IBM Systems - iSeries

i5/OS Commands
Starting with SNDRPY (Send Reply)

Version 5 Release 4

IBM Systems - iSeries

i5/OS Commands
Starting with SNDRPY (Send Reply)

Version 5 Release 4

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 513.

Second Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of i5/OS (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Send Reply (SNDRPY) 1
Parameters 1
Message key (MSGKEY) 1
Message queue (MSGQ) 2
Reply (RPY) 2
Remove message (RMV) 2
Reject default reply (RJTDFTRPY) 3
Coded character set ID (CCSID) 3
Examples 3
Error messages 3

Send Service Request (SNDSRVRQS) . . 5
Parameters 5
Action (ACTION) 5
Remote control point (RMTCPNAME) 5
Remote network identifier (RMTNETID) 6
Problem identifier (PRBID) 6
Examples 6
Error messages 7

Send TIE File (SNDTIEF) 9
Parameters 9
File (FILE) 9
Member (MBR) 10
User ID (TOUSRID) 10
Node (TONODE) 10
To file (TOFILE) 10
Type of contents of file (TYPE) 10
Priority (PTY) 11
Examples 11
Error messages 11

Send User Message (SNDUSRMSG) . . 13
Parameters 13
Message text, or (MSG) 14
Message identifier (MSGID) 14
Message file (MSGF) 14
Message data field values (MSGDTA) 15
Valid reply values (VALUES) 15
Default reply value (DFT) 15
Message type (MSGTYPE) 16
To message queue (TOMSGQ) 16
To user profile (TOUSR) 16
CL var for message reply (MSGRPY) 17
Translate table (TRNTBL) 17
Coded character set ID (CCSID) 17
Examples 18
Error messages 18

Display Mounted FS Information (STATFS) 19
Parameters 19
Object (OBJ) 19
Output (OUTPUT) 19
Examples 20
Error messages 20

Start Agent Services (STRAGTSRV) . . 21
Parameters 21
Preferences file directory (PREFDIR) 21
Home directory (HOMEDIR) 21
Additional classpath (CLASSPATH) 22
User profile for SBMJOB (SBMJOBUSER) 22
Examples 22
Error messages 23

Start ASP Balance (STRASPBAL) . . . 25
Parameters 26
Balance type (TYPE) 26
Auxiliary storage pool ID (ASP) 26
ASP device (ASPDEV) 27
Storage unit (UNIT) 27
Time limit (TIMLMT) 27
Examples 28
Error messages 28

Start CHT Server (STRCHTSVR) 31
Parameters 31
Server (SERVER) 31
Request response timeout (RSPTIMO) 32
Authorization list (AUTL) 32
Node (NODE) 32
Examples 33
Error messages 33

Start Cleanup (STRCLNUP) 35
Parameters 35
Option (OPTION) 35
Examples 36
Error messages 36

Start Cluster Node (STRCLUNOD) . . . 39
Parameters 39
Cluster (CLUSTER) 40
Node identifier (NODE) 40
Examples 40
Error messages 40

Start Communications Server (STRCMNSVR) 41
Parameters 41
Number of servers (NBRPASTHR) 41
Examples 41
Error messages 41

Start Communications Trace (STRCMNTRC) 43

© Copyright IBM Corp. 1998, 2006

Parameters 44
Configuration object (CFGOBJ) 45
Type (CFGTYPE) 45
Buffer size (MAXSTG) 45
Data direction (DTADIR) 45
Trace full (TRCFULL) 46
Number of user bytes to trace (USRDTA) 46
Communications trace options (CMNTRCOPTS) . . 46
DDI trace options (DDITRCOPTS) 47
Remote controller (RMTCTL) 47
Remote MAC address (RMTMAC) 47
Remote SAP (RMTSAP) 48
Local SAP (LCLSAP) 48
IP protocol number (IPPCLNUM) 48
Remote IP address (RMTIPADR) 48
LMI trace options (LMITRCOPTS) 48
NWS trace options (NWSTRCOPTS) 49
Watch for message (WCHMSG) 49
Watched message queue (WCHMSGQ) 50
Watched job (WCHJOB) 50
Watch for LIC log entry (WCHLICLOG) 51
Length of time to watch (WCHTIMO) 52
Trace program (TRCPGM) 53
Time interval (TRCPGMITV) 54
Trace description (TEXT) 54
Examples 55
Error messages 55

Start Commitment Control (STRCMTCTL) 57
Parameters 57
Lock level (LCKLVL) 58
Notify object (NFYOBJ) 58
Commitment definition scope (CMTSCOPE) . . . 59
Text 'description' (TEXT) 60
Journal (DFTJRN) 60
Journal entries to be omitted (OMTJRNE) 60
Examples 61
Error messages 61

Start Copy Screen (STRCPYSCN) 63
Parameters 63
Source device (SRCDEV) 64
Output device (OUTDEV) 64
Job queue (JOBQ) 64
File to receive output (OUTFILE) 65
Output member options (OUTMBR) 65
Examples 66
Error messages 66

Start Cluster Resource Group (STRCRG) 67
Parameters 68
Cluster (CLUSTER) 69
Cluster resource group (CRG) 69
Exit program data (EXITPGMDTA) 69
Examples 69
Error messages 69

Start Debug (STRDBG) 71
Parameters 72
Program (PGM) 72
Default program (DFTPGM) 73
Maximum trace statements (MAXTRC) 73
Trace full (TRCFULL) 74
Update production files (UPDPROD) 74
OPM source level debug (OPMSRC) 74
Service program (SRVPGM) 75
Class file (CLASS) 75
Display module source (DSPMODSRC) 76
Source debug program (SRCDBGPGM) 76
Unmonitored message program (UNMONPGM) . . 76
Examples 77
Error messages 78

