. Subroutines may not be nested; therefore, an ENDSUBR command must end the current subroutine before the next SUBR command starts a new subroutine. A label specified on a SUBR command, or on a null command preceding the SUBR command, can be used as a target for a Go To (GOTO) command defined in the same subroutine. This GOTO command, when run, will pass control to the first executable statement in the subroutine.

### Restrictions:

- This command is valid only within a CL program or ILE CL procedure.
- The subroutine must be defined within a procedure, and must be located at the end of the procedure.

Top

---

## Parameters

Keyword	Description	Choices	Notes
SUBR	Subroutine	<i>Simple name</i>	Required, Positional 1

Top

---

## Subroutine (SUBR)

Specifies the subroutine name. This is the name that must be specified on a Call Subroutine (CALLSUBR) command.

This is a required parameter.

### *simple-name*

Specify the name of the subroutine. A CL variable name cannot be used to specify the subroutine name.

Top

---

## Examples

```
PGM
:
SUBR SUBR1
:
ENDSUBR
ENDPGM
```

This SUBR command defines the start of a subroutine named SUBR1.

[Top](#)

---

## Error messages

None

[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)

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

**Threadsafe:** No

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.

### 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.

### 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 and the job queue is not in an independent auxiliary storage pool (ASP). 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.

[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.

### Restrictions:

1. To use this command, you must have use (\*USE) authority to the initial group program and execute (\*EXECUTE) authority to the library that contains that program.

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.

### 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. This restriction only applies if the job being transferred is an interactive job.
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.

### 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.

6. When the new routing step is started, the current user must have use (\*USE) authority to the subsystem description for the subsystem in which the routing step runs.

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	*SRC, *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.

**\*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.

---

## 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.

---

## 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.

---

## 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.

---

## 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. TRCTCP RTE (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 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 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 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**

```
TRCTCP RTE RMTSYS('IP6HOST')
```

This command traces the entire route between the local 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 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) version 4 (IPv4) and version 6 (IPv6), 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 i5/OS through System i 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 **Watched job (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. Job control (\*JOBCTL) special authority is also required if a generic user name is specified for the WCHJOB parameter.
- If you specify a generic user name in the WCHJOB parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Watch any job function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_WATCH\_ANY\_JOB, can also be used to change the list of users that are allowed to start and end watch operations.
- You must have operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the user exit program if specified in **Trace program (TRCPGM)** parameter, and execute (\*EXECUTE) authority to the library where the program is located.
- You must have use (\*USE) authority to the message queues specified in **Watched message queue (WCHMSGQ)** parameter, and use (\*USE) authority to the library where the message queue is located.

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, *IPV4, *IPV6	Optional

Keyword	Description	Choices	Notes
TRCFULL	Trace full	<b>*WRAP</b> , <b>*STOPTRC</b>	Optional
TRCTBL	Trace table name	<i>Character value</i> , <b>*GEN</b>	Optional
SIZE	Size	Single values: <b>*MAX</b> , <b>*MIN</b> Other values: <i>Element list</i>	Optional
	Element 1: Number of units	1-998000, <b>16000</b>	
	Element 2: Unit of measure	<b>*KB</b> , <b>*MB</b>	
FMTDTA	Trace data to be formatted	72-99999, <b>*CALC</b>	Optional
CCSID	Coded character set identifier	1-65533, <b>*EBCDIC</b> , <b>*ASCII</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>	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>	
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>	
WCHMSG	Watch for message	Single values: <b>*NONE</b> 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> , <b>*NONE</b>	
	Element 3: Compare against	<b>*MSGDTA</b> , <b>*FROMPGM</b> , <b>*TOPGM</b>	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional
	Element 1: Message queue	Single values: <b>*SYSOPR</b> , <b>*JOBLOG</b> , <b>*HSTLOG</b> 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
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Generic name, name</i>	
	Qualifier 3: Number	000001-999999, <b>*ALL</b>	

Keyword	Description	Choices	Notes
WCHLICLOG	Watch for LIC log entry	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Major code	<i>Character value</i> , <b>*ALL</b>	
	Element 2: Minor code	<i>Character value</i> , <b>*ALL</b>	
	Element 3: Comparison data	<i>Character value</i> , <b>*NONE</b>	
	Element 4: Compare against	<b>*ALL</b> , <b>*TDENBR</b> , <b>*TASKNAME</b> , <b>*SVRTYPE</b> , <b>*JOBNAME</b> , <b>*JOBUSR</b> , <b>*JOBNBR</b> , <b>*THDID</b> , <b>*EXCPID</b> , <b>*MODNAME</b> , <b>*MODRUNAME</b> , <b>*MODEPNAME</b> , <b>*MODOFFSET</b> , <b>*MODTSP</b>	
WCHPAL	Watch for PAL entry	Single values: <b>*NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: System reference code	<i>Character value</i> , <b>*ALL</b>	
	Element 2: Comparison data	<i>Character value</i> , <b>*NONE</b>	
	Element 3: Compare against	<b>*RSCNAME</b> , <b>*RSCTYPE</b> , <b>*RSCMODEL</b>	
WCHTIMO	Length of time to watch	1-43200, <b>*NOMAX</b>	Optional
TRCPGM	Trace program	Single values: <b>*NONE</b> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Trace program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <b>*LIBL</b>	
TRCPGMITV	Time interval	1-9999, <b>*NONE</b>	Optional
RUNPTY	Run priority	1-99, <b>25</b>	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 QPCSMPT 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 QPCSMPT 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 QPCSMPT 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 version 4 and version 6) data.

**\*SSL** Trace SSL (Secure Sockets Layer) connection data.

**\*IPV4** Trace IPv4 (internet protocol version 4) data.

**\*IPV6** Trace IPv6 (internet protocol version 6) 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.

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 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 (TCPDATA)

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, the "From program" or the "To program" includes the specified text, the watched for condition is true. If the message data, the "From program" or the "To program" 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, the "From program" or the "To program" of the watched for message. This text is case sensitive and can be quoted in order to specify imbedded or trailing blanks.

### Element 3: Compare against

Specify which part of the message the comparison data specified for element 2 is to be compared against.

#### \*MSGDATA

The comparison data will be compared against the message replacement data.

#### \*FROMPGM

The comparison data will be compared against the name of the program sending the message, or the name of the ILE program that contains the procedure sending the message.

#### \*TOPGM

The comparison data will be compared against the name of the program the message was sent to, or the name of the ILE program that contains the procedure the message was sent to.

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 message queue in library QSYS).

#### 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

##### *generic-name*

Specify the generic name of the user 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 user name specifies all jobs with the specified job name and with user names that begin with the generic prefix.

*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 or a generic user 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 field specified for element 4, the watch condition is considered to be true. This text is case sensitive. If \*ALL is specified in the LIC log compare against field, the LIC log fields which will be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile 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.

### Element 4: Compare against

Specify which part of the LIC log the comparison data specified for element 3 is to be compared against.

**\*ALL** The LIC log comparison data will be compared against all the fields described below.

#### **\*TDENBR**

The LIC log comparison data will be compared against the number of the task dispatching element (TDE) which requested the LIC log entry.

#### **\*TASKNAME**

The LIC log comparison data will be compared against the name of the task which requested the LIC log entry. Task name is blank (hex 40s) if the LIC log entry is not requested by a task.

**\*SVRTYPE**

The LIC log comparison data will be compared against the type of server that requested the LIC log entry. Server type is blank (hex 40s) if the LIC log entry is not requested by a server.

**\*JOBNAME**

The LIC log comparison data will be compared against the name of the job which requested the LIC log entry. LIC job name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBUSR**

The LIC log comparison data will be compared against the user name of the job which requested the LIC log entry. LIC user name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBNBR**

The LIC log comparison data will be compared against the job number (000001-999999) to further qualify the job name and user name of the job which requested the LIC log entry. LIC job number is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*THDID**

The LIC log comparison data will be compared against the thread which requested the LIC log entry. Thread identifier is binary zeros if the LIC log entry is not requested by a thread.

**\*EXCPID**

The LIC log comparison data will be compared against the exception that caused the LIC log entry to be requested. This is a 2-byte hexadecimal field formed by concatenating to the high-order 1-byte exception group number a low-order 1-byte exception subtype number. Exception identifier is binary zeros if the LIC log entry is not requested as a result of an exception.

**\*MODNAME**

The LIC log comparison data will be compared against the LIC module name which requested the LIC log entry. If the module name is greater than 64 characters, the LIC module name is truncated to 64 characters.

**\*MODRUNAME**

The LIC log comparison data will be compared against the LIC module replacement unit name. LIC module RU name is always in upper case EBCDIC.

**\*MODEPNAME**

The LIC log comparison data will be compared against the name of the entry point which requested the LIC log entry. If the entry point name is greater than 128 characters, the LIC module entry point name is truncated to 128 characters.

**\*MODOFFSET**

The LIC log comparison data will be compared against the byte offset into the LIC module text which requested the LIC log entry.

**\*MODTSP**

The LIC log comparison data will be compared against the timestamp of when the LIC module was compiled. The format for this field is the system time-stamp format.

Top

---

**Watch for PAL entry (WCHPAL)**

Specifies up to five Product Activity Log (PAL) entries which are to be watched for. When the watched for PAL occurs, the trace exit program is called; if no trace exit program is defined, the trace stops.

**Single values****\*NONE**

No PAL entries will be watched for.

## Other values (up to 5 repetitions)

### Element 1: System reference code

**\*ALL** Any system reference code will be considered to be a match.

#### *character-value*

Specify the system reference code (SRC) to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the eight-digit code. A question mark is a wildcard character that will match any digit in that position. Up to seven wildcard characters can be specified. You can also specify a generic SRC that 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 SRC specifies all PAL entries with system reference codes that begin with the generic prefix.

### Element 2: Comparison data

Specify comparison data to be used if a PAL entry matching the specified system reference code occurs. If the field specified in element 3 matches the specified text, the watched for condition is true. If the field specified in element 3 does not match the specified text, the watch function just continues.

#### \*NONE

No comparison data is specified. If a PAL entry matching the specified system reference code occurs, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the field specified in element 3 of the watched for PAL entry. This text is case sensitive.

You can specify question mark (?) and asterisk (\*) wildcard characters in the text string. A question mark is a single-character wildcard and will match any character in the same position. For example, '??123' will match any value that is five characters long and ends with '123'. Multiple question mark wildcard characters can be specified for the comparison data value.

An asterisk is a multiple-character wildcard character. You can specify a single asterisk wildcard character at the end of the comparison data value. For example, 'ABC\*' will match any value that begins with the letters 'ABC'.

### Element 3: Compare against

Specify which part of the PAL entry the comparison data specified for element 2 is to be compared against.

#### \*RSCNAME

The comparison data will be compared against the name of the physical device that has the entry in the log. A resource name is assigned at first by the system, but may have been changed to a new value by a user.

#### \*RSCTYPE

The comparison data will be compared against the number or word used to identify a product.

#### \*RSCMODEL

The comparison data will be compared against the numbers or letters used to identify the feature level of a product with a given type.

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 or a Product Activity Log (PAL) 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 history log.

### \*NOMAX

There is no time limit for watching for a particular message or LIC log entry or PAL 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, or a match of a Product Activity Log (PAL) entry specified for the WCHPAL 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.

**\*PAL** A match on a PAL and any associated comparison data specified on the WCHPAL parameter occurred.



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).

Allowed values for the "Comparison data" parameter when \*PAL is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch information
4 4	CHAR(8)	System reference code
12 C	CHAR(10)	Device name
22 16	CHAR(4)	Device type
26 1A	CHAR(4)	Model
30 1E	CHAR(15)	Serial number
45 2D	CHAR(10)	Resource name
55 37	CHAR(8)	Log identifier
63 3F	CHAR(8)	PAL timestamp

71	47	CHAR(4)	Reference code
75	4B	CHAR(8)	Secondary code
83	53	CHAR(8)	Table identifier
91	5B	CHAR(1)	Reserved
92	5C	BINARY(4)	Sequence
96	60	BINARY(4)	Offset to comparison data
100	64	BINARY(4)	Length of comparison data
104	68	CHAR(10)	PAL compare against
*	*	CHAR(*)	PAL comparison data

For more information on the trace exit program interface, refer to the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

### Single values

#### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry or Product Activity Log (PAL) 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

## Run priority (RUNPTY)

Specifies the priority of the job where the watch session work will be run.

**25** A job priority of 25 will be used.

**1-99** Specify the run priority of the job. For more information on job run priority, refer to the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

Top

---

## 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) version 4 (IPv4) and version 6 (IPv6) 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) version 4 (IPv4) and version 6 (IPv6) level and stores the trace data in the USER trace table.