Start Debug Server (STRDBGSVR) . . . 79
Parameters 79
Examples 79
Error messages 79

Start Database Monitor (STRDBMON) 81
Parameters 81
File to receive output (OUTFILE) 82
Output member options (OUTMBR) 82
Job name (JOB) 83
Type of records (TYPE) 83
Force record write (FRCRCD) 84
Run time threshold (RUNTHLD) 84
Include system SQL (INCSYSSQL) 84
Filter by database file (FTRFILE) 85
Filter by user profile (FTRUSER) 85
Filter by internet address (FTRINTNETA) 86
Comment (COMMENT) 86
Examples 86
Error messages 87

Start Data Base Reader (STRDBRDR) 89
Parameters 89
Data base file (FILE) 90
Member (MBR) 90
Job queue (JOBQ) 90
Queue for reader messages (MSGQ) 91
Reader (RDR) 91
Examples 91
Error messages 91

Start DFU (STRDFU) 93
Error messages for STRDFU 93
Parameters 93
Option (OPTION) 93
DFU program (DFUPGM) 94
Data base file (FILE) 94
Member (MBR) 94
Examples 94
Error messages 95

Start Directory Shadowing (STRDIRSHD) 97
Parameters 97

Examples 97
Error messages 97

Start DNS Query (STRDNSQRY) 99
Parameters 102
Host (HOSTNAME) 103
Domain Name Server (DMNNAMSVR) 103
Examples 103
Error messages 104

Start Disk Reorganization (STRDSKRGZ) 105
Parameters 105
Auxiliary storage pool ID (ASP) 105
ASP device (ASPDEV) 105
Time limit (TIMLMT) 106
Examples 106
Error messages 106

Start Education (STREDU) 109
Parameters 109
Examples 109
Error messages 109

Start 3270 Display Emulation (STREML3270) 111
Parameters 111
Emulation controller (EMLCTL) 112
Emulation device (EMLDEV) 113
Emulation location (EMLLOC) 113
Display device, batch only (DSPDEV) 113
Page Up (Roll Down) key (PAGEUP) 113
Page Down (Roll Up) key (PAGEDOWN) 114
Test Request key (TESTREQ) 114
Cursor Select key (CSRSLT) 115
SNA DBCS 3270PC emulation (IGCEMLPC) . . . 115
Emulation printer device (EMLPRTDEV) 115
Timeout wait for host (INZWAIT) 115
Numeric lock keyboard (NUMLCK) 116
Handle nulls (NULLS) 116
Host signon/logon command (LOGON) 116
Wait response (WAITRSP) 117
End emulation conditions (ENDCOND) 117
Attention emulation menu (ATNEMLMNU) . . . 118
Function key program (FKEYPGM) 118
Keyboard language type (KBDTYPE) 119
Outgoing translation table (TRNTBLOUT) 120
Incoming translation table (TRNTBLIN) 120
Configuration entry (EMLCFGE) 121
Examples 121
Error messages 121

Start EPM Environment (STREPMENV) 125
Error messages for STREPMENV 125
Parameters 125
Environment Name (EPMENV) 125
Environment Program (ROOTPGM) 125
Examples 126
Error messages 126

Start Font Management Aid (STRFMA) 127
Error messages for STRFMA 127
Parameters 127
Examples 127
Error messages 127

Start Host Server (STRHOSTSVR) . . 129
Error messages for STRHOSTSVR 129
Parameters 130
Server type (SERVER) 130
Required protocol (RQDPCL) 131
Examples 131
Error messages 131

Start IDDU (STRIDD) 133
Parameters 133
Examples 133
Error messages 133

Start IP over SNA Interface (STRIPSIFC) 135
Parameters 135
Internet address (INTNETADR) 135
Examples 135
Error messages 135

Start ITF (STRITF) 137
Parameters 137
Remote location (RMTLOCNAME) 137
Examples 137
Error messages 137

Start Journal (STRJRN) 139
Parameters 140
Objects (OBJ) 140
File identifier (OBJFID) 141
Journal (JRN) 141
Directory subtree (SUBTREE) 141
Name pattern (PATTERN) 141
New objects inherit journaling (INHERIT) 142
Images (IMAGES) 142
Omit journal entry (OMTJRNE) 142
Examples 143
Error messages 144

Start Journal Access Path (STRJRNAP) 145
Parameters 146
Journaled file (FILE) 146
Journal (JRN) 146
Examples 147
Error messages 147

Start Journal Object (STRJRNOBJ) 149
Parameters 150
Object (OBJ) 150
Object type (OBJTYPE) 150
Journal (JRN) 151
Images (IMAGES) 151

Examples 151
Error messages 151

Start Journal Physical File (STRJRNPF) 153
Parameters 154
Physical file to be journaled (FILE) 154
Journal (JRN) 154
Record images (IMAGES) 155
Journal entries to be omitted (OMTJRNE) 155
Examples 155
Error messages 155

Start Job Log Server (STRLOGSVR) 159
Parameters 159
Number of servers (NBRSVR) 159
Examples 159
Error messages 160

Start Mode (STRMOD). 161
Parameters 161
Remote location (RMTLOCNAME) 161
Device (DEV) 161
Mode (MODE) 162
Local location (LCLLOCNAME) 162
Remote network identifier (RMTNETID) 162
Examples 163
Error messages 163

Start Mail Server Framework (STRMSF) 165
Parameters 165
How to process mail messages (MSGOPT). . . . 165
Number of MSF jobs (NBRMSFJOB) 165
Examples 166
Error messages 166

Start NFS Server (STRNFSSVR) . . . 167
Parameters 167
Server daemon (SERVER) 168
Number of server daemons (NBRSVR) 168
Number of block I/O daemons (NBRBIO) 168
Retrieve RPC registration (RTVRPCREG) 169
Timeout for start of daemon (STRJOBTIMO) . . . 169
Examples 169
Error messages 169

Start NetWare Connection (STRNTWCNN) 171
Parameters 171
Server (SERVER) 172
Server type (SVRTYPE) 172
NDS tree (NDSTREE) 172
Authorized user profile (AUTUSR) 172
NDS context (NDSCTX) 173
NetWare user (NTWUSER) 173
Password (PASSWORD) 173
Authorized job (AUTJOB) 173
Connection type (CNNTYPE) 174
Connection idle time (CNNIDLTIME) 174
Examples 174
Error messages 174

Start Object Conversion (STROBJCVN) 175
Parameters 175
Library (LIB) 175
Object type (OBJTYPE) 176
Priority (PTY) 177
Examples 177
Error messages 177

Start Pass-Through (STRPASTHR) 179
Parameters 179
Remote location (RMTLOCNAME) 179
APPC device (CNNDEV) 180
Virtual controller (VRTCTL) 180
Virtual display device (VRTDEV) 181
Mode (MODE) 181
Local location (LCLLOCNAME) 181
Remote network identifier (RMTNETID) 181
System request program (SRQ10PGM) 182
User profile (RMTUSER) 182
User password (RMTPWD). 183
Initial program to call (RMTINLPGM) 183
Initial menu (RMTINLMNU) 183
Current library (RMTCURLIB). 184
Display option (PASTHRSCN). 184
Examples 184
Error messages 185

Start PC Command (STRPCCMD) . . . 189
Error messages for STRPCCMD 189
Parameters 189
PC command (PCCMD) 189
Pause (PAUSE) 189
Examples 190
Error messages 190

Start iSeries Access Organizer (STRPCO) 191
Error messages for STRPCO 191
Parameters 191
Text Assist (PCTA) 191
Examples 191
Error messages 191

Start Performance Explorer (STRPEX) 193
Parameters 193
Session ID (SSNID) 193
Option (OPTION) 194
Definition (DFN) 194
Filter (FTR) 194
Examples 194
Error messages 195
Error messages for RMVPEXDFN 195

Start Performance Collection (STRPFRCOL) 197
Parameters 197
Default collection profile (COLPRF) 197
Cycle collection (CYCCOL) 198
Examples 198
Error messages 198

Start Performance Trace (STRPFRTRC) 201
Parameters 201
Trace table size (SIZE) 201
Omit trace points (OMTTRCPNT) 202
Job types (JOBTYPE) 202
Job trace interval (JOBTRCITV) 203
Examples 203
Error messages 203

Start Program Export List (STRPGMEXP) 205
Parameters 205
Program level (PGMLVL) 205
Signature level check (LVLCHK) 205
Signature (SIGNATURE) 205
Examples 206
Error messages 206

Start Programmer Menu (STRPGMMNU) 207
Parameters 207
Source file (SRCFILE) 207
Source library (SRCLIB) 208
Object library (OBJLIB) 208
Job description (JOBD) 208
Allow changes (ALWUSRCHG) 208
Option 3 exit program (EXITPGM) 208
Delete option (DLTOPT) 209
Examples 210
Error messages 211

Start Program Profiling (STRPGMPRF) 213
Parameters 213
Examples 213
Error messages 213

Start Prestart Jobs (STRPJ) 215
Parameters 215
Subsystem (SBS) 215
Program (PGM) 216
Examples 216
Error messages 216

Start Printer Emulation (STRPRTEML) 219
Parameters 219
Emulation controller (EMLCTL) 220
Emulation device (EMLDEV) 220
Emulation location (EMLLOC) 221
Print device (PRTDEV) 221
Job name (JOB) 221
End Bracket eject (ENDBKTEJT) 221
Printer file (PRTFILE) 222
Timeout wait eject (TIMOUTEJT) 222
Defer printing spool output (DFRPRTOUT) . . . 222
Spool output (SPOOL) 223
Open printer file (OPNPRTF) 223
Print positions per line (NUMCOL) 223
Lines per page (NUMLIN) 224
Lines per inch (LPI) 224
Message queue (MSGQ) 224
Language character set (CHRSET) 225
Submit job (SBMJOB) 225
Job description (JOBD) 225
Outgoing translation table (TRNTBLOUT) 226
Job queue (JOBQ) 226
End emulation conditions (ENDCOND) 227
Acknowledge form feed (FORMFEED) 227
Configuration entry (EMLCFGE) 227
Examples 228
Error messages 228

Start Printer Writer (STRPRTWTR) 231
Parameters 231
Printer (DEV) 232
Output queue (OUTQ) 232
Queue for writer messages (MSGQ) 233
Form type options (FORMTYPE) 233
File separators (FILESEP) 234
Drawer for separators (SEPDRAWER) 234
Writer (WTR) 235
Auto-end options (AUTOEND) 235
Allow direct print (ALWDRTPRT) 235
Align page (ALIGN) 236
Initialize printer (INIT) 236
Spooled file (FILE) 236
Job name (JOB) 236
Spooled file number (SPLNBR) 237
Job system name (JOBSYSNAME) 237
Spooled file created (CRTDATE) 238
Starting page (PAGE) 238
Examples 238
Error messages 239

Start Query Management Proc (STRQMPRC) 241
Parameters 241
Source member (SRCMBR) 241
Source file (SRCFILE) 242
Relational database (RDB) 242
Connection Method (RDBCNNMTH) 242
User (USER) 242
Password (PASSWORD) 243
Naming convention (NAMING) 243
Allow information from QRYDFN (ALWQRYDFN) 243
Command source file (CMDSRCFILE) 243
Source member (CMDSRCMBR) 244
Display screens (ALWDSPLAY) 244
Examples 244
Error messages 245