### Example 7: Specifying a Trace Table Size in Megabytes

```
TRCCNN SET(*ON) TRCTYPE(*IPV6) SIZE(20000 *MB)
```

This command starts tracing for connections at the internet protocol version 6 (IPv6) 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(*IPV4) WCHMSG((MCH2804))
WCHMSGQ((*SYSOPR) (*JOBLOG))
WCHJOB((*ALL/MYUSER/MYJOBNAME))
TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts tracing for connections at the internet protocol version 4 (IPv4) 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(*IPV4)
WCHLICLOG(('99??' 9932 MYJOBNAME))
WCHTIMO(*NOMAX)
```

This command starts tracing for connections at the internet protocol version 4 (IPv4) 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 TRCCNN command is issued manually to end the trace.

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	* <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 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.

Top

---

## 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.

Top

---

## 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.

---

## 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.

---

## 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 i5/OS through System i 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 **Watched job (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. Job control (\*JOBCTL) special authority is also required if a generic user name is specified for the WCHJOB parameter.
4. If you specify a generic user name in the WCHJOB parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Watch any job function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_WATCH\_ANY\_JOB, can also be used to change the list of users that are allowed to start and end watch operations.
5. You must have operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the user exit program if specified in **Trace program (TRCPGM)** parameter, and execute (\*EXECUTE) authority to the library where the program is located.
6. You must have use (\*USE) authority to the message queues specified in **Watched message queue (WCHMSGQ)** parameter, and use (\*USE) authority to the library where the message queue is located.

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, *ALTTCP, *APPCOVRTCP, *APPCPS, *APPNALL, *APPNCPM, *APPNCPPS, *APPNDS, *APPNLM, *APPNMST, *APPNTRS, *AUTMGT, *AUXSTGALL, *AUXSTGMGT, *BSSMGT, *CCiom, *CLUE, *CMNACCMTH, *CMNTRC, *CMTMGT, *COMMON, *CRPSRV, *CSTALL, *CSTCMN, *CTXMGT, *DBGINT, *DBMGT, *DLUR, *DSPPASTHR, *EREP, *ERRLOG, *EVTMGT, *EXCMGT, *FRCA, *HDWRSC, *HEAPMGT, *IDXMG, *IFS, *IPCF, *ISCSI, *ISDN, *JRNMG, *LNKTST, *LODDMP, *MCHOBS, *MITFMALL, *MITFMIVAL, *MITFMHEAP, *MITFMINT, *MITFMMI, *MITFMSTG, *MODMGT, *MOD2, *MSCP, *MSMALL, *MSMDTL, *M36ALL, *M36ASC, *M36BSC, *M36CSP, *M36DKT, *M36DSK, *M36ILAN, *M36PRT, *M36SDLC, *M36TAP, *M36TRN, *M36WS, *M36X25, *NTBTCP, *OPC, *PASE, *PFRCOLSRV, *PGMBND, *PGMMGT, *PORTUTIL, *PRCMGT, *PRCTBL, *PSEUDOTERM, *PWRMGT, *QMG, *QSMGT, *RCYMG, *RMTSPT, *RSCMGT, *SCK, *SCKASCIO, *SCKNET, *SCKOTHER, *SCKOVRMPTN, *SCKOVRPEC, *SCKRSLV, *SCKSEL, *SCKSSL, *SCKSTDIO, *SIG, *SMBSVR, *SPCOBJMGT, *SRCSINK, *STGMGTALL, *STM, *SYNCGMT, *SYSCALL, *SYSJRNMG, *TCPIP, *TL, *TRXMG, *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, <u>0.5</u>	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>*INET6</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	
	Element 3: Compare against	*MSGDTA, *FROMPGM, *TOPGM	
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	Generic name, 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	
	Element 4: Compare against	*ALL, *TDENBR, *TASKNAME, *SVRTYPE, *JOBNAME, *JOBUSR, *JOBNBR, *THDID, *EXCPID, *MODNAME, *MODRUNAME, *MODEPNAME, *MODOFFSET, *MODTSP	
WCHPAL	Watch for PAL entry	Single values: *NONE Other values (up to 5 repetitions): Element list	Optional
	Element 1: System reference code	Character value, *ALL	
	Element 2: Comparison data	Character value, *NONE	
	Element 3: Compare against	*RSCNAME, *RSCTYPE, *RSCMODEL	
WCHTIMO	Length of time to watch	1-43200, *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
RUNPTY	Run priority	1-99, 25	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 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
Alternate TCP/IP stack	016900	*ALTTCP
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 transport	012507	*APPNMST
APPN topology and routing services	012503	*APPNTRS
Authority management	010900	*AUTMGT
Auxiliary storage management -- detailedd	011104	*AUXSTGALL
Auxiliary storage management	011101	*AUXSTGMGT
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

Internet SCSI host adapter	017000	*ISCSI
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	*PGBND
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	*QMGT
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 PĒC	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	*SYNCGMT
MI system call	015200	*SYSCALL
System journal management	014100	*SYSJRNMG
Transmission Control Protocol/ Internet Protocol (TCP/IP)	013800	*TCPIP
Transport Layer Interface	016000	*TLI



Transaction Management	016800	*TRXMG
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 topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/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) or TRCTYPE(\*SMBSVR) or TRCTYPE(015800) or TRCTYPE(\*FRCA) or TRCTYPE(016600) 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\_CCSID address families.

**\*INET6**

Enable trace for AF\_INET6 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.

---

## 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.

Top

---

## 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.

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, the "From program" or the "To program" includes the specified text, the watched for condition is true. If the message data, the "From program" or the "To program" 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, the "From program" or the "To program" of the watched for message. This text is case sensitive and can be quoted in order to specify imbedded or trailing blanks.

## Element 3: Compare against

Specify which part of the message the comparison data specified for element 2 is to be compared against.

### \*MSGDATA

The comparison data will be compared against the message replacement data.

### \*FROMPGM

The comparison data will be compared against the name of the program sending the message, or the name of the ILE program that contains the procedure sending the message.

### \*TOPGM

The comparison data will be compared against the name of the program the message was sent to, or the name of the ILE program that contains the procedure the message was sent to.

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 message queue in library QSYS).

### **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**

##### *generic-name*

Specify the generic name of the user 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 user name specifies all jobs with the specified job name and with user names that begin with the generic prefix.

*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 or a generic user 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 field specified for element 4, the watch condition is considered to be true. This text is case sensitive. If \*ALL is specified in the LIC log compare



against field, the LIC log fields which will be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile 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.

#### **Element 4: Compare against**

Specify which part of the LIC log the comparison data specified for element 3 is to be compared against.

**\*ALL** The LIC log comparison data will be compared against all the fields described below.

##### **\*TDENBR**

The LIC log comparison data will be compared against the number of the task dispatching element (TDE) which requested the LIC log entry.

##### **\*TASKNAME**

The LIC log comparison data will be compared against the name of the task which requested the LIC log entry. Task name is blank (hex 40s) if the LIC log entry is not requested by a task.

##### **\*SVRTYPE**

The LIC log comparison data will be compared against the type of server that requested the LIC log entry. Server type is blank (hex 40s) if the LIC log entry is not requested by a server.

##### **\*JOBNAME**

The LIC log comparison data will be compared against the name of the job which requested the LIC log entry. LIC job name is blank (hex 40s) if the LIC log entry is not requested by a job.

##### **\*JOBUSR**

The LIC log comparison data will be compared against the user name of the job which requested the LIC log entry. LIC user name is blank (hex 40s) if the LIC log entry is not requested by a job.

##### **\*JOBNBR**

The LIC log comparison data will be compared against the job number (000001-999999) to further qualify the job name and user name of the job which requested the LIC log entry. LIC job number is blank (hex 40s) if the LIC log entry is not requested by a job.

##### **\*THDID**

The LIC log comparison data will be compared against the thread which requested the LIC log entry. Thread identifier is binary zeros if the LIC log entry is not requested by a thread.

##### **\*EXCPID**

The LIC log comparison data will be compared against the exception that caused the LIC log entry to be requested. This is a 2-byte hexadecimal field formed by concatenating to the high-order 1-byte exception group number a low-order 1-byte exception subtype number. Exception identifier is binary zeros if the LIC log entry is not requested as a result of an exception.

##### **\*MODNAME**

The LIC log comparison data will be compared against the LIC module name which requested the LIC log entry. If the module name is greater than 64 characters, the LIC module name is truncated to 64 characters.

##### **\*MODRUNAME**

The LIC log comparison data will be compared against the LIC module replacement unit name. LIC module RU name is always in upper case EBCDIC.

**\*MODEPNAME**

The LIC log comparison data will be compared against the name of the entry point which requested the LIC log entry. If the entry point name is greater than 128 characters, the LIC module entry point name is truncated to 128 characters.

**\*MODOFFSET**

The LIC log comparison data will be compared against the byte offset into the LIC module text which requested the LIC log entry.

**\*MODTSP**

The LIC log comparison data will be compared against the timestamp of when the LIC module was compiled. The format for this field is the system time-stamp format.

Top

## Watch for PAL entry (WCHPAL)

Specifies up to five Product Activity Log (PAL) entries which are to be watched for. When the watched for PAL occurs, the trace exit program is called; if no trace exit program is defined, the trace stops.

### Single values

**\*NONE**

No PAL entries will be watched for.

### Other values (up to 5 repetitions)

#### Element 1: System reference code

**\*ALL** Any system reference code will be considered to be a match.

#### *character-value*

Specify the system reference code (SRC) to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the eight-digit code. A question mark is a wildcard character that will match any digit in that position. Up to seven wildcard characters can be specified. You can also specify a generic SRC that 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 SRC specifies all PAL entries with system reference codes that begin with the generic prefix.

#### Element 2: Comparison data

Specify comparison data to be used if a PAL entry matching the specified system reference code occurs. If the field specified in element 3 matches the specified text, the watched for condition is true. If the field specified in element 3 does not match the specified text, the watch function just continues.

**\*NONE**

No comparison data is specified. If a PAL entry matching the specified system reference code occurs, the watched for condition is true.

#### *character-value*

Specify the text string used to compare against the field specified in element 3 of the watched for PAL entry. This text is case sensitive.

You can specify question mark (?) and asterisk (\*) wildcard characters in the text string. A question mark is a single-character wildcard and will match any character in the same position.

For example, '??123' will match any value that is five characters long and ends with '123'. Multiple question mark wildcard characters can be specified for the comparison data value.

An asterisk is a multiple-character wildcard character. You can specify a single asterisk wildcard character at the end of the comparison data value. For example, 'ABC\*' will match any value that begins with the letters 'ABC'.

### Element 3: Compare against

Specify which part of the PAL entry the comparison data specified for element 2 is to be compared against.

#### \*RSCNAME

The comparison data will be compared against the name of the physical device that has the entry in the log. A resource name is assigned at first by the system, but may have been changed to a new value by a user.

#### \*RSCTYPE

The comparison data will be compared against the number or word used to identify a product.

#### \*RSCMODEL

The comparison data will be compared against the numbers or letters used to identify the feature level of a product with a given type.

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 or a Product Activity Log (PAL) 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 history log.

#### \*NOMAX

There is no time limit for watching for a particular message or LIC log entry or PAL 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, or a match of a Product Activity Log (PAL) entry specified for the WCHPAL 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.

**\*PAL** A match on a PAL and any associated comparison data specified on the WCHPAL parameter occurred.

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).

Allowed values for the "Comparison data" parameter when \*PAL is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch information
4 4	CHAR(8)	System reference code
12 C	CHAR(10)	Device name
22 16	CHAR(4)	Device type
26 1A	CHAR(4)	Model
30 1E	CHAR(15)	Serial number
45 2D	CHAR(10)	Resource name
55 37	CHAR(8)	Log identifier
63 3F	CHAR(8)	PAL timestamp
71 47	CHAR(4)	Reference code
75 4B	CHAR(8)	Secondary code
83 53	CHAR(8)	Table identifier
91 5B	CHAR(1)	Reserved
92 5C	BINARY(4)	Sequence
96 60	BINARY(4)	Offset to comparison data
100 64	BINARY(4)	Length of comparison data
104 68	CHAR(10)	PAL compare against
* *	CHAR(*)	PAL comparison data

For more information on the trace exit program interface, refer to the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

### Single values

#### \*NONE

No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry or Product Activity Log (PAL) 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

---

## Run priority (RUNPTY)

Specifies the priority of the job where the watch session work will be run.

**25** A job priority of 25 will be used.

**1-99** Specify the run priority of the job. For more information on job run priority, refer to the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

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 TRCINT command is issued manually to stop the trace.

#### 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**  
TCPDATA only valid if \*TCPIP or \*SMBSVR or \*FRCA or one or more of the socket TRCTYPE is specified.
- CPD3687**  
SCKDATA 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 or API.

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.

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 TRCJOB command issues under the covers the STRTRC command when SET(\*ON) is specified or the ENDTRC command when SET(\*OFF) or SET(\*END) is specified.

### 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
EXITPGM	Program to call before trace	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Program to call before trace	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	

Keyword	Description	Choices	Notes
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.

Top

---

## 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.)

Top

---

## 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.

Top

---

## Program to call before trace (EXITPGM)

This parameter is no longer used.

Top

---

## Select procedures to trace (SLTPRC)

This parameter is no longer used.

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.

---

## 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 One Thread

```
TRCJOB SET(*ON) SLTTHD(00000001)
```

This command traces only the specified thread of the current job.

---

## 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.

Top



---

## 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.

#### **\*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)  
**Threadsafe:** 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, SMTP APIs, REXEC, 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 i5/OS through System i 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 **Additional traces (ADLTRC)** parameter is specified, you must have all object (\*ALLOBJ) special authority or be authorized to the Trace Any User function of i5/OS through System i 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.

When the **Watched job (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. Job control (\*JOBCTL) special authority is also required if a generic user name is specified for the **Watched job (WCHJOB)** parameter.

If you specify a generic user name in the **Watched job (WCHJOB)** parameter, you must have all object (\*ALLOBJ) special authority, or be authorized to the Watch any job function of Operating System through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_WATCH\_ANY\_JOB, can also be used to change the list of users that are allowed to start and end watch operations.

You must have operational (\*OBJOPR) and execute (\*EXECUTE) authorities to the user exit program if specified in **Trace program (TRCPGM)** parameter, and execute (\*EXECUTE) authority to the library where the program is located.

You must have use (\*USE) authority to the message queues specified in **Watched message queue (WCHMSGQ)** parameter, and use (\*USE) authority to the library where the message queue is located.

A trace cannot be defined to trace all job names and all users.

**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, *POP, *MSF, *SMTPAPI, *REXEC	Required, Positional 1
SET	Trace option setting	*ON, *OFF, *END, *CHK	Optional, Positional 2
MAXSTG	Maximum storage for trace	1-16000, <u>*APP</u>	Optional, Positional 3
TRCFULL	Trace full action	*WRAP, *STOPTRC	Optional, Positional 4
ADLTRC	Additional traces	Single values: *NONE Other values (up to 11 repetitions): *CMNARB, *CMNTRC, *DEVVD, *DEVMGR, *JOB, *SOCKETS, *SRCSINK, *SYSARB, *TCPIP, *USER, *USERDEVVD	Optional, Positional 5
TRCPGM	Trace program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional, Positional 6
	Qualifier 1: Trace program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
TITLE	Trace title	<i>Character value</i> , <u>*DFT</u>	Optional, Positional 7
USER	User profile	<i>Simple name</i>	Optional, Positional 8
MAILADR	Recipient mail address	<i>Character value</i>	Optional, Positional 9
HOST	Recipient host name	<i>Character value</i>	Optional, Positional 10
RMTNETADR	Remote network address	<i>Element list</i>	Optional, Positional 11
	Element 1: Address family	<u>*CALC</u>	
	Element 2: IP address	<i>Character value</i>	
	Element 3: Subnet mask or Prefix length	<u>*HOST</u>	
	Element 4: Port number	1-65535, <u>*ANY</u>	
LCLNETADR	Local network address	<i>Element list</i>	Optional, Positional 12
	Element 1: Address family	<u>*CALC</u> , *UNIX	
	Element 2: IP address or UNIX path	<i>Character value</i>	
	Element 3: Subnet mask or Prefix length	<u>*HOST</u>	
	Element 4: Port number	1-65535, <u>*ANY</u>	
DEVVD	Device description	<i>Generic name, name</i>	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): <i>Character value</i>	Optional, Positional 15
ARGLIST	Argument list	<i>Character value</i>	Optional, Positional 16
VPNSVR	Virtual private network server	Values (up to 2 repetitions): *KEYMGR, *CNNMGR	Optional, Positional 17

Keyword	Description	Choices	Notes
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>*PPALL</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
INSTANCE	Instance	Character value, <u>*DFT</u> , *ALL	Optional, Positional 24
TRCLVL	Trace level	<u>*ERROR</u> , *INFO, *VERBOSE	Optional, Positional 25
PKTTRCPNT	Packet rules trace points	<u>*TRAFFIC</u> , *LOAD	Optional, Positional 26
CFGOBJ	Configuration object	Name	Optional, Positional 27
CFGTYPE	Type	*LIN, *NWI, *NWS	Optional, Positional 28
WCHMSG	Watch for message	Single values: <u>*NONE</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 29
	Element 1: Message identifier	Name	
	Element 2: Comparison data	Character value, <u>*NONE</u>	
	Element 3: Compare against	<u>*MSGDTA</u> , *FROMPGM, *TOPGM	
WCHMSGQ	Watched message queue	Values (up to 3 repetitions): <i>Element list</i>	Optional, Positional 30
	Element 1: Message queue	Single values: <u>*SYSOPR</u> , *JOBLOG, *HSTLOG Other values: <i>Qualified object name</i>	
	Qualifier 1: Message queue	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u>	
WCHJOB	Watched job	Single values: * Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 31
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Generic name, name</i>	
	Qualifier 3: Number	000001-999999, <u>*ALL</u>	
WCHLICLOG	Watch for LIC log entry	Single values: <u>*NONE</u> Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 32
	Element 1: Major code	Character value, *ALL	
	Element 2: Minor code	Character value, *ALL	
	Element 3: Comparison data	Character value, <u>*NONE</u>	
	Element 4: Compare against	<u>*ALL</u> , *TDENBR, *TASKNAME, *SVRTYPE, *JOBNAME, *JOBUSR, *JOBNBR, *THDID, *EXCPID, *MODNAME, *MODRUNAME, *MODEPNAME, *MODOFFSET, *MODTSP	

Keyword	Description	Choices	Notes
WCHPAL	Watch for PAL entry	Single values: * <b>NONE</b> Other values (up to 5 repetitions): <i>Element list</i>	Optional, Positional 33
	Element 1: System reference code	<i>Character value</i> , * <b>ALL</b>	
	Element 2: Comparison data	<i>Character value</i> , * <b>NONE</b>	
	Element 3: Compare against	* <b>RSCNAME</b> , * <b>RSCTYPE</b> , * <b>RSCMODEL</b>	
WCHTIMO	Length of time to watch	1-43200, * <b>NOMAX</b>	Optional, Positional 34
TRCPGMITV	Time interval	1-9999, * <b>NONE</b>	Optional, Positional 35
RUNPTY	Run priority	1-99, <u>25</u>	Optional, Positional 36
JOB	Jobs	Single values: * Other values (up to 16 repetitions): <i>Element list</i>	Optional, Positional 37
	Element 1: Job name	<i>Qualified job name</i>	
	Qualifier 1: Job name	<i>Generic name, name</i> , * <b>ALL</b>	
	Qualifier 2: User	<i>Generic name, name</i> , * <b>ALL</b>	
	Qualifier 3: Number	000001-999999, * <b>ALL</b>	

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).

**\*MSF** Specifies tracing for the Simple Mail Transport Protocol (SMTP) Mail Server Framework (MSF) exit programs.

**\*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).

**\*POP** Specifies tracing for the Post Office Protocol (POP) server.

**\*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.

**\*SMTPAPI**

Specifies tracing for job(s) that invoke SMTP APIs.

**\*REXEC**

Specifies tracing for job(s) that invoke the REXEC application.

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 - 16000K bytes per job
- \*SMTPSVR - 16000K 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 - 16000K bytes per job
- \*DIRSRV - 16000K bytes per job
- \*POP - 16000K bytes per job
- \*MSF - 16000K bytes per job
- \*SMTPAPI - 16000K bytes per job
- \*REXEC - 16000K bytes per job

**1-300000**

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.

**Note:** Specifying \*CMNARB, \*DEVVD, \*DEVMGR, \*JOB, \*SYSARB, \*USER or \*USERDEVVD is only valid when tracing the \*TELNET application.

### Single values

#### **\*NONE**

No additional trace will be included.

### Other values (up to 11 repetitions)

#### **\*CMNARB**

A job-level trace of all communication arbiter jobs will be started. For more information about communication arbiters, use the Search function in the i5/OS Information Center at <http://publib.boulder.ibm.com/iseres>.

#### **\*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.

#### **\*DEVVD**

All jobs associated with the device specified for the Device description (DEVVD) parameter will be traced.

#### **\*DEVMGR**

All information related with Telnet device management functions will be traced.

**\*JOB** All jobs specified in job parameter (JOB) will be traced.

#### **\*SOCKETS**

A single Sockets 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.

#### **\*SYSARB**

A job-level trace of all system arbiter jobs will be started. For more information about system arbiters, use the Search function in the i5/OS Information Center at <http://publib.boulder.ibm.com/iseres>.

#### **\*TCPIP**

A single TCP/IP component trace will be included in the trace information for the specified application.

#### **\*USER**

All information related with the user profile parameter (USER) will be traced.

#### **\*USERDEVVD**

All user jobs associated with the device specified for the Device description (DEVVD) parameter will be traced.

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, WCHLICLOG or WCHPAL 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, or a match of a Product Activity Log (PAL) entry specified on WCHPAL 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.
- \*PAL, A match on a PAL and any associated comparison data specified on the WCHPAL parameter occurred.

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).

Allowed values for the "Comparison data" parameter when \*PAL is specified for the "Trace option setting" parameter will be the following structure:

OFFSET	TYPE	FIELD
Dec Hex		
0 0	BINARY(4)	Length of watch information
4 4	CHAR(8)	System reference code
12 C	CHAR(10)	Device name
22 16	CHAR(4)	Device type
26 1A	CHAR(4)	Model
30 1E	CHAR(15)	Serial number
45 2D	CHAR(10)	Resource name
55 37	CHAR(8)	Log identifier
63 3F	CHAR(8)	PAL timestamp
71 47	CHAR(4)	Reference code
75 4B	CHAR(8)	Secondary code
83 53	CHAR(8)	Table identifier

91	5B	CHAR(1)	Reserved
92	5C	BINARY(4)	Sequence
96	60	BINARY(4)	Offset to comparison data
100	64	BINARY(4)	Length of comparison data
104	68	CHAR(10)	PAL compare against
*	*	CHAR(*)	PAL comparison data

For more information on the trace exit program interface, refer to the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

## Single values

### \*NONE

No user supplied trace program will be called. If a watched for message or Licensed Internal Code (LIC) log entry or Product Activity Log (PAL) 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, or when APP(\*TELNET) and ADLTRC(\*USER) are 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

#### \*CALC

The filter is AF\_INET for IPv4 addresses. The filter is AF\_INET6 for IPv6 addresses.

### Element 2: IP address

#### *character-value*

Specify the remote TCP/IP address for which trace information is to be collected. Specify either an IPv4 or an IPv6 address.

### Element 3: Subnet mask or Prefix length

#### \*HOST

Tracing will be done for only the IP address specified as the second element of this parameter. For IPv4, this will specify a subnet mask of 255.255.255.255. For IPv6 this will specify a prefix length of /128.

#### *character-value*

Specify the IPv4 subnet mask or IPv6 prefix length 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.

---

## 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

#### \*CALC

The filter is AF\_INET for IPv4 addresses. The filter is AF\_INET6 for IPv6 addresses.

#### \*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 \*CALC is specified for element 1 of this parameter, specify the local TCP/IP address for which trace information is to be collected. Specify either an IPv4 or an IPv6 address.

When \*UNIX is specified for element 1 of this parameter, specify the UNIX path for which trace information is to be collected. Note that an UNIX-path can be entered for only APP(\*DDM) or APP(host server).

### Element 3: Subnet mask or Prefix length

#### \*HOST

Tracing will be done for only the IP address specified as the second element of this parameter. For IPv4, this will specify a subnet mask of 255.255.255.255. For IPv6 this will specify a prefix length of /128.

#### *character-value*

Specify the IPv4 subnet mask or IPv6 prefix length 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.

---

## 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.

### \*OBSIGN

Filtering of trace information is done to include only the OBSIGN 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.



---

## 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

---

## Instance (INSTANCE)

Specifies the name of the server instance to trace. This parameter is only valid when APP(\*DIRSVR) is specified.

**\*DFT** The default server instance is traced. When APP(\*DIRSVR) is specified, the QUSRDIR server instance will be traced.

**\*ALL** All instances of the server that are currently running are traced.

*character-value*

Specify the name of the server instance to be traced. Up to 32 characters may be specified.

Top

---

## Trace level (TRCLVL)

Specifies the level of granularity of the service trace. This parameter is only valid when APP(\*HTTP), APP(\*SMTPAPI) or APP(\*REXEC) 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, the "From program" or the "To program" includes the specified text, the watched for condition is true. If the message data, the "From program" or the "To program" 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, the "From program" or the "To program" of the watched for message. This text is case sensitive and can be quoted in order to specify imbedded or trailing blanks.

### Element 3: Compare against

Specify which part of the message the comparison data specified for element 2 is to be compared against.

#### \*MSGDATA

The comparison data will be compared against the message replacement data.

#### \*FROMPGM

The comparison data will be compared against the name of the program sending the message, or the name of the ILE program that contains the procedure sending the message.

#### \*TOPGM

The comparison data will be compared against the name of the program the message was sent to, or the name of the ILE program that contains the procedure the message was sent to.

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.

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

### *generic-name*

Specify the generic name of the user 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 user name specifies all jobs with the specified job name and with user names that begin with the generic prefix.

*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 or a generic user 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 field specified for element 4, the watch condition is considered to be true. This text is case sensitive. If \*ALL is specified in the LIC log compare against field, the LIC log fields which will be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile 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.

**Element 4: Compare against**

Specify which part of the LIC log the comparison data specified for element 3 is to be compared against.

**\*ALL** The LIC log comparison data will be compared against all the fields described below.

**\*TDENBR**

The LIC log comparison data will be compared against the number of the task dispatching element (TDE) which requested the LIC log entry.

**\*TASKNAME**

The LIC log comparison data will be compared against the name of the task which requested the LIC log entry. Task name is blank (hex 40s) if the LIC log entry is not requested by a task.

**\*SVRTYPE**

The LIC log comparison data will be compared against the type of server that requested the LIC log entry. Server type is blank (hex 40s) if the LIC log entry is not requested by a server.

**\*JOBNAME**

The LIC log comparison data will be compared against the name of the job which requested the LIC log entry. LIC job name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBUSR**

The LIC log comparison data will be compared against the user name of the job which requested the LIC log entry. LIC user name is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*JOBNBR**

The LIC log comparison data will be compared against the job number (000001-999999) to further qualify the job name and user name of the job which requested the LIC log entry. LIC job number is blank (hex 40s) if the LIC log entry is not requested by a job.

**\*THDID**

The LIC log comparison data will be compared against the thread which requested the LIC log entry. Thread identifier is binary zeros if the LIC log entry is not requested by a thread.

**\*EXCPID**

The LIC log comparison data will be compared against the exception that caused the LIC log entry to be requested. This is a 2-byte hexadecimal field formed by concatenating to the high-order 1-byte exception group number a low-order 1-byte exception subtype number. Exception identifier is binary zeros if the LIC log entry is not requested as a result of an exception.

**\*MODNAME**

The LIC log comparison data will be compared against the LIC module name which requested the LIC log entry. If the module name is greater than 64 characters, the LIC module name is truncated to 64 characters.

**\*MODRUNAME**

The LIC log comparison data will be compared against the LIC module replacement unit name. LIC module RU name is always in upper case EBCDIC.

**\*MODEPNAME**

The LIC log comparison data will be compared against the name of the entry point which requested the LIC log entry. If the entry point name is greater than 128 characters, the LIC module entry point name is truncated to 128 characters.

**\*MODOFFSET**

The LIC log comparison data will be compared against the byte offset into the LIC module text which requested the LIC log entry.

**\*MODTSP**

The LIC log comparison data will be compared against the timestamp of when the LIC module was compiled. The format for this field is the system time-stamp format.

Top

---

## Watch for PAL entry (WCHPAL)

Specifies up to five Product Activity Log (PAL) entries which are to be watched for. When the watched for PAL occurs, the trace exit program is called; if no trace exit program is defined, the trace stops.

**Single values****\*NONE**

No PAL entries will be watched for.

**Other values (up to 5 repetitions)****Element 1: System reference code**

**\*ALL** Any system reference code will be considered to be a match.

***character-value***

Specify the system reference code (SRC) to be watched for. You can specify either a hexadecimal digit or a question mark for each character in the eight-digit code. A question mark is a wildcard character that will match any digit in that position. Up to seven wildcard characters can be specified. You can also specify a generic SRC that 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 SRC specifies all PAL entries with system reference codes that begin with the generic prefix.

**Element 2: Comparison data**

Specify comparison data to be used if a PAL entry matching the specified system reference code occurs. If the field specified in element 3 matches the specified text, the watched for condition is true. If the field specified in element 3 does not match the specified text, the watch function just continues.

#### **\*NONE**

No comparison data is specified. If a PAL entry matching the specified system reference code occurs, the watched for condition is true.

#### ***character-value***

Specify the text string used to compare against the field specified in element 3 of the watched for PAL entry. This text is case sensitive.

You can specify question mark (?) and asterisk (\*) wildcard characters in the text string. A question mark is a single-character wildcard and will match any character in the same position. For example, '??123' will match any value that is five characters long and ends with '123'. Multiple question mark wildcard characters can be specified for the comparison data value.

An asterisk is a multiple-character wildcard character. You can specify a single asterisk wildcard character at the end of the comparison data value. For example, 'ABC\*' will match any value that begins with the letters 'ABC'.

### **Element 3: Compare against**

Specify which part of the PAL entry the comparison data specified for element 2 is to be compared against.

#### **\*RSCNAME**

The comparison data will be compared against the name of the physical device that has the entry in the log. A resource name is assigned at first by the system, but may have been changed to a new value by a user.

#### **\*RSCTYPE**

The comparison data will be compared against the number or word used to identify a product.

#### **\*RSCMODEL**

The comparison data will be compared against the numbers or letters used to identify the feature level of a product with a given type.

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 or a Product Activity Log (PAL) 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.

#### **\*NOMAX**

There is no time limit for watching for a particular message or LIC log entry or PAL 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. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

---

## Run priority (RUNPTY)

Specifies the priority of the job where the watch session work will be run.

**25** A job priority of 25 will be used.

**1-99** Specify the run priority of the job. For more information on job run priority, refer to the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

Top

---

## Job name (JOB)

Specifies which jobs are to be traced. This parameter is only valid when APP(\*SMPTAPI) or APP(\*REXEC) is specified, or when APP(\*TELNET) and ADLTRC(\*JOB) are specified.

### Single values

**\*** Only the job that issues the TRCTCPAPP (Trace TCP/IP application) command is to be traced.

### Element : 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 sixteen job names may be specified.

#### 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.

Top

---

## Examples

### Example 1: Starting Trace for Database