Start Query Management Query (STRQMQRY) 247
Parameters 247
Query management query (QMQRY) 248
Output (OUTPUT). 248
Query management report form (QMFORM) . . . 249
File to receive output (OUTFILE). 249
Output member options (OUTMBR). 249
Date and time (DATETIME) 250
Page numbers (PAGNBR) 250
Relational database (RDB) 250
Connection Method (RDBCNNMTH) 251
User (USER). 251
Password (PASSWORD). 251
Naming convention (NAMING) 251
Allow information from QRYDFN (ALWQRYDFN) 252
Set variables (SETVAR) 252
Examples. 252
Error messages 253

Start Query (STRQRY) 255
Parameters 255
Examples. 255
Error messages 255

Start QSH (STRQSH) 257
Error messages for STRQSH 258
Parameters 258
Command (CMD) 258
Examples. 259
Error messages 259

Start Question and Answer (STRQST) 261
Parameters 261
Q/A database (QSTDB) 261
Lib containing Q/A database (LIB) 261
Examples. 262
Error messages 262

Start REXX Procedure (STRREXPRC) 263
Parameters 263
Source member (SRCMBR) 263
Source file (SRCFILE). 263
Parameters (PARM) 264
Command environment (CMDENV). 264
Exit program (EXITPGM) 265
Examples. 266
Error messages 266

Start Remote Support (STRRMTSPT) 267
Parameters 267
Device class (DEVCLS) 268
Display type (DSPTYPE) 268
Display model (DSPMODEL) 269
Address (STNADR) 269
User profile (USRPRF) 269
Resource name (RSRCNAME). 270
Remote location (RMTLOCNAME) 270
Local location (LCLLOCNAME) 270
Remote network identifier (RMTNETID) 271
Maximum length of request unit (MAXLENRU) 271
Data compression (DTACPR) 271
Inbound data compression (INDTACPR) 272
Outbound data compression (OUTDTACPR) . . . 273
Modem (MODEM) 273
Modem remote system (MDMRMTSYS) 274
Examples. 274
Error messages 275

Start Remote Writer (STRRMTWTR) 277
Parameters 277
Output queue (OUTQ) 278
Queue for writer messages (MSGQ) 278
Form type options (FORMTYPE) 279
Writer (WTR) 279
Auto-end options (AUTOEND) 280
Examples. 280
Error messages 280

Start S/36 Session (STRS36) 281
Parameters 281
Menu (MENU) 281
Current library (CURLIB) 281
Procedure or program (PRC) 282
Force menu (FRCMNU) 282
Examples. 282
Error messages 282

Start S/36 Procedure (STRS36PRC) 285
Parameters 285
Procedure (PRC) 285
Current library (CURLIB) 285
Procedure parameters (PARM). 286
Examples. 286
Error messages 286

Start Subsystem (STRSBS) 287
Parameters 288
Subsystem description (SBSD) 288
Examples. 288
Error messages 289

Start Search Index (STRSCHIDX) . . . 291
Parameters 291
Search index (SCHIDX) 291
Examples. 292
Error messages 292

Start Support Network (STRSPTN) 293
Parameters 293
Account (ACCOUNT) 293
User ID (SPTUSRID) 293
Password (SPTPWD) 294
Device description (FEADEV) 294
Destination application (DESTAPP) 294
Examples. 294
Error messages 294

Start Service Agent (STRSRVAGT) 295
Parameters . . . 295
Type (TYPE) . . . 295
Activation password (ACTPWD) . . . 296
Master password (MSTPWD) . . . 296
Examples . . . 296
Error messages . . . 296

Start Service Job (STRSRVJOB) . . . 297
Parameters . . . 297
Job name (JOB) . . . 297
Duplicate job option (DUPJOBOPT) . . . 298
Examples . . . 298
Error messages . . . 298

Start System Service Tools (STRSST) 299
Parameters . . . 299
Examples . . . 299
Error messages . . . 299

Start TCP/IP (STRTCP) . . . 301
Parameters . . . 303
Start application servers (STRSVR) . . . 304
Start TCP/IP interfaces (STRIFC) . . . 304
Start point-to-point profiles (STRPTPPRF) . . . 304
Start IPv6 (STRIP6) . . . 304
Examples . . . 305
Error messages . . . 305

Start TCP/IP Interface (STRTCPIFC) 307
Parameters . . . 307
Internet address (INTNETADR) . . . 307
Alias name (ALIASNAME) . . . 308
Examples . . . 308
Error messages . . . 308

Start Point-to-Point TCP/IP (STRTCPPTP) . . . 311
Parameters . . . 311
Configuration profile (CFGPRF) . . . 311
Script dialog output (OUTPUT) . . . 312
Restart (RESTART) . . . 312
Send inquiry message (SNDINQMSG) . . . 312
Autodelete configuration (AUTODLTCFG) . . . 313
Examples . . . 313
Error messages . . . 314

Start TCP/IP Server (STRTCPSVR) 315
Parameters . . . 315
Server application (SERVER) . . . 316
Restart server (RESTART) . . . 319
HTTP server (HTTPSVR) . . . 320
DNS server (DNSSVR) . . . 321
TCM server (TCMSVR) . . . 322
ASFTOMCAT server (TOMCATSVR) . . . 322
SNTP service (NTPSRV) . . . 323
Examples . . . 323
Error messages . . . 324

Start TIE Session (STRTIESSN) . . . 327
Parameters . . . 327
User ID (SPTUSRID) . . . 327
Password (SPTPWD) . . . 327
Account (ACCOUNT) . . . 327
Examples . . . 328
Error messages . . . 328

Start Trace (STRTRC) . . . 329
Parameters . . . 330
Session ID (SSNID) . . . 331
Job name (JOB) . . . 331
Job types (JOBTYPE) . . . 332
Maximum storage to use (MAXSTG) . . . 333
Trace full (TRCFULL) . . . 333
Trace type (JOBTRCTYPE) . . . 333
Trace type (TRCTYPE) . . . 334
Trace filter (TRCFTR) . . . 335
Restart after next IPL (RESTRIPL) . . . 335
Watch for message (WCHMSG) . . . 335
Watched message queue (WCHMSGQ) . . . 336
Watched job (WCHJOB) . . . 337
Watch for LIC log entry (WCHLICLOG) . . . 338
Length of time to watch (WCHTIMO) . . . 339
Trace program (TRCPGM) . . . 339
Time interval (TRCPGMITV) . . . 341
Examples . . . 341
Error messages . . . 342

Start Trap Manager (STRTRPMGR) 345
Parameters . . . 345
Forward Trap (FWDTRP) . . . 345
Examples . . . 345
Error messages . . . 346

Start Watch (STRWCH) . . . 347
Parameters . . . 347
Session ID (SSNID) . . . 348
Watch program (WCHPGM) . . . 348
Watch for message (WCHMSG) . . . 351
Watched message queue (WCHMSGQ) . . . 352
Watched job (WCHJOB) . . . 352
Watch for LIC log entry (WCHLICLOG) . . . 353
Examples . . . 354
Error messages . . . 355

Subroutine (SUBR) . . . 357
Parameters . . . 357
Subroutine (SUBR) . . . 357
Examples . . . 357
Error messages . . . 358

Transfer Batch Job (TFRBCHJOB) 359
Parameters . . . 360
Job queue (JOBQ) . . . 360
Routing data (RTGDTA) . . . 360
Request data or command (RQSDTA) . . . 360
Examples . . . 361
Error messages . . . 361

Transfer Control (TFRCTL) 363
Parameters 363
Program (PGM) 363
Parameter CL variable names (PARM) 364
Examples 364
Error messages 364

Transfer to Group Job (TFRGRPJOB) 365
Parameters 365
Group job (GRPJOB) 365
Initial group program (INLGRPPGM) 366
Special environment (SPCENV) 366
Restore display (RSTDSP) 366
Text 'description' (TEXT) 367
Examples 367
Error messages 367

Transfer Job (TFRJOB) 369
Parameters 370
Job queue (JOBQ) 370
Routing data (RTGDTA) 370
Request data or command (RQSDTA) 371
Examples 371
Error messages 371

Transfer Pass-Through (TFRPASTHR) 373
Parameters 373
To job (TOJOB) 373
Examples 374
Error messages 374

Transfer Secondary Job (TFRSECJOB) 375
Parameters 375
Examples 375
Error messages 375

Trace TCP/IP Route (TRACEROUTE) 377
Parameters 377
Remote system (RMTSYS) 378
Range of hops to probe (RANGE) 378
Probes sent per hop (PROBES) 378
Response wait time (WAITTIME) 378
Packet length (in bytes) (PKTLEN) 379
Output (OUTPUT) 379
Data queue (DTAQ) 379
Address version format (ADRVERFMT) 379
Source IP address (LCLINTNETA) 380
Base remote port (RMTPORT) 380
Lookup host names (NAMELOOKUP) 380
Probing protocol (PROBEPCL) 380
Allow fragmentation (FRAGMENT) 381
Examples 381
Error messages 382

Trace ASP Balance (TRCASPBAL) . . 383
Parameters 383
Trace option setting (SET) 384
ASP number (ASP) 384
ASP device (ASPDEV) 384
Time limit (TIMLMT) 385
Examples 385
Error messages 385

Trace Connection (TRCCNN) 387
Parameters 387
Trace option setting (SET) 389
Trace type (TRCTYPE) 389
Trace full (TRCFULL) 390
Trace table name (TRCTBL) 390
Size (SIZE) 390
Trace data to be formatted (FMTDTA) 391
Coded character set identifier (CCSID) 391
Job name (JOB) 391
Spooled file number (SPLNBR) 392
Job system name (JOBSYSNAME) 392
Spooled file created (CRTDATE) 392
TCP/IP data (TCPDTA) 393
Watch for message (WCHMSG) 394
Watched message queue (WCHMSGQ) 395
Watched job (WCHJOB) 395
Watch for LIC log entry (WCHLICLOG) 396
Length of time to watch (WCHTIMO) 397
Trace program (TRCPGM) 397
Time interval (TRCPGMITV) 399
Examples 399
Error messages 401

Trace CPI Communications (TRCCPIC) 403
Parameters 403
Trace option setting (SET) 404
Maximum storage to use (MAXSTG) 404
Trace full (TRCFULL) 404
User data length (DTALEN) 404
Output (OUTPUT) 405
Output file (OUTFILE) 405
Output member options (OUTMBR) 405
Examples 406
Error messages 406

Trace ICF (TRCICF) 409
Parameters 409
Trace option setting (SET) 409
Maximum storage to use (MAXSTG) 410
Trace full (TRCFULL) 410
User data length (DTALEN) 410
Output (OUTPUT) 410
Output file (OUTFILE) 411
Output member options (OUTMBR) 411
Examples 411
Error messages 412

Trace Internal (TRCINT) 415
Parameters 415
Trace option setting (SET) 418
Trace table name (TRCTBL) 419
Trace table size (SIZE) 419
Trace full (TRCFULL) 420
Trace type (TRCTYPE) 420