```
TRCTCPAPP APP(*DATABASE) SET(*ON) TRCPNT(QZDA1050 QZDA1060)
LCLNETADR(*CALC
          '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: Starting Trace for Database using IPv6 Address

```
TRCTCPAPP APP(*DATABASE) SET(*ON)
LCLNETADR(*CALC
          '1080:8:800:417A:1::'
          *HOST)
```

This command will start tracing for the database host server using \*INET6 filter, a local address of 1080:8:800:417A:1:: and prefix length of /128.

### Example 3: 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(*CALC '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 4: 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:

```
TCP45B8 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 5: Starting Trace for Packet Rules

```
TRCTCPAPP APP(*PKTRULES) SET(*ON)
          ARGLIST('DebugLv1=1 TraceLv1=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 6: Starting Trace for FTP

```
TRCTCPAPP APP(*FTP) SET(*ON)
          RMTNETADR(*CALC '9.130.69.16'
                    '255.255.255.255' 5000)
```

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 7: Starting Trace for FTP using IPv6 Address

```
TRCTCPAPP APP(*FTP) SET(*ON)
          RMTNETADR(*CALC
                '2001:0DB8:0::0:1428:57AB')
```

This command starts tracing the FTP server using AF\_INET6 filter and IPv6 remote network address 2001:0DB8:0::0:1428:57AB.

#### **Example 8: 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.

#### **Example 9: Starting Trace for TELNET using IPv6 Address**

```
TRCTCPAPP APP(*TELNET) SET(*ON)
          RMTNETADR(*CALC
                'FE80::210:11FF:FE76:7800')
```

This command will start tracing for the TELNET server using AF\_INET6 filter and IPv6 remote network address FE80::210:11FF:FE76:7800.

#### **Example 10: Starting Trace for Analyzing a Device Problem**

```
TRCTCPAPP APP(*TELNET) SET(*ON)
          ADLTRC(*DEVD *JOB *CMNARB *SYSARB)
          DEVD(MYDEVICE)
          JOB((*ALL/*ALL/MYSBS1) (*ALL/*ALL/MYSBS2))
```

This command will trace all the jobs that will be associated with the device description MYDEVICE, which is encountering a device problem. The command will start tracing for the TELNET server. Trace information is collected for all user jobs associated with device description MYDEVICE. Trace information is collected for the TELNET device manager jobs. Trace information is collected for all jobs that were supplied in the JOB parameter. In this example, all jobs with the name MYSBS1 and MYSBS2 will be traced, where MYSBS1 is the subsystem that displays the sign on screen for interactive device sessions and MYSBS2 is the subsystem jobs transfer to when a user signs on. Trace information is also collected for the system jobs that interact with display devices, which are all the QCMNARB and QSYSARB system jobs.

#### **Example 11: Starting Trace for a HTTP Server Instance**

```
TRCTCPAPP APP(*HTTP) SET(*ON) INSTANCE(HTTP1)
```

This command traces the HTTP server instance named HTTP1.

#### **Example 12: Starting Trace for a Directory Services Server Instance**

```
TRCTCPAPP APP(*DIRSRV) SET(*ON) INSTANCE(DIRSRV1)
```

This command traces the Directory Services server instance named 'DIRSRV1'.

**Example 13: Starting Trace for Default Directory Services Server Instance**

```
TRCTCPAPP APP(*DIRSRV) SET(*ON) INSTANCE(*DFT)
```

The default server instance (QUSRDIR) is traced.

**Example 14: Starting Trace for one of the SMTP APIs**

```
TRCTCPAPP APP(*SMTPAPI) SET(*ON) TRCLVL(*VERBOSE)  
JOB(044235/TOCUSR/QPADEV000G)
```

This command traces one of the SMTP API applications using a trace level of verbose and specifying the qualified job 044235/TOCUSR/QPADEV00G.

[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.

---

## 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.

---

## 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.

---

## 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.



---

## 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 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 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 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**

```
TRCTCP RTE RMTSYS('IP6HOST')
```

This command traces the entire route between the local 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 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

---

## Translate Keystore File (TRNCKMKSF)

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

Parameters  
Examples  
Error messages

The Translate Keystore File (TRNCKMKSF) command translates key values stored in the specified keystore files to another master key, or if the same master key is specified, to the current version of the master key. If an error occurs, processing halts immediately.

For more information on keystore, refer to the Cryptographic Services Keystore article in the Cryptographic Services section of the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

### Restrictions:

- You must have object operational (\*OBJOPR), read (\*READ) and update (\*UPD) authority to the keystore file.

Top

---

## Parameters

Keyword	Description	Choices	Notes
KEYSTORE	Keystore file	Values (up to 10 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Keystore file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MSTKEY	Master key	1-8, *SAME	Optional

Top

---

## Keystore file (KEYSTORE)

Specifies the keystore files to use. Up to 10 keystore files can be specified.

This is a required parameter.

### Qualifier 1: Keystore file

*name* Specify the name of the keystore 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 thread is searched. If no library is specified as the current library for the thread, the QGPL library is searched.

*name* Specify the name of the library to search for the file.

Top

---

## Master key (MSTKEY)

Specifies the master key under which the key values will be translated.

This is a required parameter.

### \*SAME

The keystore key values that are encrypted under the old version of the file's master key will be translated to the current version.

1-8 The keystore key values will be translated to the current version of the chosen master key.

Top

---

## Examples

### Example 1: Translate Keystore Keys to the Current Version of the Master Key

```
TRNCKMKSF  KEYSTORE(MYLIB/KEYSTORE1 MYLIB/KEYSTORE2)
```

This command re-encrypts all keys in keystore files KEYSTORE1 and KEYSTORE2 in library MYLIB that are encrypted under the old version of the master key to encryption under the current version.

### Example 2: Translate Keystore Keys to Another Master Key

```
TRNCKMKSF  KEYSTORE(MYLIB/MYKEYSTORE)  MSTKEY(8)
```

This command re-encrypts all keys in a keystore file under the current version of Master Key 8.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF3CF2

Error(s) occurred during running of &1 API.

#### CPF9872

Program or service program &1 in library &2 ended. Reason code &3.

#### CPF9D88

An error occurred during exit program post-processing.

#### CPF9D89

An error occurred during exit program pre-processing.

#### CPF9D8E

Keystore &1 in library &2 was not translated due to exit program cancel.

#### CPF9D96

Key store file requires recovery.

#### CPF9D9F

User not authorized to key store file.

#### CPF9DA0

Error opening key store file.

**CPF9DA5**

Key store file not found.

**CPF9DA6**

Key store file is not available.

**CPF9DA7**

File is corrupt or not a valid key store file.

**CPF9DAB**

One or more keys could not be decrypted.

**CPF9DB3**

Qualified keystore file name is not valid.

**CPF9DB7**

Error occurred writing to the key store file.

**CPF9DB8**

Error occurred reading record from key store.

**CPF9DDA**

Unexpected return code &1 from cryptographic service provider &2.

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 or a remote file system accessed through a Network File System server (\*NFS). 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 i5/OS Network File System Support book, SC41-5714

## Restrictions:

- 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, *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.

**\*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.

**Note:** This parameter is Unicode-enabled. See "Unicode support in CL" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for additional information.

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.

**Note:** This parameter is Unicode-enabled. See "Unicode support in CL" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for additional information.

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.

#### CPFA1B8

\*IOSYSCFG authority required to use &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 QRPLIB 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 QRPLIB 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>Element list</i>	Optional
	Element 1: Service program	<i>Qualified object name</i>	
	Qualifier 1: Service program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
	Element 2: Activation	<b>*SAME, *IMMED, *DEFER</b>	
SRVPGMLIB	Bound *SRVPGM library name	Single values: <b>*SAME, *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, *BASIC, *EXTENDED, *FULL</b>	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.

You can control the activation of each service program. You can specify whether the referenced service program is activated at the same time as the program program being updated, or is deferred until a procedure exported from the referenced service program is called. Deferring activation may improve your application's performance.

### Single values

#### \*NONE

No service programs are examined during symbol resolution.

### Element 1: Service program

#### Qualifier 1: 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.

### Element 2: Activation

#### \*SAME

The referenced service program activation does not change.

#### **\*IMMED**

Activation of the bound service program takes place immediately when the program being updated is activated.

#### **\*DEFER**

Activation of the bound service program may be deferred until a function it exports is called.

---

## 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 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.

Top

---

## 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.

Top

---

## 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 PTF Information (UPDPTFINF)

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

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

The Update Program Temporary Fix Information (UPDPTFINF) command updates PTF information for all PTF save files in library QGPL. This command only needs to be run during a complete system recovery after library QGPL has been restored.

### Restrictions:

- This command is shipped with exclude (\*EXCLUDE) public authority.

[Top](#)

---

## Parameters

None

[Top](#)

---

## Examples

UPDPTFINF

This command updates PTF information for all PTF save files in library QGPL.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF3613

PTF operation &5 failed.

[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>QSRVSR</u></i>	
	Qualifier 2: Library	<i>Name, *<u>LIBL</u>, *CURLIB</i>	
SRCMBR	Export source member	<i>Name, *<u>SRVPGM</u></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>Element list</i>	Optional
	Element 1: Service program	<i>Qualified object name</i>	
	Qualifier 1: Service program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *<u>LIBL</u></i>	
	Element 2: Activation	<b>*<u>SAME</u></b> , <b>*IMMED</b> , <b>*DEFER</b>	
SRVPGMLIB	Bound *SRVPGM library name	Single values: <b>*<u>SAME</u></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, *<u>LIBL</u>, *CURLIB, *USRLIBL</i>	
ACTGRP	Activation group	<i>Name, *<u>SAME</u></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>*<u>NONE</u></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. You can control the activation of each service program. You can specify whether the referenced service program is activated at the same time as the program or service program being created, or is deferred until a procedure exported from the referenced service program is called. Deferring activation may improve your application's performance.

### Single values

#### \*NONE

No service programs, except those in the bound service program being updated, are examined during symbol resolution.

### Element 1: Service program

#### Qualifier 1: 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.

## Element 2: Activation

### \*SAME

The referenced service program activation does not change.

### \*IMMED

Activation of the bound service program takes place immediately when the service program being updated is activated.

### \*DEFER

Activation of the bound service program may be deferred until a function it exports is called.

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 (RTVSYSDINF) 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 source system was at V3R0M5 or later.

**Restrictions:** To use this command, you must have \*SECADM, \*ALLOBJ, \*AUDIT, \*JOBCTL, and \*SAVSYS authority.

Top

---

## Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i>	Required, Positional 1
TYPE	Type of information	*ALL, *EDTD, *NETA, *RPLYE, *SRVATR, *SRVPVD, *SYSVAL	Optional, Positional 2

Top

---

## Library (LIB)

Specifies the library where the system information was placed by the Retrieve System Information (RTVSYSDINF) 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

#### **CPF9810**

Library &1 not found.

#### **CPFA976**

Error occurred updating system information for type &1.

#### **CPFB0CD**

Error occurred during Enhanced Upgrade Assistant for i5/OS function.

**CPFBOE1**

Not all objects were checked.

**CPFBOE2**

Not all objects were converted.

[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.

#### \*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.

### \*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.

### *character-value*

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.

### \*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.

### *character-value*

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.

### \*VERBOSE

Display verification message after each iteration.

### \*QUIET

Display only initial and summary messages.

Top

---

## Packet length (in bytes) (PKTLEN)

Specifies the length (in bytes) of the packets that are exchanged between the local and remote systems.

100 The packet length is 100 bytes.



0-32763

Specify the number of bytes for each packet.

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.

2 Two iterations are performed.

1-32767

Specify the number of iterations.

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.

1 One packet is sent for each iteration.

1-32767

Specify the number of packets that are sent for each iteration.

Top

---

## Echo (ECHO)

Whether the remote location should echo packets back to the local location.

\*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.

10 The system waits 10 seconds.

\*NOMAX

The system waits forever.

\*NOWAIT

The system returns immediately if there is a connection ready and available.

2-3600 Specify the number of seconds to wait.

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, X''</i> , *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 IBM System Manager for i5/OS 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 System i5.

### **\*REMOTE**

Communications hardware is checked to verify that it is operating correctly on another System i5 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:

- The following authorities are required to verify an image catalog:
  1. Execute (\*EXECUTE) authority to library QUSRSYS.
  2. Use (\*USE) authority to the image catalog.
  3. Use (\*USE) authority to the virtual device description.
  4. Execute (\*X) authority to each directory in the image catalog path name.
- This command is not supported for dependent image catalogs.

[Top](#)

---

## Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
TYPE	Verify type	*UPGRADE, *PTF, *OTHER	Optional
SORT	Sort image catalog	*NO, *YES	Optional
NFSSHR	Network file server share	*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](#)

---

## Verify type (TYPE)

Specifies the type of image catalog to be verified.

### **\*UPGRADE**

The optical 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 optical images in order for i5/OS:

1. Licensed Internal Code (Required)
2. Operating system (Required)
3. Library QGPL (Required)
4. Library QUSRSYS (Required)
5. No-charge options
6. No-charge (bonus) licensed programs and options
7. Keyed set products
8. Single products
9. Secondary languages
10. Program temporary fixes

The following list shows the required tape images in order for i5/OS:

1. Licensed Internal Code
2. Operating system
3. Library QGPL
4. Library QUSRSYS

**\*PTF** The optical 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 device. All non-PTF volumes will be unloaded. This value is only supported for optical image catalogs.

### **\*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

---

## **Network file server share (NFSSHR)**

Specifies whether the images in the image catalog are prepared for use by a network file server. This option is only valid for optical image catalogs.

**\*NO** The images in the image catalog and directory are not prepared for use as a network file server (NFS) share.

**\*YES** The images in the image catalog and directory are prepared for use as a network file server (NFS)



share. In addition, a VOLUME\_LIST file in ASCII format will be created in the image catalog directory. This file will contain a list of image files to be used by a virtual device of type 632B-003.

Top

---

## Examples

### Example 1: Verify Image Catalog for a Software Upgrade

```
VFYIMGCLG  IMGCLG(MYCLG)  TYPE(*UPGRADE)  SORT(*YES)
```

This command verifies that image catalog **MYCLG** contains the necessary files for a software upgrade. If the necessary image files exist, the images will be sorted in the order required for a software upgrade.

### Example 2: Verify Image Catalog for Applying PTFs

```
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.

### Example 3: Verify Image Catalog for Sharing with a Network File Server

```
VFYIMGCLG  IMGCLG(MYCLG)  TYPE(*PTF)  SORT(*YES)  
NFSSHR(*YES)
```

This command verifies that image catalog **MYCLG** contains the necessary files for loading and applying Program Temporary Fixes (PTF). If the necessary image files exist, the images will be sorted in the order required for a PTF load and apply. The image catalog directory will be setup for use by a network file server.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPFBC20

Verification for image catalog &1 failed.

#### CPFBC45

Image catalog &1 not found.

#### CPF9802

Not authorized to object &2 in &3.

#### CPF9820

Not authorized to use library &1.

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	01-FB, *LCL	Optional
RMTDCEADR	Remote DCE address	01-FB, *ANY	Optional
OUTPUT	Output	_, *PRINT	Optional
SEQCOUNT	Number of sequences	1-3, 1	Optional
DTEPORT	Remote DTE port	A, 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.

Top

---

## 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.

Top

---

## 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.

Top

---

## 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.

Top

---

## 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.

Top

---

## 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.

Top

---

## 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.

Top

---

## 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 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.

[Top](#)

---

## 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	<i>Hexadecimal value, <u>00000000</u></i>	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 service information collection and transmission whether or not Service Agent is activated for problem reporting.

#### \*TSTPRB

Service Agent is to create a problem log entry with either a valid error log identifier or a null (00000000) Error Log ID. The problem will then be reported using the normal Service Agent problem reporting process. This allows a test of the Service Agent problem reporting function. QSRV and QSYSOPR messages may be checked after a few minutes for Service Agent messages and the Problem Management Record (PMR).

Top

---

### Error log identifier (ERRLOGID)

Specifies the error log identifier for the Product Activity Log entry 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 entry will be used. The problem log entry and test problem will have no product activity log identifier.

### *hexadecimal-value*

Specify a valid eight character identifier from the product activity log. Product activity log 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 verifies the selected service and support application: Electronic Customer Support (ECS) and Electronic Service Agent.

Communications will be started for the selected service using the existing configuration on the system. After the session becomes active and the service configuration has been verified, the communication session will be ended.

Top

---

### Parameters

Keyword	Description	Choices	Notes
SERVICE	Service	*ECS, *SRVAGT, *DOC	Required, Positional 1

Top

---

### Service (SERVICE)

Specifies the service that will be verified.

This is a required parameter.

**\*ECS** Electronic Customer Support (ECS) and the problem reporting function of Electronic Service Agent connection will be verified.

**\*SRVAGT**  
Electronic Service Agent service information transmission connection will be verified.

Top

---

### Examples

#### Example 1: Verify the Electronic Customer Support Service Configuration

```
VFYSRVCFG SERVICE(*ECS)
```

This command will verify that the service configuration used by Electronic Customer Support (ECS) and the problem reporting function of Electronic Service Agent can connect to IBM.

#### Example 2: Verify the Electronic Service Agent Service Configuration

```
VFYSRVCFG SERVICE(*SRVAGT)
```

This command will verify that the service configuration used by the service information transmission function of Electronic 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. IP Version 6 host name resolution will be performed if the system has at least one IP Version 6 address configured. If an IP Version 6 address is not found, IP Version 4 host name resolution will be performed if the system has at least one IP Version 4 address configured. The loopback address is not considered in this case as a configured 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:** 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 descriptions, 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 descriptions of type \*WINDOWSNT are automatically started by default and can optionally be controlled with the **Start TCP/IP interfaces (STRTCPIFC)** parameter.

Downline attached objects can be varied on or off along with the specified object by specifying the value \*NET for the **Range (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 parameter has no affect when varying devices.

Varying on network interfaces, and lines synchronously or asynchronously can be controlled by the **Vary on wait (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 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 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

A network server description of type \*ISCSI cannot be varied on:

- Until the Network Server Host Adapter Device Description is varied on.

A line cannot be varied on:

- Until the Network Interface Description is varied on, in the case of IDLC lines.
- Until the Network Server Description is varied on.

- 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 some tape devices, they are not attached to a controller, so this restriction does not apply.

A network server description cannot be varied off:

- Until all attached devices and controllers are varied off. Varying off the network server description also varies off the attached line descriptions.
- If any system clients have files open on the integrated 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.

A network server host adapter device cannot be varied off:

- If any network server description that is using it is varied on.

When the 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.

When the 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 VRYWAIT parameter is specified on the VRYCFG command for a line description that is not Token-Ring, Ethernet, X.25, or switched SDLC, BSC, or Async, the parameter is accepted but ignored.

When the **Reset system (RESETSYS)** parameter is used:

- For Network Server Descriptions: When varying on, the value \*YES will force IBM Director to interface with the xSeries system and validate the xSeries is available for activity.

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
VRYWAIT	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
FRCVRYOFF	Forced vary off	*NO, *YES, *LOCK	Optional
SBMMLTJOB	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> , *LIBL	
GENPTHCERT	Generate path certificate	*NO, *YES	Optional
RESETSYS	Reset system	*NO, *YES	Optional

Top

## Configuration object (CFGOBJ)

Specifies the configuration objects to be varied.

This is a required parameter.

### Single values

#### \*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.

## Other values (up to 256 repetitions)

### *generic-name*

Specify a generic name of the configuration objects to be varied.

**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.

*name* Specify the name of the configuration object to be varied.

Top

---

## Type (CFGTYPE)

Specifies the type of object to be varied.

This is a required parameter.

**\*NWS** The network server description 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.

This is a required parameter.

**\*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 for the **Resource name (RSRCNAME)** 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 for the **Resource name (RSRCNAME)** 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 for the **Resource name (RSRCNAME)** 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 object is varied on asynchronously or synchronously. For synchronous vary on, specifies how long the system waits for the vary on to complete.

**\*CFGOBJ**

The system uses the value specified for the **Vary on wait (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.

**15-180** Specify the number of seconds to wait. 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.

**\*NO** The associated IOP is not reset.

**\*YES** The associated IOP is reset.

---

## Resource name (RSRCNAME)

Specifies the drives within the media library device to be reset or reallocated.

### Single values

**\*ALL** All drives within the media library device are to be reset or reallocated.

### Other values (up to 16 repetitions)

*name* Specify the name of the drive within the media library device to be reset or reallocated.

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.

### Notes:

1. This parameter is valid for STATUS(\*DEALLOCATE) only when a tape media library object is specified for the **Configuration object (CFGOBJ)** parameter.
2. 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 exit point documentation in the i5/OS Information Center for additional information on vary configuration exit point processing.

**\*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, Intra, NWSH, and MLB 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

---

## 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 \*NWS or \*DEV is specified for the **Type (CFGTYPE)** parameter.

**\*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 (JOBBD)

Specifies the job description 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.

### Qualifier 1: Job description

#### QBATCH

Job description QBATCH is used as the job description of the submitted job. Note that the QBATCH job description, as shipped with the system, specifies job queue QBATCH which is configured to allow a maximum of 1 job to run at a time.

*name* Specify the 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:

```
CRTJOBBD  JOBBD(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.

### 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 job description is located.

Top

---

## Generate path certificate (GENPTHCERT)

Specifies whether to generate a new set of certificates on a remote network server system. This parameter is valid only when CFGTYPE is \*NWS and the network server description is of type \*ISCSI and the STATUS is \*ON.

\*NO The set of certificates is not generated.

\*YES The set of certificates is generated.

**Note:** You must have input/output system configuration (\*IOSYSCFG) and security administrator (\*SECADM) special authority to use this parameter value.

Top

---

## Reset system (RESETSYS)

Specifies whether to force a reset of the remote network server system, regardless of its current power state. This is an advanced function, caution is advised when this option is selected. This parameter is valid only when CFGTYPE is \*NWS and the network server description is of type \*ISCSI and the STATUS is \*ON.

\*NO The system is not reset.

\*YES The system is reset.

**Note:** You must have all object (\*ALLOBJ) special authority to use this parameter value.

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 Description**

```
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 Server Descriptions 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 description 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.

**Example 12: Reset Certificates when Varying On Network Server Description**

```
VRYCFG CFGOBJ(SERVER1) CFGTYPE(*NWS) STATUS(*ON)  
GENPHTCERT(*YES)
```

This command will vary on the network server description and request that a new set of certificates are generated during the vary on.

**Example 13: Restart Remote Integrated Server at Vary on of Server Description**

```
VRYCFG CFGOBJ(SERVER1) CFGTYPE(*NWS) STATUS(*ON)  
RESETSYS(*YES)
```

This command will vary on the network server description and request reset of the remote integrated server during the vary on of SERVER1.

---

## 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.

---

## 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 program or ILE 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 program or ILE 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 program or ILE 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 program or ILE 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 program or ILE 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 program or ILE 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 program or ILE 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 program or ILE 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)  RCDfmt(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 program or ILE 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 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 program or ILE 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 topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/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

---

## 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	*NO, *YES	Optional, Positional 2
SBS	Subsystem	Single values: *ALL Other values (up to 25 repetitions): <i>Name</i>	Optional
CPUPCTLMT	CPU percent limit	0.1-99.9, *NONE	Optional
RSPLMT	Response time limit	0.1-999.9, *NONE	Optional
SEQ	Sequence	*SBS, *AUXIO, *CPU, *CPUPCT, *CURUSR, *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, *ALL, *SYS, *SBS</i>	
INTERVAL	Automatic refresh interval	5-999, *PRV	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.

**\*CURUSR**

Jobs are put in alphabetical order by the user profile under which the initial thread is currently running.

**\*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)  
Threadsafes: 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.

Top

---

## 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 DSPORT(*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 valid.

**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	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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](#)

---

## Work with ARM Jobs (WRKARMJOB)

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

Parameters  
Examples  
Error messages

Work with ARM Jobs (WRKARMJOB) command allows you to work with jobs which are actively using ARM APIs version 4.0 (or later).

ARM APIs version 4.0 (or later version) is an implementation of a respective version of Application Response Measurement (ARM) standard. ARM standards are developed by The Open Group. For more information on ARM standards see

<http://www.theopengroup.org>

For more information on IBM use of ARM standards see IBM Systems Software Information Center.

(navigate to Product Listing and then to Enterprise Workload Manager).

### Restrictions:

- WRKARMJOB command will only show jobs which use implementation of ARM APIs shipped with the operating system. Jobs which use other implementations of ARM APIs will not be shown.
- In order to work with a specific job or to display details of a specific job, user must have authority to the job. It means that WRKARMJOB command must be issued from within the job being worked with, or the issuer of the command must be running under a user profile which is the same as the job user identity of the job, or the issuer of the command must be running under a user profile which has job control (\*JOBCTL) special authority.

Top

---

## Parameters

None

Top

---

## Examples

WRKARMJOB

This command will show the Work with ARM Jobs panel. This panel shows the list of jobs which are actively using ARM APIs version 4.0 (or later). From this panel, you can display detailed ARM information for the jobs.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF0AA7

ARM support is not enabled.

#### CPF3792

Information not displayed. Error occurred.

[Top](#)

---

## Work with ASP Jobs (WRKASPJOB)

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

Parameters  
Examples  
Error messages

The Work with ASP Jobs (WRKASPJOB) command allows you to work with a list of jobs that are using an auxiliary storage pool (ASP).

You can use this command to perform the following tasks:

- Select jobs to end or select an ASP to end all jobs using it.
- Select jobs to work with or select an ASP to work with all jobs using it.
- Select jobs to send messages to or select an ASP to send a message to all jobs using it.

### Restrictions:

- You must have use (\*USE) authority to the ASP device description.

Top

---

## Parameters

Keyword	Description	Choices	Notes
ASPDEV	ASP device	Name, <u>*ALLAVL</u>	Optional, Positional 1

Top

---

## ASP device (ASPDEV)

Specify the name of the auxiliary storage pool (ASP) device whose jobs are to be displayed. When working with an entire ASP group, the ASP device specified should be that of the primary. Specifying the name of a secondary will only provide the detail vary status when the secondary is being varied to join an already online ASP group.

### \*ALLAVL

Using job information will be shown for all ASP devices that currently have a state of 'Available'.

**name** Specify the name of the ASP whose using jobs you want to show.

Top

---

## Examples

### Example 1: Work with Jobs Using any Available ASP