Job name (JOB) 423
Thread ID to include (SLTTHD) 424
Server type (SVRTYPE) 424
Task name (TASK) 425
Task number (TASKNBR) 425
Select trace points (SLTTRCPNT) 425
Omit trace points (OMTTRCPNT) 426
Stop trace points (STOPTRCPNT) 427
Job types (JOBTYPE) 427
Job trace interval (JOBTRCITV) 428
TCP/IP data (TCPDTA) 428
Sockets data (SCKDTA) 429
Device (DEV) 430
Controller (CTL) 431
Line (LIN) 432
Network interface (NWI) 432
Network Server (NWS) 433
Hardware Resource (RSRCNAME) 433
Device (OUTDEV) 434
Task information (TASKINF) 434
Output (OUTPUT) 434
File to receive output (OUTFILE) 434
Output member options (OUTMBR) 435
Watch for message (WCHMSG) 435
Watched message queue (WCHMSGQ) 436
Watched job (WCHJOB) 437
Watch for LIC log entry (WCHLICLOG) 438
Length of time to watch (WCHTIMO) 439
Trace program (TRCPGM) 439
Time interval (TRCPGMITV) 441
Examples 441
Error messages 442

Trace Job (TRCJOB) 445
Parameters 445
Trace option setting (SET) 446
Trace type (TRCTYPE) 446
Maximum storage to use (MAXSTG) 447
Trace full (TRCFULL) 447
Program to call before trace (EXITPGM) 447
Select procedures to trace (SLTPRC) 447
Thread ID to include (SLTTHD) 447
Output (OUTPUT) 448
File to receive output (OUTFILE) 448
Output member options (OUTMBR) 448
Examples 449
Error messages 449

Trace REXX (TRCREX) 451
Parameters 451
Trace option setting (SET) 451
Examples 452
Error messages 452

Trace TCP/IP Application (TRCTCPAPP) 453
Parameters 454
TCP/IP application (APP) 456
Trace option setting (SET) 457
Maximum storage for trace (MAXSTG) 457
Trace full action (TRCFULL) 458
Additional traces (ADLTRC) 458
Trace program (TRCPGM) 459
Trace title (TITLE) 461
User profile (USER) 462
Recipient mail address (MAILADR) 462
Recipient host name (HOST) 462
Remote network address (RMTNETADR) 462
Local network address (LCLNETADR) 463
Device description (DEVD) 464
Device type (DEVTYPE) 464
Trace point (TRCPNT) 464
Argument list (ARGLIST) 465
Virtual private network server (VPNSVR) 465
Certificate services type (CERTTYPE) 465
Domain name service (DNS) 466
PPP connection profile (PPPCNNPRF) 466
TCP/IP data to trace (TCPTRCDTA) 466
QOS trace type (QOSTRCTYPE) 466
HTTP server instance (HTTPSVR) 467
Trace level (TRCLVL) 467
Packet rules trace points (PKTTRCPNT) 467
Configuration object (CFGOBJ) 467
Type (CFGTYPE) 468
Watch for message (WCHMSG) 468
Watched message queue (WCHMSGQ) 469
Watched job (WCHJOB) 469
Watch for LIC log entry (WCHLICLOG) 470
Length of time to watch (WCHTIMO) 471
Time interval (TRCPGMITV) 471
Job name (JOB) 472
Examples 473
Error messages 474

Trace TCP/IP Route (TRCTCPRTE) 475
Parameters 475
Remote system (RMTSYS) 476
Range of hops to probe (RANGE) 476
Probes sent per hop (PROBES) 476
Response wait time (WAITTIME) 476
Packet length (in bytes) (PKTLEN) 477
Output (OUTPUT) 477
Data queue (DTAQ) 477
Address version format (ADRVERFMT) 477
Source IP address (LCLINTNETA) 478
Base remote port (RMTPORT) 478
Lookup host names (NAMELOOKUP) 478
Probing protocol (PROBEPCL) 478
Allow fragmentation (FRAGMENT) 479
Examples 479
Error messages 480

Remove Mounted FS (UNMOUNT) . . 481
Parameters 481
Type of file system (TYPE) 481
Directory mounted over (MNTOVRDIR) 482
Mounted file system (MFS) 482
Examples 482
Error messages 482

Update Data with Temp Program (UPDDTA) 485
Error messages for UPDDTA 485
Parameters 485
Data base file (FILE) 485
Member (MBR) 485
Examples 486
Error messages 486

Update Program (UPDPGM) 487
Parameters 487
Program (PGM) 488
Module (MODULE) 488
Replacement library (RPLLIB) 489
Bind service program (BNDSRVPGM) 490
Bound *SRVPGM library name (SRVPGMLIB) . . 490
Binding directory (BNDDIR) 491
Activation group (ACTGRP) 491
Creation options (OPTION). 492
Listing detail (DETAIL) 493
Examples 493
Error messages 493

Update PTF Information (UPDPTFINF) 495
Parameters 495
Examples 495
Error messages 495

Update Service Program (UPDSRVPGM) 497
Parameters 497
Service program (SRVPGM) 498
Module (MODULE) 499
Export (EXPORT) 499
Export source file (SRCFILE) 500
Export source member (SRCMBR) 500
Replacement library (RPLLIB) 501
Bind service program (BNDSRVPGM) 501
Bound *SRVPGM library name (SRVPGMLIB) . . 502
Binding directory (BNDDIR) 502
Activation group (ACTGRP) 503
Creation options (OPTION). 503
Listing detail (DETAIL) 504
Examples 504
Error messages 505

Update System Information (UPDSYSINF) 507
Parameters 507
Library (LIB) 507
Type of information (TYPE) 508
Examples 508
Error messages 508

Verify APPC Connection (VFYAPPCCNN) 509
Parameters 509
Remote location (RMTLOCNAME) 509
Mode (MODE) 509
Remote user ID (RMTUSER) 510
Remote password (RMTPWD). 510
Message mode (MSGMODE) 510
Packet length (in bytes) (PKTLEN) 511
Number of iterations (NBRITER) 511
Number of packets (NBRPKT). 511
Echo (ECHO) 511
Wait time (in seconds) (WAITTIME) 512
Examples 512
Error messages 512

Appendix. Notices 513
Trademarks 514
Terms and conditions. 515

Send Reply (SNDRPY)

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

Parameters
Examples
Error messages

The Send Reply (SNDRPY) command sends a reply message to the sender of an inquiry message. The message that is answered is the one having the specified message reference key that was received at the specified message queue.