```
WRKASPJOB ASPDEV(*ALLAVL)
```

This command shows every available ASP device and the jobs that are using it.

## Example 2: Work with Jobs Using a Specific ASP

WRKASPJOB ASPDEV(WAREHOUSE)

This command shows the jobs that are using an ASP device named WAREHOUSE.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF9814

Device &1 not found.

#### CPF9825

Not authorized to device &1.

#### CPF9871

Error occurred while processing.

#### CPF9899

Error occurred during processing of command.

[Top](#)



---

## Work with Authority (WRKAUT)

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

Parameters  
Examples  
Error messages

The Work with Authority (WRKAUT) command shows the list of authorized users of an object and their associated authorities. From the list, you can select from options to perform the following:

- Add a user
- Change user authority
- Remove a user

The following are displayed for the specified object:

- The object path name
- The name of the object's owner
- The name of the object's primary group
- A list of all the users who are authorized to use the object
- The authorities that each user has for the object

If an object does not have an owner name associated with it, no authorities for the object are shown.

See Appendix D of the System i Security Reference, SC41-5302 for the authorities needed to use this command.

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
SYMLNK	Symbolic link	*NO, *YES	Optional

Top

---

### Object (OBJ)

Specifies the objects for which the authorized users and their authorities are to be shown.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

### *path-name*

Specify the path name of the objects for which specific authorities are to be shown.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. 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.

Top

---

## Symbolic link (SYMLNK)

If the last component in the path name is a symbolic link, specifies whether or not to work with the symbolic link or the object pointed to by the symbolic link.

**\*NO** The symbolic link object is not worked with. The object pointed to by the symbolic link is worked with.

**\*YES** If the object is a symbolic link, the symbolic link is worked with. The object pointed to by the symbolic link is not worked with.

Top

---

## Examples

```
WRKAUT OBJ('/QSYS.LIB/ARLIB.LIB/PROG1.PGM')
```

This command causes the list of authorized users and their authorities for the object named PROG1 to be shown. PROG1 is a program located in the library named ARLIB.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPDA080

User profile name too long.

#### CPE3101

A non-recoverable I/O error occurred.

#### CPE3408

The address used for an argument was not correct.

#### CPE3418

Possible APAR condition or hardware failure.

#### CPE3474

Unknown system state.

#### CPFA0AA

Error occurred while attempting to obtain space.

#### CPFA0AB

Operation failed for object. Object is &1.

**CPFA0AD**  
Function not supported by file system.

**CPFA0A1**  
An input or output error occurred.

**CPFA0A2**  
Information passed to this operation was not valid.

**CPFA0A3**  
Path name resolution causes looping.

**CPFA0A4**  
Too many open files for process.

**CPFA0A5**  
Too many open files.

**CPFA0A7**  
Path name too long.

**CPFA0A9**  
Object not found. Object is &1.

**CPFA0B1**  
Requested operation not allowed. Access problem.

**CPFA0C0**  
Buffer overflow occurred.

**CPFA0C1**  
CCSID &1 not valid.

**CPFA08B**  
Path name cannot begin with \*.

**CPFA08C**  
Pattern not allowed in path name directory.

**CPFA08E**  
More than one name matches pattern.

**CPFA085**  
Home directory not found for user &1.

**CPFA086**  
Matching quote not found in path name.

**CPFA087**  
Path name contains null character.

**CPFA088**  
Path name pattern not valid.

**CPFA09C**  
Not authorized to object. Object is &1.

**CPFA09D**  
Error occurred in program &1.

**CPFA09E**  
Object in use. Object is &1.

**CPFA09F**  
Object damaged. Object is &1.

**CPFA091**  
Pattern not allowed in user name.

**CPFA092**  
Path name not converted.

**CPFA093**  
Name matching pattern not found.

**CPFA094**  
Path name not specified.

**CPF1F05**  
Directory handle not valid.

**CPF1F41**  
Severe error occurred while addressing parameter list.

**CPF1F4A**  
Value for number of directory entries not valid.

**CPF1F53**  
Value for length of data buffer not valid.

**CPF2203**  
User profile &1 not correct.

**CPF2225**  
Not able to allocate internal system object.

**CPF22F0**  
Unexpected errors occurred during processing.

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

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

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

Top

---

## Work with Authorization Lists (WRKAUTL)

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

Parameters  
Examples  
Error messages

The Work with Authorization Lists (WRKAUTL) command allows you to show a list of authorization lists from which you can display and change authorization lists.

### Restrictions:

- Only the authorization lists to which you have some authority will be shown on the display.
- To perform operations on the authorization lists, you must have use (\*USE) authority to the command used by the operation, and the appropriate authority to the authorization list on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
AUTL	Authorization list	<i>Qualifier list</i>	Required, Positional 1
	Qualifier 1: Authorization list	<i>Generic name, name, *ALL</i>	

Top

---

## Authorization list (AUTL)

Specifies the authorization lists to be shown.

This is a required parameter.

**\*ALL** A list of all the authorization lists that you own or have authority to view is shown.

### *generic-name*

Specify the generic name of the authorization lists 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 authorization lists that have names with the same prefix as the generic name are shown.

**name** Specify the name of the authorization list to be shown.

Top

---

## Examples

```
WRKAUTL AUTL(FR*)
```

This command allows you to work with a list of all the authorization lists that begin with 'FR' that you have authority to see.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Work with Binding Directories (WRKBNDDIR)

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

Parameters  
Examples  
Error messages

The Work with Binding Directory (WRKBNDDIR) command allows you to display and work with a list of binding directories.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority are searched.
- Only the binding directories to which you have some authority are shown on the display.
- To perform operations on the binding directories, you must have \*USE authority to the command and the appropriate authority to the binding directory on which the operation is performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
BNDDIR	Binding directory	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Binding directory	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Binding directory (BNDDIR)

Specifies the binding directory to work with.

This is a required parameter.

### Qualifier 1: Binding directory

**\*ALL** Find all binding directories in the specified library or libraries.

#### *generic-name*

Specify the generic name of the binding directories. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. If a generic name is specified, then all binding directory objects with names that begin with the generic name, and for which the user has authority, are shown. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

*name* Specify the name of the binding directory to work with.

### 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 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.

\*ALL All libraries in the system portion of the job's library list, including QSYS, 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	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVRxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, 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 QUSRVRxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

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

Top

---

## Examples

```
WRKBNDDIR BNDDIR(HOLDER)
```

This command allows you to work with a binding directory named HOLDER.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF5D0B

Binding directory &1 was not created



**CPF9809**

Library &1 cannot be accessed.

**CPF9820**

Not authorized to use library &1.

[Top](#)



---

## Work with Binding Dir Entries (WRKBNDDIRE)

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

Parameters  
Examples  
Error messages

The Work with Binding Directory Entries (WRKBNDDIRE) command allows you to work with the entries in a binding directory.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority are searched.
- To perform operations on the binding directories, you must have \*USE authority to the command and the appropriate authority to the binding directory on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
BNDDIR	Binding directory	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Binding directory	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	

Top

---

## Binding directory (BNDDIR)

Work with the entries in the specified binding directory.

This is a required parameter.

### Qualifier 1: Binding directory

*name* Specify the name of the binding directory whose entries are to be shown.

### 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

---

## Examples

WRKBNDDIRE BND DIR(COINS)

This command allows you to work with the entries in binding directory COINS.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

#### CPF5D01

Binding directory &1 in library &2 is not usable.

#### CPF980F

Binding directory &1 in library &2 not found.

#### CPF9801

Object &2 in library &3 not found.

#### 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.

#### CPF9809

Library &1 cannot be accessed.

#### CPF9810

Library &1 not found.

#### CPF9820

Not authorized to use library &1.

#### CPF9830

Cannot assign library &1.

[Top](#)

---

## Work with BOOTP table (WRKBPTBL)

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

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

The Work with BOOTP Table (WRKBPTBL) command allows you to work with entries in the BOOTP table. Each table entry contains a client host name, an MAC address, and an internet protocol (IP) address.

You can add, change, remove, or display entries in this table. You can also print the table.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKBPTBL

This command displays the Configure TCP/IP BOOTP Table menu.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Work with Configuration Lists (WRKCFGL)

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

Parameters  
Examples  
Error messages

The Work with Configuration Lists (WRKCFGL) command allows you to work with configuration list functions through the Work with Configuration Lists display.

[Top](#)

---

### Parameters

Keyword	Description	Choices	Notes
CFGL	Configuration list	<i>Generic name, name, *ALL, *APPNDIR, *APPNLCL, *APPNRMT, *APPNSSN, *ASYNCADR, *ASYNCLC, *RTLPASTR, *SNAPASTR</i>	Optional, Positional 1

[Top](#)

---

### Configuration list (CFGL)

Specifies the configuration list to work with.

**\*ALL** Work with all configuration lists.

**\*APPNDIR**

Work with the advanced peer-to-peer networking (APPN) directory configuration list.

*generic-configuration-list-name*

Specify a generic configuration list 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.

*configuration-list-name*

Specify the name of a specific configuration list.

**\*APPNLCL**

Work with the APPN local location configuration list.

**\*APPNRMT**

Work with the APPN remote location configuration list.

**\*APPNSSN**

Work with the APPN session configuration list.

**\*ASYNCADR**

Work with asynchronous PAD network address configuration list(s).

**\*ASYNCLC**

Work with the asynchronous remote location configuration list.

**\*RTLPASTR**

Work with the retail pass-through configuration list.

### **\*SNAPASTHR**

Work with the SNA pass-through configuration list.

This is a required parameter.

[Top](#)

---

## **Examples**

WRKCFGL CFGL(PEG\*)

This command allows you to utilize the Work with Configuration Lists panel to work with entries for all configuration lists whose names start with 'PEG'.

[Top](#)

---

## **Error messages**

None

[Top](#)



---

## Work with Configuration Status (WRKCFGSTS)

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

Parameters  
Examples  
Error messages

The Work with Configuration Status (WRKCFGSTS) command is used to display and to work with configuration status functions. When you run this command, the Work with Configuration Status display is shown.

For network server descriptions, the status displayed is the configuration status of the object. Options are available from Work with Configuration Status to display the status of the integrated server functions and client sessions. Use this option to check for active clients before varying off a network server description.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CFGTYPE	Type	*NWS, *NWI, *LIN, *CTL, *DEV	Required, Positional 1
CFGD	Configuration description	<i>Generic name, name</i> , *ALL, *CMN, *ASYN, *BSC, *DDI, *ELAN, *FAX, *FR, *IDLC, *NET, *PPP, *SDLC, *TDLC, *TRLAN, *WLS, *X25, *WS, *LWS, *RWS, *VWS, *CRP, *DKT, *MLB, *OPTMLB, *TAPMLB, *OPT, *ASP, *NWSH, *TAP, *DSP, *LCLDSP, *RMTDSP, *VRTDSP, *PRT, *LANPRT, *LCLPRT, *RMTPT, *VRTPT, *LOC, *SNPT, *APPC, *FNC, *HOST, *RTL, *INTRA, *ATM, *ISDN, *T1	Optional, Positional 2
OUTPUT	Output	*, *PRINT	Optional, Positional 3
RMTLOCNAME	Remote location	<i>Generic name, name</i> , *NONE	Optional
RANGE	Range	*NET, *OBJ	Optional
STATUS	Status	*ALL, *ACTIVE, *FAILED, *VARYOFF, *VARYON, *AVAILABLE	Optional
ASTLVL	Assistance level	*PRV, *USRPRF, *BASIC, *INTERMED, *ADVANCED	Optional

Top

---

### Type (CFGTYPE)

Specifies the type of description for which you want the status to be shown.

\*NWS Status for network server descriptions is shown.

\*NWI Status for network interfaces is displayed.

\*LIN Status for lines is displayed.

\*CTL Status for controllers is displayed.

\*DEV Status for devices is displayed.

Top

---

## Configuration description (CFGD)

Specifies the descriptions you want displayed on the Work with Configuration Status display.

- \*ALL** Status for all network interfaces, all lines, all controllers, or all devices is displayed depending on the value for the **Type (CFGTYPE)** parameter. Status for any attached controllers and devices is also shown.
- \*CMN** Status for communications controllers or devices is displayed depending on the value for the CFGTYPE parameter.
- \*APPC**  
Status for Advanced Program-to-Program Communications (APPC) controllers or devices are displayed.
- \*ASP** Status for Auxiliary storage pool (ASP) devices is displayed.
- \*ASYNC**  
Status for Async lines, controllers, or devices is displayed.
- \*ATM** Status for Asynchronous Transfer Mode (ATM) network interfaces is displayed.
- \*BSC** Status for Bisynchronous lines, controllers, or devices is displayed.
- \*DDI** Status for all distributed data interface lines is displayed.
- \*DKT** Status for diskette devices is displayed.
- \*DSP** Status for display devices is displayed.
- \*ELAN**  
Status for Ethernet lines is displayed.
- \*FAX** Status for all facsimile (fax) lines is displayed.
- \*FNC** Status for finance controllers or devices is displayed.
- \*FR** Status for all frame relay network interfaces or lines is displayed.
- \*HOST**  
Status for all host controllers or devices is displayed.
- \*IDLC** Status for IDLC lines is displayed.
- \*INTRA**  
Status for intrasystem devices are displayed.
- \*ISDN**  
Status for Integrated Systems Digital Network (ISDN) network interfaces are displayed.
- \*LANPRT**  
Status descriptions of local area network (LAN) printer devices are shown.
- \*LCLDSP**  
Status for local display station devices is displayed.
- \*LCLPRT**  
Status for local printer devices is displayed.
- \*LOC** Status for devices at a specific remote location is displayed. To specify \*LOC, you must specify \*DEV value for the CFGTYPE parameter.
- \*LWS** Status for local work station controllers is displayed.
- \*MLB** Status for both optical and tape media library devices is displayed.
- \*NET** Status for network lines, controllers, or devices is displayed.

- \*NWSH** Status for network server host adapter devices is displayed.
- \*OPT** Status for optical devices is displayed.
- \*CRP** Status for cryptographic devices is displayed.
- \*OPTMLB** Status for optical media library devices is displayed.
- \*PPP** Status descriptions of Point-to-Point Protocol (PPP) lines are shown.
- \*PRT** Status for all printer devices is displayed.
- \*RMTDSP** Status for remote display station devices is displayed.
- \*RMTprt** Status for remote printer devices is displayed.
- \*RTL** Status for retail controllers or devices is displayed.
- \*RWS** Status for remote work station controllers is displayed.
- \*SDLC** Status for SDLC lines is displayed.
- \*SNPT** Status descriptions of SNA pass-through devices are shown.
- \*T1** Status for T1 network interfaces is displayed.
- \*TAP** Status for tape controllers or devices is displayed, depending on the value for the CFGTYPE parameter.
- \*TAPMLB** Status of tape media library devices is displayed.
- \*TDLC** Status for TDLC lines is displayed.
- \*TRLAN** Status for Token-ring lines is displayed.
- \*VRTDSP** Status for virtual (pass-through) display station devices is displayed.
- \*VRTprt** Status for virtual (pass-through) printer devices is displayed.
- \*VWS** Status for virtual (pass-through) work station controllers is displayed.
- \*WS** Status for all work station controllers is displayed.
- \*X25** Status for X.25 lines is displayed. Status for all display station devices is displayed.

***generic-name***

Specify the generic name of the configuration descriptions to be displayed.

**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.

***name*** Specify the name of the configuration description to be displayed. Status for the specific description and any attachments is displayed.

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

---

## Remote location (RMTLOCNAME)

Specifies the remote location name of the devices for which you want status displayed. Only those device descriptions with the specified remote location name are listed on the Work with Configuration Status display.

This parameter is required if \*LOC value is specified for the **Configuration description (CFGD)** parameter. It is not a valid parameter if any value other than \*LOC is specified for the CFGD parameter.

### **\*NONE**

Status is not being displayed depending on the remote location name. \*NONE should be specified if \*NWI, \*LIN, or \*CTL value is specified for the **Type (CFGTYPE)** parameter. \*NONE should also be specified if any value other than \*LOC is specified for the CFGD parameter.

### *generic-name*

Specify a generic remote location 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.

*name* Specify the remote location name of the devices for which you want status displayed.

Top

---

## Range (RANGE)

Specifies whether downline or upline attached configuration descriptions are shown.

**\*NET** If a name for a single description is specified for the **Configuration description (CFGD)** parameter, both downline and upline descriptions are shown. If a special value or generic name is specified for the CFGD parameter, downline descriptions are shown.

**\*OBJ** Only objects of the type specified for the **Type (CFGTYPE)** parameter are shown.

Top

---

## Status (STATUS)

Specifies the status values used to subset the list of descriptions shown. This parameter is ignored if a name for a single description is specified for the **Configuration description (CFGD)** parameter.

**\*ALL** All descriptions are included in the list regardless of their status.

### **\*ACTIVE**

All descriptions with an active status are shown.

**\*AVAILABLE**

All descriptions with an available status are shown.

**\*FAILED**

All descriptions with a failed, recovery, damaged, shutdown, or unknown status are shown.

**\*VARYOFF**

All descriptions with a varied off or vary off pending status are shown.

**\*VARYON**

All descriptions that do not have a varied off or vary off pending status are shown.

Top

---

## Assistance level (ASTLVL)

Specifies which user interface to use.

**\*PRV** The previous user interface used is shown.

**\*USRPRF**

The user interface specified for the **Assistance level (ASTLVL)** parameter of your user profile is used.

**\*BASIC**

The Operational Assistant user interface is used.

**Note:** The \*BASIC value is valid on the ASTLVL parameter only when \*DEV value is specified for the **Type (CFGTYPE)** parameter.

**\*INTERMED**

The system user interface is used.

**\*ADVANCED**

The expert user interface is used.

Top

---

## Examples

### Example 1: Showing the Status for All Remote Display Stations

```
WRKCFGSTS  CFGTYPE(*DEV)  CFGD(*RMTDSP)
```

This command uses the Work with Configuration Status display to show the status for all remote display stations.

### Example 2: Showing the Status for All Network Servers

```
WRKCFGSTS  CFGTYPE(*NWS)  CFGD(*ALL)
```

This command allows the user to utilize the Work with Configuration Status command to show the status for all network server descriptions on the system.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF1E99

Unexpected error occurred.

#### CPF2602

Controller &1 not found.

#### CPF2702

Device description &1 not found.

#### CPF2703

Controller description &1 not found.

#### CPF2704

Line description &1 not found.

#### CPF9846

Error while processing file &1 in library &2.

Top

---

## Work with Chart Formats (WRKCHTFMT)

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

Parameters  
Examples  
Error messages

The Work with Chart Formats (WRKCHTFMT) command allows you to show a list of chart formats from one or more libraries.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the chart formats to which you have some authority will be shown on the display.
- To perform operations on the chart formats, you must have \*USE authority to the command used by the operation, and appropriate authority to the chart formats on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CHTFMT	Chart format	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Chart format	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Chart format (CHTFMT)

Specifies the chart formats to be shown.

This is a required parameter.

### Qualifier 1: Chart format

**\*ALL** All chart formats in the libraries identified in the library qualifier are shown.

#### *generic-name*

Specify the generic name of the chart formats 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 chart formats that have names with the same prefix as the generic name are shown.

*name* Specify the name of the chart format 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	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRRDARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

```
WRKCHTFMT  CHTFMT(LIB01/ABC*)
```

This command allows you to work with a list of chart formats *beginning* with chart formats whose names begin with 'ABC' that are stored in library LIB01.

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 Classes (WRKCLS)

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

Parameters  
Examples  
Error messages

The Work with Classes (WRKCLS) command allows you to show a list of available classes from one or more libraries.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the classes to which you have some authority will be shown on the display.
- To perform operations on the classes, you must have \*USE authority to the command used by the operation, and the appropriate authority to the classes on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CLS	Class	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Class	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Class (CLS)

Specifies the class descriptions to be shown.

This is a required parameter.

### Qualifier 1: Class

**\*ALL** All class descriptions are shown.

#### *generic-name*

Specify the generic name of the class descriptions 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 class descriptions that have names with the same prefix as the generic name are shown.

*name* Specify the name of the class description 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  QUSRDIRDB  QUSRVI
QGGL       QSRVAGT   QUSRIJS    QUSRVxRxMx
QGGL38     QSYS2     QUSRINFSKR
QMGTC      QSYS2xxxxx QSRNOTES
QMGTC2     QS36F     QUSROND
QMPGDATA   QUSER38   QUSRPOSGS
QMOMDATA   QUSRADSM  QUSRPOSSA
QMOMPROC   QUSRBRM   QUSRPYMSVR
QPFRDATA   QUSRDIRCF QUSRDRARS
QRCL       QUSRDIRCL QUSRSYS
```

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

```
WRKCLS CLS(LIB01/ABC*)
```

This command allows you to display and work with a list of classes beginning with class objects whose names begin with 'ABC' that are in library LIB01.

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 Commands (WRKCMD)

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

Parameters  
Examples  
Error messages

The Work with Commands (WRKCMD) command allows you to display a list of commands from one or more libraries.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the commands to which you have some authority will be shown on the display.
- To perform operations on the commands, you must have \*USE authority to the command used by the operation, and the appropriate authority to the commands on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
CMD	Command	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Command	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Command (CMD)

Specifies the commands to be shown on the Work with Commands display.

This is a required parameter.

### Qualifier 1: Command

**\*ALL** All commands are shown on the Work with Commands display.

#### *generic-name*

Specify the generic name of the commands 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 commands that have names with the same prefix as the generic name are shown.

*name* Specify the name of the command 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	QUSRDIRDB	QUSRVI
QGGL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGGL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

```
WRKCMD  CMD(QGPL/DSP*)
```

This command allows you to work with a list of all commands in the QGPL library that start with 'DSP'.

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 Commitment Def (WRKCMTDFN)

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

Parameters  
Examples  
Error messages

The Work with Commitment Definitions (WRKCMTDFN) command is used to work with the commitment definitions on the system. A commitment definition is used to store information about commitment control when commitment control is started by the Start Commitment Control (STRCMTCTL) command. These commitment definitions may or may not be associated with an active job. Those not associated with an active job have been ended, but one or more of its logical units of work has not yet been completed.

The STATUS parameter can be used to subset the list of commitment definitions by their status. For example, a status value of \*PENDING displays a list of all commitment definitions that have local pending changes. A status value of \*RESYNC displays a list of all commitment definitions that are involved with resynchronizing their resources. Commitment definitions may be involved with resynchronizing resources in an effort to reestablish a synchronization point across the logical unit of work. A synchronization point is where all resources within a logical unit of work are in consistent state. A status value of \*UNDECIDED displays a list of all commitment definitions involved with a commit operation that are waiting to receive the decision to either commit or rollback. A status value of \*XOPEN displays a list of all commitment definitions associated with an X/Open global transaction.

The ASP group parameter can be used to subset the list of commitment definitions by the auxiliary storage pool (ASP) on which they reside.

The logical unit of work identifier (LUWID) parameter can be used when trying to find the commitment definition on the system which is working with a commitment definition on another system. The jobs containing these commitment definitions are communicating using an APPC conversation. An LUWID can be found by displaying the commitment definition on one system and then using it as input to the WRKCMTDFN command to find the corresponding commitment definition.

The duplicate job option (DUPJOB OPT) parameter specifies the action taken when duplicate jobs are found by this command. If duplicate jobs are found they can either be displayed in a list the user can select from or a message can be issued for each duplicate job found.

Top

---

### Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Single values: *, *ALL Other values: <i>Qualified job name</i>	Optional, Positional 1
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
STATUS	Status	*ALL, *PENDING, *RESYNC, *UNDECIDED, *XOPEN	Optional, Positional 2
ASPGRP	ASP group	<i>Name</i> , *ALLAVL, *SYSBAS	Optional
LUWID	Logical unit of work ID	<i>Character value</i> , *ALL	Optional
OUTPUT	Output	*, *PRINT	Optional

Keyword	Description	Choices	Notes
DUPJOB OPT	Duplicate job option	*SELECT, *MSG	Optional

Top

---

## Job name (JOB)

Specifies the names of the jobs (if any) whose commitment definitions are shown. If a job name is not qualified, all jobs by that name have their commitment definitions displayed. When \*ALL is not specified, only commitment definitions with lock scope \*JOB are shown, even though commitment definitions with lock scope \*TNSOBJ may be attached to threads in the specified jobs. To show commitment definitions with lock scope \*TNSOBJ, \*ALL must be specified.

A job identifier is a special value or a qualified name with up to three elements. For example:

```
*
*ALL
job-name
user-name/job-name
job-number/user-name/job-name
```

\*  
\_ The commitment definitions which are associated with the job where the WRKCMTDFN command is issued are shown.

\*ALL Commitment definitions for all jobs on the system are shown.

*job-name*

Specify the name of the job which is associated with the commitment definitions to be shown.

*user-name*

Specify the name of the user which is associated with the commitment definitions to be shown.

*job-number*

Specify the number of the job which is associated with the commitment definitions to be shown.

Top

---

## Status (STATUS)

Specifies that only the commitment definitions with a status that matches the value specified on this parameter are listed.

\*ALL Commitment definitions with all status' are shown.

\*PENDING

Only commitment definitions with local pending changes are shown. A local pending change can be any of the following cases:

- One or more record level changes are pending,
- One or more object level changes are pending,
- An API resource is registered that does not allow save-while-active requests to perform normally or does not allow the independent ASP to be quiesced.

\*RESYNC

Only commitment definitions which are involved with resynchronizing resources are shown. A

commitment definition may be involved with resynchronization in an effort to reestablish a synchronization point. A synchronization point is the point where all resources are in a consistent state.

**\*UNDECIDED**

Only commitment definitions whose logical unit of work is in a state that is undecided are shown. A commitment definition is in an undecided state when the decision to either commit or rollback resources is unknown to the commitment definition.

**\*XOPEN**

Only commitment definitions associated with an X/Open global transaction are shown.

Top

---

## ASP group (ASP Group)

Specifies the Auxiliary Storage Pool (ASP) group of the commitment definitions to be shown.

**\*ALLAVL**

All commitment definitions in all online ASPs are shown.

**\*SYSBAS**

Only commitment definitions in the system ASP (ASP number 1) and basic ASPs (ASP numbers 2-32) are shown.

*auxiliary-storage-pool-group-name*

Only commitment definitions in the specified ASP group are shown.

Top

---

## Logical unit of work ID (LUWID)

Specifies the logical unit of work identifier of the commitment definition to be shown.

A logical unit of work identifier is a character string made up of three elements:

- Network-qualified logical unit (LU) name
- Instance number
- Sequence number

The network-qualified LU name consists of a character network ID with a maximum of 8 characters, a period delimiter, followed by a LU name with a maximum of 8 characters. The instance number is entered as a 12 character value, each character representing a single hexadecimal digit. The value must be entered in hexadecimal format. For example, X'123456789012'. The sequence number is a decimal value with values ranging from 1 through 65535. For example:

APPN.RCHASLGU.X'12578A3BDCFF'.23657

**\*ALL** Commitment definitions associated with all logical units of work are shown.

*generic\*-logical-unit-of-work-identifier*

Specify the generic name of a logical unit of work identifier. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, ABC\*. If a generic name is specified, all commitment definitions with logical unit of work identifiers that begin with the generic name are shown. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete logical unit of work identifier.

### *logical-unit-of-work-identifier*

Specify a maximum of 39 characters for the logical unit of work identifier associated with a commitment definition.

Top

---

## Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting display 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

---

## 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. For a batch job, a message is shown.

\*MSG A message is issued when duplicate jobs are found.

Top

---

## Examples

### Example 1: Limiting by Job

The following example will display a list of commitment definitions associated with the specified job. Of all the commitment definitions on the system, only those associated with the specified job will be listed.

```
WRKCMTDFN JOB(012345/WULF/WULFS1)
```

### Example 2: Limiting by Commitment Definitions Involved with Resynchronizing Resources

The following example will display a list of all of the commitment definitions on the system that are involved with resynchronizing their resources.

```
WRKCMTDFN JOB(*ALL) STATUS(*RESYNC)
```

### Example 3: Limiting by Commitment Definitions That Are Undecided

The following example will display a list of all of the commitment definitions on the system that are in an undecided state. The commitment definitions are in an undecided state when their logical unit of work state is either prepared or last-agent pending.

```
WRKCMTDFN JOB(*ALL) STATUS(*UNDECIDED)
```

#### **Example 4: Limiting by Commitment Definitions That Are Associated With An X/Open Global Transaction**

The following example will display a list of all the commitment definitions associated with an X/Open global transaction.

```
WRKCMTDFN JOB(*ALL) STATUS(*XOPEN)
```

#### **Example 5: Limiting by LUWID**

```
WRKCMTDFN JOB(*ALL) LUWID(APPN.RCHASL7E.X'11223344BDF' .*)
```

This command will display a list of all the commitment definitions whose logical unit of work ID begins with the specified generic value. Of all those commitment definitions on the system, only those whose logical unit of work id's begin with the generic value will be listed.

#### **Example 6: Limiting by System ASP**

```
WRKCMTDFN JOB(*ALL) ASPGRP(*SYSBAS)
```

This command will display a list of all the commitment definitions that reside on the system auxiliary storage pool (ASP number 1).

#### **Example 7: Limiting by ASP Device Description Name**

```
WRKCMTDFN JOB(*ALL) ASPGRP(IASP0035)
```

This command will display a list of all the commitment definitions that reside on the independent auxiliary storage pool (ASP) associated with ASP device description IASP0035.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF0941**

Job &3/&2/&1 no longer in system.

#### **CPF1069**

End of duplicate names.

#### **CPF1070**

Job &3/&2/&1 not found.

#### **CPF1071**

No authority to job &3/&2/&1.

#### **CPF83E5**

Not authorized to jobs.

**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 Connection Lists (WRKCNNL)

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

Parameters  
Examples  
Error messages

The Work with Connection Lists (WRKCNNL) command allows you to work with a connection list.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CNNL	Connection list	<i>Generic name, name, <u>*ALL</u></i>	Optional, Positional 1

Top

---

### Connection list (CNNL)

This is a required parameter.

Specifies the connection list to work with.

**\*ALL** All connection lists are worked with.

***generic-name***

Specify a generic name of connection lists to be worked with. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, all connection lists that have names with the same prefix as the generic connection list are displayed.

***name*** Specify a specific connection list name.

Top

---

### Examples

WRKCNNL CNNL(CHI\*)

This command displays the Work with Connection List panel to work with connection lists which have names that begin with 'CHI' and for which the user has authority.

Top

---

### Error messages

**\*ESCAPE Messages**

**CPF2625**

Not able to allocate object &1.

**CPF2634**

Not authorized to object &1.

**CPF266C**

Connection list &1 not found.

**CPF266D**

Program name &1 not found in system library.

**CPF266E**

Connection list &1 has been damaged.

[Top](#)

---

## Work with Contact Information (WRKCNTINF)

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

Parameters  
Examples  
Error messages

The Work with Contact Information (WRKCNTINF) command is used to work with the information that helps you contact, or be contacted by, various support centers. The contact information is supplied to you by your market support center and your service support center. This command shows a menu where you select a support function.

**Restriction:** To use this command, the user must be signed on as QSRV or QSRVBAS, or have \*ALLOBJ authority.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKCNTINF

This command displays the Work with Support Contact Information panel.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF8C84

Error detected while processing support contact data.

##### CPF8C96

Description is a required field.

##### CPF8C97

Description already exists in system directory.

[Top](#)



---

## Work with COS Descriptions (WRKCODS)

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

Parameters  
Examples  
Error messages

The Work with Class-of-Service Descriptions (WRKCODS) command provides an interactive interface to class-of-service description functions through the Work with Class-of-Service Descriptions display.

Top

---

### Parameters

Keyword	Description	Choices	Notes
COSD	Class-of-service description	<i>Generic name, name, <u>*ALL</u></i>	Optional, Positional 1

Top

---

### Class-of-service description (COSD)

Specifies the class-of-service description to work with.

**\*ALL** Work with all class-of-service descriptions.

*generic-COS-description-name*

Specify a generic class-of-service 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.

*COS-description-name*

Specify a specific class-of-service description.

Top

---

### Examples

```
WRKCODS COSD(MPLS*)
```

This command displays the Work with Class-of-Service Descriptions panel which shows entries for all class-of-service descriptions whose names start with 'MPLS'.

Top

---

### Error messages

None



---

## Work Comm Side Information (WRKCSI)

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

Parameters  
Examples  
Error messages

The Work with Communications Side Information (WRKCSI) command allows you to work with side information in the specified library or libraries. From the list of side information objects that is displayed, you can create, delete, change, print, or display entries.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CSI	Side information	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Side information	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

### Side information (CSI)

Specifies the object name for the side information object you want to work with.

This is a required parameter.

The possible values are:

**\*ALL** All lists of side information objects are shown.

***side-information-name***

Specify the name and library of the list of side information objects to be shown.

***generic\*-side-information-name***

Specify the generic name of side information objects to be shown. A generic name is a character string that contains one or more characters followed by an asterisk (\*). All side information objects with same prefix as the generic side information object are shown.

The possible library values are:

**\*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 job is searched to locate the side information object. If no current library entry exists in the library list, QGPL is used.