A reply handling exit program can be registered in the system registration facility for exit point QIBM_QMH_REPLY_INQ. A reply handling exit program can accept, reject or replace a reply value. If the reply is rejected by an exit program, diagnostic message CPD2476 (Reply rejected by a reply handling exit program) is sent to the program running this SNDRPY command. This is followed by an escape message CPF2422 (Reply not valid). If the reply is replaced by an exit program, the send reply function sends a diagnostic message to itself. The message is CPD2479 (Reply handling exit program requested to replace a reply value). After the reply is sent, CPF2458 (Reply replaced by a reply handling exit program) will be sent as a diagnostic message and a status message to the program running this SNDRPY command. The status message can be monitored if the program needs to be aware of the condition when a reply value other than what was specified was sent.

If the specified message queue is not allocated to the job in which this command is entered, it is implicitly allocated by this command for the duration of the command.

Top

Parameters

Keyword	Description	Choices	Notes
MSGKEY	Message key	*Character value*	Required, Positional 1
MSGQ	Message queue	*Qualified object name*	Required, Positional 2
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RPY	Reply	*Character value*, ***DFT**	Optional, Positional 3
RMV	Remove message	***YES**, *NO	Optional
RJTDFTRPY	Reject default reply	***NOALWRJT**, *ALWRJT	Optional
CCSID	Coded character set ID	1-65535, *HEX, ***JOB**	Optional

Top

Message key (MSGKEY)

Specifies the message reference key of the message that the reply answers.

This is a required parameter.

Top

© Copyright IBM Corp. 1998, 2006

Message queue (MSGQ)

Specifies the message queue that received the inquiry message to be answered.

This is a required parameter.

Qualifier 1: Message queue

name Specify the name of the message queue.

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 message queue. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the message queue is located.

Top

Reply (RPY)

Specifies the reply that the program sends as a response to the inquiry message.

***DFT** The default reply stored in the message description of the inquiry message that was sent is sent as the reply. If no default is specified in the message description of the inquiry message, the system default reply, *N, is used.

'reply-text'
Enter the text (enclosed in apostrophes if it contains blanks or special characters) or a CL variable that contains the text that is sent as the program's reply to the inquiry message. The number of characters allowed for the reply and their format are defined by the validity specifications given in the Add Message Description (ADDMSGD) command for the specified inquiry message. However, if no validity specifications are defined for replies in the ADDMSGD command, as many as 132 characters can be used in the reply text.

Coded Character Set Identifier (CCSID) Considerations

If the inquiry message that this reply is being sent to is an immediate message, the text supplied for the RPY parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. If the inquiry message that this reply is sent to is a predefined message, the text supplied in the RPY parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Top

Remove message (RMV)

Specifies whether the inquiry message and its reply are removed from the specified message queue.

***YES** The message and its reply are removed from the message queue when the reply is sent.

***NO** The message and its reply are held in the message queue. The inquiry message cannot be replied to more than once, but it can be received or displayed multiple times.

Top

Reject default reply (RJTDFTRPY)

Specifies whether a reply handling exit program will be allowed to reject a default reply. A default reply is requested by using blanks as the value for the reply parameter. A reply handling exit program can be registered via the system registration facility for exit point QIBM_QMH_REPLY_INQ. If a default reply is not being sent, this parameter is ignored and a reply handling exit program can reject or replace the reply value.

***NOALWRJT**
A reply handling exit program will not be allowed to reject a default reply.

***ALWRJT**
A reply handling exit program will be allowed to reject a default reply. If an exit program rejects the reply, message CPD2476 (Reply rejected by a reply handling exit program) will be sent as a diagnostic message to the program using this command. The CPD2476 will be followed by a CPF2422 (Reply not valid) escape message that the program using this command should monitor for to handle and recover from error situations.

Top

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that the supplied reply text is in. If the inquiry message that this reply is sent to is a predefined message, this parameter is ignored and the text supplied for the RPY parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

***JOB** The reply text is assumed to be in the CCSID of the job running this command.

***HEX** The reply text is not converted. CCSID 65535 is used.

coded-character-set-identifier
Specify a valid CCSID in which you want your reply text to be considered in. Valid values range from 1 through 65535. This command will validate the CCSID. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of valid CCSID values.

Top

Examples

```
SNDRPY   MSGKEY(&KEY)  MSGQ(SMITH)  RPY(YES)
```

This command sends a reply of YES to the message whose reference key is specified by &KEY, which was received at message queue SMITH. Because the reply contains only one word, the reply does not have to be enclosed in apostrophes.

Top

Error messages

***ESCAPE Messages**

CPF2401
Not authorized to library &1.

CPF2403
Message queue &1 in &2 not found.

CPF2408
Not authorized to message queue &1.

CPF2410
Message key not found in message queue &1.

CPF2411
Not authorized to message file &1 in &2.

CPF2420
Reply already sent for inquiry or notify message.

CPF2422
Reply not valid.

CPF2432
Cannot send reply to message type other than *INQ or *NOTIFY.

CPF2433
Function not allowed for system log message queue &1.

CPF2460
Message queue &1 could not be extended.

CPF247E
CCSID &1 is not valid.

CPF2471
Length of field not valid.

CPF2477
Message queue &1 currently in use.

CPF2547
Damage to message file QCPFMSG.

CPF2548
Damage to message file &1 in &2.

CPF9830
Cannot assign library &1.

CPF9838
User profile storage limit exceeded.

***STATUS Messages**

CPF2458
Reply replaced by reply handling exit program.

Top

Send Service Request (SNDSRVRQS)

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

Parameters
Examples
Error messages

The Send Service Request (SNDSRVRQS) command establishes a communications session and sends problem information to your service support system or tests the communications link to your service provider.