**\*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  QUSRDIRDB  QUSRVI
QGGL       QSRVAGT   QUSRIJS    QUSRVxRxMx
QGGL38     QSYS2     QUSRINFSKR
QMGTC      QSYS2xxxxx QSRNOTES
QMGTC2     QS36F     QUSROND
QMPGDATA   QUSER38   QUSRPOSGS
QMQMDATA   QUSRADSM  QUSRPOSSA
QMQMPROC   QUSRBRM   QUSRPYMSVR
QPFRDATA   QUSRDIRCF QUSRDRARS
QRCL       QUSRDIRCL QUSRSYS
```

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.

### *library-name*

Specify the library name where the side information object is located. Only the library named in this parameter is searched.

Top

---

## Examples

### Example 1: Displaying Information Objects

```
WRKCSI
```

This command displays all the side information objects that exist in any library in the library list. From the Work with Communications Side Information panel, you can work with the side information objects.

### Example 2: Displaying Objects the Begin with 'SIDE'

```
WRKCSI  CSI(QGPL/SIDE*)
```

This command displays all the side information objects that begin with the letters 'SIDE' and are located in library QGPL. From the Work with Communications Side Information panel, you can work with the side information objects.

Top



---

## Error messages

None

[Top](#)



---

## Work with Ctl Descriptions (WRKCTLD)

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

Parameters  
Examples  
Error messages

The Work with Controller Descriptions (WRKCTLD) command allows you to work with controller description functions through the Work with Controller Descriptions display.

Top

---

### Parameters

Keyword	Description	Choices	Notes
CTLD	Controller description	Generic name, name, <u>*ALL</u> , *CMN, *WS, *TAP, *LWS, *RWS, *VWS	Optional, Positional 1

Top

---

### Controller description (CTLD)

Specifies the name of the controller description.

**\*ALL** Work with all controller descriptions.

*generic-controller-description-NAME*

Specify a generic controller 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.

*controller-description-NAME*

Work with a specific controller description.

**\*CMN** Work with communications controller descriptions.

**\*WS** Work with work station controller descriptions.

**\*TAP** Work with tape controller descriptions.

**\*LWS** Work with local work station controller descriptions.

**\*RWS** Work with remote work station controller descriptions.

**\*VWS** Work with virtual (pass-through) work station controller descriptions.

Top

---

### Examples

WRKCTLD CTLD(\*LWS)

This command displays the Work with Controller Descriptions panel to work with entries for all local work station controllers to which you have authority.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Work with DB Files using IDDU (WRKDBFIDD)

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

Parameters  
Examples  
Error messages

The Work with Database Files Using the Interactive Data Definition Utility (IDDU) (WRKDBFIDD) command shows the Work with Database Files display. From this display, you can select options that allow you to create physical files or to enter data into a file.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, <u>*PRV</u> , *CURLIB	Optional

Top

---

### Library (LIB)

Specifies the name of the library containing the files.

**\*PRV** The files are located in the last library you worked with in IDDU. If this is the first time you worked with IDDU, your current library is used.

**\*CURLIB**

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

*library-name*

Specify the library where the files are located.

Top

---

### Examples

```
WRKDBFIDD DEPT245
```

This command displays the Work with Database Files panel and allows you to work with IDDU database files in the DEPT245 library.

Top

---

### Error messages

None

Top



---

## Work with DDM Files (WRKDDMF)

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

Parameters  
Examples  
Error messages

The Work with Distributed Data Management Files (WRKDDMF) command shows a list of DDM files. From this list, you can:

- Change DDM files
- Delete DDM files
- Display details of DDM files
- Create DDM files
- Print a list of the DDM files
- Print details of DDM files

Top

---

### Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALL, *ALLUSR</i>	
OUTPUT	Output	<i>*, *PRINT</i>	Optional, Positional 2

Top

---

### File (FILE)

Specifies the name and library of the DDM files to be selected. A generic DDM file name or \*ALL can be specified.

The possible file values are:

**\*ALL** All files in the specified library (or all libraries identified in the library qualifier to which the user has access) are listed.

#### *file-name*

Specify the name of the DDM file to be selected. If \*LIBL or \*USRLIBL is specified as the library name, all DDM files found with the specified name are listed.

#### *generic\*-file-name*

Specify the generic name of the DDM files to be selected. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified on this parameter, all files that have names with the same prefix as the generic file are selected.

#### 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	#RPLIB	
#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	QUSRDIRB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

---

## 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.

Top



---

## Examples

WRKDDMF

This command shows the Work with DDM Files panel.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Work with Device Descriptions (WRKDEVD)

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

Parameters  
Examples  
Error messages

The Work with Device Descriptions (WRKDEVD) command is used to display and to work with device description functions through the Work with Device Descriptions panel.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEVD	Device description	<i>Generic name, name, *ALL, *ASP, *CMN, *DKT, *DSP, *LCLDSP, *RMTDSP, *VRTDSP, *LOC, *MLB, *OPTMLB, *TAPMLB, *OPT, *NWSH, *PRT, *LANPRT, *LCLPRT, *RMTTPRT, *VRTPRT, *CRP, *TAP, *SNPT</i>	Optional, Positional 1
RMTLOCNAME	Remote location	<i>Generic name, name, *NONE</i>	Optional

Top

---

### Device description (DEVD)

Specifies the name of the device description.

#### \*ALL

Work with all device descriptions.

#### \*ASP

Work with auxiliary storage pool devices.

#### \*CMN

Work with communications devices.

#### \*CRP

Work with cryptographic devices.

#### \*DKT

Work with diskette unit (drive) devices.

#### \*LCLDSP

Work with local display station devices.

**\*RMTDSP**

Work with remote display station devices.

**\*VRTDSP**

Work with virtual (pass-through) display station devices.

**\*LOC**

Work with devices at a specific remote location.

**\*MLB**

Both optical and tape media library devices are shown.

**\*OPTMLB**

Optical media library devices are shown.

**\*TAPMLB**

Tape media library devices are shown.

**\*OPT**

Optical disk devices are shown.

**\*NWSH**

Work with network server host adapter devices.

**\*PRT**

Work with all printer devices.

**\*LANPRT**

Printer devices attached to a local area network (LAN) are shown.

**\*LCLPRT**

Work with local printer devices.

**\*RMTprt**

Work with remote printer devices.

## **\*VRTPRT**

Work with virtual (pass-through) printer devices.

## **\*TAP**

Work with tape unit (drive) devices.

## **\*SNPT**

Work with SNA pass-through devices.

### *generic\*-device-description-name*

Specify a generic device 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.

### *device-description-name*

Work with a specific device description.

Top

---

## **Remote location (RMTLOCNAME)**

Specifies the remote location name for those device descriptions that are to be displayed. Only those device descriptions with the specified remote location name are listed on the Work with Device Descriptions display. This parameter is required if \*LOC is specified for the **Device description (DEVD)** parameter. It is not a valid parameter if any value other than \*LOC is specified for the DEVD parameter.

### **\*NONE**

If \*NONE is specified, all of the devices with a remote location name of \*NONE can be worked with.

**Note:** \*NONE should be specified if any value other than \*LOC is specified for the DEVD parameter.

### *remote-location-name*

Specify the remote location name, remote system name, or internet address.

### *generic-controller-description*

Specify a generic controller description.

**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.

[Top](#)

---

## Examples

```
WRKDEV D  DEVD(*LCLPRT)
```

This command displays the Work with Device Descriptions panel showing all the local printers to which you have authority.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Work with Device Tables (WRKDEVTBL)

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

Parameters  
Examples  
Error messages

The Work with Device Tables (WRKDEVTBL) command allows you to display and work with finance device tables and, once they are created, allows addition or deletion of device names in these tables. Several finance device tables can be defined, but each table must have a unique name.

An updated finance device table can be accessed by any finance job submitted after all changes are completed.

**Restriction:** Only the QFNC user profile is authorized to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DEVTBL	Device table	Name, *SELECT, <u>*ALL</u>	Optional, Positional 1
TEXT	Text 'description'	Character value, <u>*BLANK</u>	Optional

Top

---

### Device table (DEVTBL)

Specifies the name of a device table that contains 4704 or 3624 device names.

The possible values are:

\*ALL Displays all of the device tables currently defined.

**\*SELECT**

The Work with Device Tables display is shown. Through this display you can create, change, delete, or display device tables.

*device-table-name*

Specify the name of the device table that you want to work with.

Top

---

### Text 'description' (TEXT)

Specifies the text that briefly describes the object.

The possible values are:

**\*BLANK**

No text is specified.

*'description'*

Specify no more than 50 characters of text, enclosed in apostrophes.

Top

---

## Examples

### Example 1: Working With All Finance Device Tables

```
WRKDEVTBL  DEVTBL(*SELECT)
```

This command allows you to work with all of the finance device tables. The options are to add a new table, select an existing table for update, or create, delete, or display tables.

### Example 2: Working With One Finance Device Table

```
WRKDEVTBL  DEVTBL(DEVTBL1)
```

This command allows you to work with the device table DEVTBL1. The options are to create a new table, or change, display, or delete the table.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF8379

Error while processing file &1 in &2.

#### CPF8380

Error while opening file &1 in &2.

#### CPF8381

Error while closing file &1 in &2.

Top



---

## Work with Directory Entries (WRKDIRE)

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

Parameters  
Examples  
Error messages

The Work with Directory Entries (WRKDIRE) command allows you to work with a set of panels to view, add, change, and remove entries in the distribution directory. When the WRKDIRE command is entered, the system shows either one or all of the entries in the system distribution directory, depending on the parameters specified. If the parameter specified applies to more than one directory entry, the system displays a list of directory entries. If the parameter identifies a specific directory user, the system displays a list of entries for which that user has authority.

**Restriction:** You must have security administrator authority (\*SECADM) to update all entries in the directory. Restrictions on the data entries that can be updated apply when this command is run without \*SECADM authority. General access to view and print the directory is provided by the DSPDIRE command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
USRID	User identifier	Single values: <b>*ALL</b> Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
USER	User profile	<i>Name</i> , *CURRENT	Optional, Positional 2
CMDCHRID	Command character identifier	Single values: <b>*SYSVAL</b> , *DEVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

---

### User identifier (USRID)

Specifies the user ID and the address of user for whom the request is made. If the USRID parameter is specified, the USER parameter cannot be specified.

**\*ALL** All directory entries in the system distribution directory are shown. The entries are shown in alphabetical order by the user ID and address.

The possible user ID value is:

*user-ID*

Specify the user ID of the user for whom the directory entry is shown.

The possible user address value is:

### *user-ID*

Specify the user address of the user for whom the directory entry is shown.

Top

---

## User profile (USER)

Specifies, by user profile, which directory entry to display. If the user profile has no directory entries associated with it, an error message is sent. If the USER parameter is specified, the USRID parameter cannot be specified.

### \*CURRENT

The user profile under which the current job is running is used.

### *user-profile-name*

Specify the user profile of a directory entry shown. This is the 10-character profile used to sign on the system.

Top

---

## Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for data being specified as parameter values on this command. This character identifier (CHRID) is related to the display device used to specify the command. More information about CHRID processing is in the Application Display Programming book, SC41-5715.

### \*SYSVAL

The system determines the graphic character set and code page values for the command parameters from the QCHRID system values.

### \*DEV D

The system determines the graphic character set and code page values for the command parameter from the display device description where the command is entered. This option is valid only when specified from an interactive job. If this value is specified in an interactive CL program or a batch job, an error message is sent.

The possible character set value is:

### *graphic-character-set*

Specify the graphic character set values used to create the command parameters. Valid values range from 1 through 9999.

The possible code page value is:

### *code-page*

Specify the code page. Valid values range from 1 through 9999.

Top

---

## Examples

### Example 1: Using WRKDIRE with Administrator Authority

```
WRKDIRE  USRID(HURST NEWYORK)
```

Assume the user who is running this command has administrator authority. If the user ID and address of HURST NEWYORK exists in the directory, the Work with Directory Entries panel is shown listing all entries for HURST NEWYORK.

**Example 2: Using WRKDIRE with Security Administrator Authority**

```
WRKDIRE USER(JONES)
```

Assume the user who is running this command has security administrator authority. If the user profile of JONES exists in the directory, the Work with Directory Entries panel displays the entry with the user profile name of JONES. Multiple entries are displayed if JONES has more than one description.

**Example 3: Using WRKDIRE with Security Administrator Authority**

```
WRKDIRE
```

Assume the user who is running this command has security administrator authority. The Work with Directory Entries panel displays a listing of all entries in the directory.

**Example 4: Using WRKDIRE Without Security Administrator Authority**

```
WRKDIRE
```

Assume the user who is running this command does not have security administrator authority. The Change Your Directory Details panel is displayed for this user. A message appears on the message line of this panel indicating that this user is authorized only to change the user's directory entry.

[Top](#)

---

## Error messages

### \*ESCAPE Messages

**CPF9006**

User not enrolled in system distribution directory.

**CPF905C**

Error occurred trying to find a translation table.

**CPF9838**

User profile storage limit exceeded.

[Top](#)



---

## Work with Directory Locations (WRKDIRLOC)

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

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

The Work with Directory Locations (WRKDIRLOC) command provides a set of displays that allow an administrator to add, change, remove, display, print, and combine locations. When the WRKDIRLOC command is entered, the Work with Directory Locations display is shown with all the locations defined.

**Restriction:** The user of this command must have at least security administrator (\*SECADM) authority.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKDIRLOC

This command displays the Work with Directory Locations panel. The panel lists all of the locations currently defined.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Work with Dir Shadow Systems (WRKDIRSHD)

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

Parameters  
Examples  
Error messages

The Work with Directory Shadow Systems (WRKDIRSHD) command provides a set of displays that allows an administrator to view, add, change, and remove shadow system entries. The user can work with systems that are supplying the local system or are collecting from the local system.

**Restriction:** You must have security administrator (\*SECADM) authority to use this command.

Top

---

### Parameters

Keyword	Description	Choices	Notes
TYPE	Type of shadow system	* <u>SUPPLIER</u> , *COLLECTOR	Optional, Positional 1

Top

---

### Type of shadow system (TYPE)

Specifies the shadow systems with which the user wants to work.

#### \*SUPPLIER

The user is allowed to work with the systems supplying directory data to the local system. The Work with Directory Shadow Suppliers display is shown.

#### \*COLLECTOR

The user is allowed to work with the systems collecting from the local system. The Work with Directory Shadow Collectors display is shown.

Top

---

### Examples

```
WRKDIRSHD TYPE(*SUPPLIER)
```

This command displays the Work with Directory Shadow Suppliers panel. The Work with Directory Shadow Suppliers panel allows you to add, change, remove, and display supplier systems, to suspend shadowing from supplier systems, and to resume shadowing for supplier systems that have previously been suspended.

Top

---

## **Error messages**

### **\*ESCAPE Messages**

#### **CPF90A8**

\*SECADM special authority required to do requested operation.

#### **CPF905C**

Error occurred trying to find a translation table.

#### **CPF9838**

User profile storage limit exceeded.

#### **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 Documents (WRKDOC)

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

Parameters  
Examples  
Error messages

The Work with Documents (WRKDOC) command is a request to show the Work With Documents In Folders display or the Work With Non-text Document Data display.

From the Work With Documents In Folders display, you can select options to create, revise, copy, delete, view, print, rename, describe, print with options, send, check spelling, file document remotely, paginate, and work with document authority.

From the Work With Non-text Document Data display, you can select an option to copy, delete, and rename non-text data, such as graphs and images.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DOC	Document	<u>*ALL</u> , *NONTXTDTA	Optional, Positional 1
FLR	Folder	<i>Character value</i> , <u>*PRV</u> , *SELECT	Optional, Positional 2

Top

---

### Document (DOC)

Specifies which display to show.

\*ALL The Work With Documents In Folders display is shown.

\*NONTXTDTA

The Work With Non-text Document Data display is shown. On this display, you can work with non-text data such as graphs or images.

Top

---

### Folder (FLR)

Specifies the name of the folder to be used on the Work With Documents In Folders display or Work With Non-text Document Data display.

\*PRV The name of the folder from your last session is used.

\*SELECT

A list of folders is displayed from which you can select a folder.

*folder-name*

Specify the name of the folder you want to work with on the specified display.

---

## Examples

WRKDOC DOC(\*ALL) FLR(\*SELECT)

This command displays the Work with Documents panel, and shows a list of folders from which to select the working folder.

---

## Error messages

### \*ESCAPE Messages

#### CPF90A8

\*SECADM special authority required to do requested operation.

#### CPF9024

System cannot get correct record to finish operation.

#### 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.

#### CPF9847

Error occurred while closing file &1 in library &2.

---

## Work with Document Libraries (WRKDOCLIB)

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

Parameters  
Examples  
Error messages

The Work with Document Libraries (WRKDOCLIB) command allows you to manage the Document Interchange Architecture (DIA) library services available on remote systems in the network. A document library name and level of DIA supported can be specified for each distribution queue with a Document Library Services (DLS) queue type.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKDOCLIB

This command displays the Work with Document Libraries panel.

[Top](#)

---

### Error messages

#### \*ESCAPE Messages

##### CPF90A8

\*SECADM special authority required to do requested operation.

##### CPF9024

System cannot get correct record to finish operation.

##### 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 Document Print Queue (WRKDOCPRTQ)

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

Parameters  
Examples  
Error messages

The Work with Document Print Queue (WRKDOCPRTQ) command calls OfficeVision/400 to show the Work With Documents To Be Printed display. From this display, users can manage their printed output.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKDOCPRTQ

This command displays the Work with Documents to be Printed panel.

[Top](#)

---

### Error messages

None

[Top](#)



---

## Work with DSNX/PC Queues (WRKDPCQ)

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

Parameters  
Examples  
Error messages

The Work with DSNX/PC Distribution Queues (WRKDPCQ) command allows you to display and delete specific DSNX/PC queue entries from a DSNX/PC queue. These queues are where Distributed System Node Executive (DSNX) distributions that are bound for a personal computer (PC)(locally attached to the system and configured in the system directory as a DSXN-PC node) are held. A personal computer running DSNX-PC starts the DS-SEND function which sends the queued distributions to that requesting personal computer.

**Restriction:** To use this command, the user must be signed on as QPGMR or QSYSOPR, or have \*ALLOBJ authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
PCNODE	Distribution queue	Character value, <u>*ALL</u>	Optional, Positional 1
OUTPUT	Output	<u>_</u> , *PRINT	Optional

Top

---

### Distribution queue (PCNODE)

Specifies the names of the PC node's for which queue entries are shown.

\*ALL All the PC nodes that currently have queue entries are shown.

*PC-node-name*

Specify the name of the PC node that is to have its entries shown.

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.

Top

---

## Examples

WRKDPCQ

This command displays the Work with DSNX/PC Distribution Queues panel.

[Top](#)

---

## Error messages

None

[Top](#)



---

## Work with Disk Status (WRKDSKSTS)

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

Parameters  
Examples  
Error messages

The Work with Disk Status (WRKDSKSTS) command allows you to display and work with performance and status information for the disk units on the system.

Top

---

### Parameters

Keyword	Description	Choices	Notes
OUTPUT	Output	*_, *PRINT	Optional, Positional 1
RESET	Reset status statistics	*NO, *YES	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 disk statistics are reset to zero.

\*NO   The disk statistics are not reset. The measurement time interval is extended if a previous work with disk status command has started in the current job.

\*YES   The disk statistics are reset. A measurement time interval of zero is used.

Top

---

### Examples

```
WRKDSKSTS  OUTPUT(*PRINT)
```

This command prints a report on the performance and status information for the disk units on the system.

---

## 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.

---

## Work with Distribution Lists (WRKDSTL)

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

Parameters  
Examples  
Error messages

The Work with Distribution Lists (WRKDSTL) command provides a set of displays that you can use to view, create, add to, remove from, and delete distribution lists. A distribution list contains a list of directory entries used to simplify sending distributions to a group of users.

**Restriction:** You must have security administrator (\*SECADM) authority to change, delete, or rename another user's distribution list.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LSTID	List identifier	Single values: <b>*ALL</b> Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: List ID	<i>Character value</i>	
	Element 2: List ID qualifier	<i>Character value</i>	
CMDCHRID	Command character identifier	Single values: <b>*SYSVAL</b> , <b>*DEVVD</b> Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

---

### List identifier (LSTID)

Specifies by the two-part list ID which distribution lists are to be shown.

**\*ALL** All distribution lists in the system distribution directory are shown. The entries are displayed in alphabetical order by list ID.

The possible list identifier value is:

***list-ID***

Specify the list identifier (ID) of the distribution list.

The possible list qualifier value is:

***list-ID-qualifier***

Specify the list ID qualifier of the distribution list.

**Note:** The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services book, SC41-5410.

---

## Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values.

### Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

### Single values

#### \*SYSVAL

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

#### \*DEVVD

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

### Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

### Element 2: Code page

1-32767

Specify the code page to use.

### Note:

- Only the user ID and address, system name and group, department, and the X.400 O/R parameters are translated to the graphic character set identifier (GCID) specified on this parameter. All other parameter values that you specify are stored exactly as they are entered; the GCID value is stored with them.
- If this command is run interactively, the default GCID value is taken from the display device description. If it is run in batch, the default GCID value is taken from the QCHRID system value. You can override these values by specifying a specific character set and code page on this parameter.

---

## Examples

### Example 1: Showing a Distribution List

```
WRKDSTL  LSTID(WILL DISTLIST)
```

This command shows the Work with Distribution Lists panel with one entry, WILL DISTLIST. This example assumes that WILL DISTLIST exists.

### Example 2: Showing All Distribution Lists

WRKDSTL

This command shows the Work with Distribution Lists panel with a list of all distribution lists in the distribution directory.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9A83

Public nickname &1 not found.

#### CPF9A85

Nickname &1 not found.

#### CPF9024

System cannot get correct record to finish operation.

#### CPF905C

Error occurred trying to find a translation table.

#### CPF9052

List ID, &1 &2, cannot be found.

#### CPF9838

User profile storage limit exceeded.

Top



---

## Work with Distribution Queue (WRKDSTQ)

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

Parameters  
Examples  
Error messages

The Work with Distribution Queue (WRKDSTQ) command displays and controls the distribution requests on the Systems Network Architecture distribution services (SNADS) distribution queues. A detailed description of SNADS is in the SNA Distribution Services book, SC41-5410.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

### Restrictions:

- This command is shipped with public \*EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Before this command is run for the first time, the QSNADS subsystem must have been previously started to create the internal SNADS objects that this command uses.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

---

## Parameters

Keyword	Description	Choices	Notes
QUEUE	Distribution	<i>Element list</i>	Optional, Positional 1
	Element 1: Queue	<i>Character value, *ALL</i>	
OUTPUT	Output	<i>*, *PRINT</i>	Optional

Top

---

## Distribution (QUEUE)

Specifies the name of the distribution queue that is shown or printed. The queue specified must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

The possible values are:

**\*ALL** All SNADS distribution queues are shown or printed, in alphabetic order, by queue name.

*distribution-queue-name*

Specify a particular SNADS distribution queue.

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.

Top

---

## Examples

### Example 1: Working With All Distribution Queues

WRKDSTQ

This command allows the user to work with the status and contents of all distribution queues. The normal and high priority portions of each distribution queue are shown or printed.

### Example 2: Printing Information

WRKDSTQ OUTPUT(\*PRINT)

This command prints information on all distribution queues. The status of the normal and priority portions of the distribution queues are printed followed by a list of the distribution requests on the normal and high priority portions of each distribution queue.

Top

---

## Error messages

### \*ESCAPE Messages

**CPF8802**

Distribution queue &1 was not found.

**CPF8806**

Value &1 not valid for system name or system group.

**CPF8807**

Error occurred while using QSNADS journal.

**CPF8809**

Errors detected on SNADS internal queues.

**CPF8812**

Error occurred while processing distribution queues.

**CPF8813**

No entries exist.



**CPF8819**

Special value for distribution queue name not permitted.

**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.

[Top](#)



---

## Work with Data Areas (WRKDTAARA)

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

Parameters  
Examples  
Error messages

The Work with Data Areas (WRKDTAARA) command allows you to show a list of data areas from one or more libraries.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the data areas to which you have some authority will be shown on the display.
- To perform operations on the data areas, you must have \*USE authority to the command used by the operation, and the appropriate authority to the data areas on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DTAARA	Data area	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data area	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Data area (DTAARA)

Specifies the data areas to be shown.

This is a required parameter.

### Qualifier 1: Data area

**\*ALL** All data areas are shown.

#### *generic-name*

Specify the generic name of the data areas 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 data areas that have names with the same prefix as the generic name are shown.

*name* Specify the name of the data area to be that is 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  QUSRDIRDB  QUSRVI
QGGL38     QSRVAGT   QUSRIJS    QUSRVxRxMx
QGPL38     QSYS2     QUSRINFSKR
QMGTC      QSYS2xxxxx QSRNOTES
QMGTC2     QS36F     QUSROND
QMPGDATA   QUSER38   QUSRPOSGS
QMOMDATA   QUSRADSM  QUSRPOSSA
QMOMPROC   QUSRBRM   QUSRPYMSVR
QPFRDATA   QUSRDIRCF QUSRDRARS
QRCL       QUSRDIRCL QUSRSYS
```

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

```
WRKDTAARA  DTAARA(LIB01/ABC*)
```

This command allows you to display and work with a list of data areas beginning with the letters 'ABC' that are stored in library LIB01.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9809

Library &1 cannot be accessed.

**CPF9810**

Library &1 not found.

[Top](#)



---

## Work with Data Dictionaries (WRKDTADCT)

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

Parameters  
Examples  
Error messages

The Work with Data Dictionaries (WRKDTADCT) command shows a display that allows you to select options to create, change, delete, or print the contents of a data dictionary.

There are no parameters for this command.

[Top](#)

---

### Parameters

None

[Top](#)

---

### Examples

WRKDTADCT

This command displays the Work with Data Dictionaries panel.

[Top](#)

---

### Error messages

None

[Top](#)





---

## Work with Data Definitions (WRKDTADFN)

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

Parameters  
Examples  
Error messages

The Work with Data Definitions (WRKDTADFN) command allows you to work with data definitions. You can create, change, copy, delete, print, rename, or display where defined terms in a data dictionary are used. If the definition type is not specified, the Interactive Data Definition Utility (IDDU) Select Definition Type display is shown. The dictionary and type of definition to process is selected from this display.

Top

---

### Parameters

Keyword	Description	Choices	Notes
DTADCT	Data dictionary	Name, <u>*PRV</u>	Optional, Positional 1
DFNTYPE	Definition type	<u>*ALL</u> , *FILE, *RCDFMT, *FLD	Optional

Top

---

### Data dictionary (DTADCT)

Specifies the data dictionary to use.

\*PRV The last data dictionary you worked with in IDDU is used.

*data-dictionary-name*

Specify the data dictionary name to use.

Top

---

### Definition type (DFNTYPE)

Specifies the data definition type to use.

\*ALL Allows you to select the definition type and data dictionary from a list of all the data dictionaries and definition types.

\*FILE Allows you to work with file definitions in the data dictionary specified.

\*RCDFMT  
Allows you to work with record format definitions in the data dictionary specified.

\*FLD Allows you to work with field definitions in the data dictionary specified.

Top

---

## Examples

WRKDTADFN DFNTYPE(\*FILE)

This command allows you to work with file definitions in the data dictionary you worked with last.

[Top](#)

---

## Error messages

None

[Top](#)

---

## Work with Data Queues (WRKDTAQ)

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

Parameters  
Examples  
Error messages

The Work with Data Queues (WRKDTAQ) command allows you to show a list of available data queues from one or more libraries.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the data queues to which you have some authority will be shown on the display.
- To perform operations on the data queues, you must have \*USE authority to the command used by the operation, and the appropriate authority to the data queues on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
DTAQ	Data queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data queue	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	

Top

---

## Data queue (DTAQ)

Specifies a list of data queues to be shown. .\*

This is a required parameter.

### Qualifier 1: Data queue

**\*ALL** All data queues are shown.

#### *generic-name*

Specify the generic name of the data queues 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 data queues that have names with the same prefix as the generic name are shown.

*name* Specify the name of the data queue 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	QUSRDIRDB	QUSRVI
QGGL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGGL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

```
WRKDQAQ DTAQ(LIB01/ABC*)
```

This command allows you to display and work with a list of data queues beginning with the letters 'ABC' that are stored in library LIB01.

Top

---

## Error messages

### \*ESCAPE Messages

#### CPF9809

Library &1 cannot be accessed.

**CPF9810**

Library &1 not found.

[Top](#)



---

## Work with Edit Descriptions (WRKEDTD)

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

Parameters  
Examples  
Error messages

The Work with Edit Descriptions (WRKEDTD) command allows you to show a list of edit descriptions.

### Restrictions:

- Only the edit descriptions to which you have some authority will be shown on the display.
- To perform operations on the edit descriptions, you must have use (\*USE) authority to the command used by the operation, and the appropriate authority to the edit descriptions on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
EDTD	Edit description	<i>Qualifier list</i>	Required, Positional 1
	Qualifier 1: Edit description	<i>Generic name, name, *ALL</i>	

Top

---

## Edit description (EDTD)

Specifies the edit description to be shown.

**\*ALL** All edit descriptions are shown.

### *generic-name*

Specify the generic name of the edit descriptions 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 edit descriptions that have names with the same prefix as the generic name are shown.

***name*** Specify the name of the edit description to be shown.

Top

---

## Examples

WRKEDTD EDTD(ABC\*)

This command allows you to display and work with a list of edit descriptions beginning with the letters 'ABC'.

Top

---

## Error messages

None

[Top](#)



---

## Work with Environment Var (WRKENVVAR)

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

Parameters  
Examples  
Error messages

The Work with Environment Variables (WRKENVVAR) command can be used to show a list of the environment variables on the Work with Environment Variables display. From this display, you can select options to add, to change, to remove, to display the details of, or to print the environment variables.

**Restriction:** You must have \*JOBCTL special authority to use this command to add, change, or remove system-level environment variables.

Top

---

### Parameters

Keyword	Description	Choices	Notes
LEVEL	Level	*JOB, *SYS	Optional

Top

---

### Level of the environment variable. (LEVEL)

Specifies the level of the environment variable.

The possible values are:

\*JOB Work with job-level environment variables.

\*SYS Work with system-level environment variables.

Top

---

### Examples

#### Example 1: Work with Job-level Environment Variables

WRKENVVAR

This command allows you to display and work with all job-level environment variables.

#### Example 2: Work with System-level Environment Variables

WRKENVVAR LEVEL(\*SYS)

This command allows you to display and work with all system-level environment variables.

Top

---

## Error messages

None

[Top](#)

---

## Work with Files (WRKF)

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

Parameters  
Examples  
Error messages

The Work with Files (WRKF) command shows a list of files and allows you to copy, delete, save, and restore files, and to display file descriptions.

### Restrictions:

- Only the libraries to which you have use (\*USE) authority will be searched.
- Only the files to which you have some authority will be shown on the display.
- To perform operations on the files, you must have \*USE authority to the command used by the operation, and the appropriate authority to the files on which the operation is to be performed.

Top

---

## Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL</i>	
FILEATR	File attributes	<i>*ALL, BSCF38, CMNF38, DDMF, DFU, DFUEXC, DFUNOTEXC, DKTE, DSPE, DSPF38, ICFE, LE, LF38, MXDF38, PF, PF38, PRTE, PRTF38, SAVE, TAPF</i>	Optional, Positional 2

Top

---

## File (FILE)

Specifies the files to be shown on the Work with Files display.

This is a required parameter.

### Qualifier 1: File

**\*ALL** All files are shown.

### *generic-name*

Specify the generic name of the files 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 files that have names with the same prefix as the generic name are shown.

*name* Specify the name of the file 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	QUSRDIRB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

---

## File attributes (FILEATR)

Specifies the type of file whose attributes are to be shown.

\*ALL Files with all attributes are shown.

### BSCF38

A list of binary synchronous communication (BSC) communications files for a System/38 is shown.

### CMNF38

A file that is used to read data from, or write data to a logical unit (LU1) or advanced program-to-program communications (APPC) device and that allows the user to define the format of the data on the logical unit or device. A communications (CMNF38) file is a device file that is either created in the System/38 environment or migrated from a System/38 to support a communications device.

**DDMF**

A list of the Distributed Data Management (DDM) files is shown.

**DFU** A list of data file utility (DFU) files is shown.

**DFUEXEC**

A list of the System/38 data file utility (DFU) files which can be run using the System/38 data file utility is shown.

**DFUNOTEXEC**

A list of the System/38 data file utility (DFU) files which cannot be run using the System/38 data file utility is shown.

**DKTF** A list of diskette files is shown.

**DSPF** A list of display files is shown.

**DSPF38**

A list of display files for a System/38 is shown.

**ICFF** A list of Interactive Communications Function (ICF) files is shown.

**LF** A list of logical files is shown.

**LF38** A list of logical files for a System/38 is shown.

**MXDF38**

A list of mixed device files for a System/38 is shown.

**PF** A list of physical files is shown.

**PF38** A list of physical files for a System/38 is shown.

**PRTF** A list of printer files is shown.

**PRTF38**

A list of printer files for a System/38 is shown.

**SAVF** A list of save files is shown.

**TAPF** A list of tape files is shown.

Top

---

## Examples

**WRKF** FILE(X/PAY)

This command allows you to work with the file named PAY which is located in library X.

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 Function Usage (WRKFCNUSG)

**Where allowed to run:** Interactive environments (\*INTERACT  
\*IPGM \*IREXX \*EXEC)  
**Threadsafe:** Yes

Parameters  
Examples  
Error messages

The Work with Function Usage (WRKFCNUSG) command shows a list of function identifiers and allows you to change or display specified functions.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FCNID	Function ID	Generic name, name, <u>*ALL</u>	Optional, Positional 1

Top

---

### Function ID (FCNID)

Specifies the function ID of the functions that are listed on the Work with Function Usage display.

**\*ALL** All function identifiers are listed.

**generic-name**

Specify the generic name of the function identifiers to be listed. A generic name is a character string that contains one or more characters followed by an asterisk (\*). If a generic name is specified, all function identifiers that have names with the same prefix as the generic name are listed.

**name** Specify the name of the function ID to be listed.

Top

---

### Examples

```
WRKFCNUSG FCNID(QIBM_SERVICE*)
```

This command shows the Work with Function Usage panel listing all functions with names starting with QIBM\_SERVICE.

Top

---

### Error messages

**\*ESCAPE Messages**

**CPF3CDA**

Registration facility repository not available for use.





---

## Work with Folders (WRKFLR)

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

Parameters  
Examples  
Error messages

The Work with Folders (WRKFLR) command allows you to display and work with the word processing function of OfficeVision to show the Work with Folders display. From this display, you can optionally create, delete, rename, describe entries; put security on a folder; work with documents; or work with folder authority.

Top

---

### Parameters

Keyword	Description	Choices	Notes
FLR	Folder	Character value, <u>*ALL</u>	Optional, Positional 1

Top

---

### Folder (FLR)

Specifies the name of the folder used on the Work with Folders display.

**\*ALL** A list that consists of all first level folders is displayed.

*folder-name*

Specify the name of the folder that contains the folders to display.

Top

---

### Examples

WRKFLR FLR(\*ALL)

This command allows you to utilize the Work with Folders display. A list of all folders you are authorized to use is shown.

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  
North Castle Drive  
Armonk, NY 10504-1785  
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 106-0032, 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 YBWA  
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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Programming interface information

This i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

---

## 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  
Distributed Relational Database Architecture  
Domino  
DRDA  
IBM  
Infoprint  
InfoWindow  
i5/OS  
iSeries  
Integrated Language Environment  
Lotus  
LPDA  
OfficeVision  
Print Services Facility  
RPG/400  
System i  
System x  
SystemView  
System/36  
TCS  
Tivoli  
WebSphere  
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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, or service names may be trademarks or service marks of others.

---

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**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, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

---

## Code license and disclaimer information

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.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.





Printed in USA