Restriction: To use this command, the user must be signed on as QSRV or QSRVBAS, or have *ALLOBJ authority.

Top

Parameters

Keyword	Description	Choices	Notes
ACTION	Action	*PREPARED, *OPENED, *TEST, *PRBID	Required, Positional 1
RMTCPNAME	Remote control point	*Communications name*, ***IBMSRV**, *SELECT	Optional
RMTNETID	Remote network identifier	*Communications name*, ***NETATR**	Optional
PRBID	Problem identifier	*Character value*	Optional

Top

Action (ACTION)

Specifies the type of request.

***PREPARED**
All records in the Problem Log with a status of PREPARED are sent to the remote service support system.

***TEST** A test is performed on the communications link to the remote service support system.

***OPENED**
All records in the Problem Log with a status of OPEN are sent to the remote service support system.

***PRBID**
A specific record from the Problem Log with a status of PREPARED or OPENED will be sent to the selected service provider.

Top

Remote control point (RMTCPNAME)

Specifies the service provider to whom the service request is sent. When *PREPARED is specified for the **Action (ACTION)** parameter, only the problem log entries that have defined destinations will be processed.

© Copyright IBM Corp. 1998, 2006

***IBMSRV**
The service request is sent to IBM service support.

***SELECT**
A list of service providers is shown so the user can select the destination for the service request.

remote-control-point-name
Specify the remote control point name of the service provider to whom the request is sent.

Top

Remote network identifier (RMTNETID)

Specifies the remote name of the service provider's network.

***NETATR**
The service provider is in the local network.

remote-network-identifier
Specify the network name of the service provider to whom the request is being sent.

Top

Problem identifier (PRBID)

Specifies the specific problem record identifier. If the problem identifier is for a PREPARED problem, the problem must have been prepared for the selected service provider.

Top

Examples

Example 1: Sending Prepared Records

```
SNDSRVRQS   ACTION(*PREPARED)
```

This command establishes the communications link to the IBM service support system and sends all records in the problem log with the status prepared. The result of each problem log entry reported may be one of the following:

- PTFs sent to the system
- PTFs ordered from the code distribution center
- CE contacted automatically
- Service support center contacted automatically. The service support center representative will call you.
- Parts list

Example 2: Selecting a Service Provider

```
SNDSRVRQS   ACTION(*OPENED)  RMTCPNAME(*SELECT)
```

This command allows the user to select a service provider from a list. The service provider will receive all records in the problem log with an opened status.

Top

Error messages

***ESCAPE Messages**

CPF2702
Device description &1 not found.

CPF7C08
No support network connection.

CPF8C0A
Data received from &1 not recognized.

CPF8C0B
Test request routed to different system than specified.

CPF8C0C
Content of problem record &1 not valid.

CPF8C0E
Library QGPL not found.

CPF8C0F
Error indicated in reply to request.

CPF8C01
Cannot connect to IBM service system. One session allowed.

CPF8C06
No problems in problem log can be sent.

CPF8C07
A parameter is not valid.

CPF8C08
Cannot specify *SELECT for the control point name.

CPF8C09
&1 not defined as a service provider.

CPF8C16
Error occurred while processing request.

CPF8C17
Sign-on failed.

CPF8C18
No support network connection.

CPF8C19
Remote support application failed.

CPF8C2A
Cannot connect to IBM service system.

CPF8C2B
Error indicated in reply to request.

CPF8C2D
Problem &1 cannot be sent.

CPF8C24
Error occurred while processing request.

CPF8C27
Alternate load device not found.

CPF9824

Not authorized to command &1 in library &2.

CPF9846

Error while processing file &1 in library &2.

Top

Send TIE File (SNDTIEF)

Where allowed to run:

- Batch job (*BATCH)
- Batch program (*BPGM)
- Interactive program (*IPGM)
- Batch REXX procedure (*BREXX)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The Send Technical Information Exchange File (SNDTIEF) command allows you to send specified files to the remote support network.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	*Qualified object name*	Required, Positional 1
	Qualifier 1: File	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	*Name*, *FIRST	Required, Positional 2
TOUSRID	User ID	*Name*	Required, Positional 3
TONODE	Node	*Name*	Required, Positional 4
TOFILE	To file	*Name*, ***FILE**	Optional
TYPE	Type of contents of file	***OTHER**, *SYSTPLD	Optional
PTY	Priority	1, **2**	Optional

Top

File (FILE)

Specifies the physical file to be sent to the remote support network.

This is a required parameter.

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, QGPL is used.

library-name
Specify the name of the library where the database file is located.

© Copyright IBM Corp. 1998, 2006

Top

Member (MBR)

Specifies the name of the database file member being transmitted to the remote support network.

This is a required parameter.

***FIRST**
The first member of the database file specified by the **File** prompt (FILE parameter) is transmitted.

member-name
The specified member of the database file is transmitted.

Top

User ID (TOUSRID)

Specifies the user who will receive the database file.

This is a required parameter.

Top

Node (TONODE)

Specifies the system that will receive the database file.

This is a required parameter.

Top

To file (TOFILE)

Specifies the name the database file is to have on the receiving system.

***FILE** The database file on the receiving system will have the name specified on the **File** prompt (FILE parameter).

file-name
Specify the name the database file is to have on the receiving system.

Top

Type of contents of file (TYPE)

Specifies the contents of the database file.

***OTHER**
The contents of the database file are not specified.

***SYSTPLD**
The database file contains the system configuration (topology) description.

Top

Priority (PTY)

Specifies the priority of the database file on the receiving system.

2 The data file has normal priority.

1 The file has the highest priority.

Top

Examples

```
SNDTIEF   FILE(QSYS/MYTOPO)  TOUSRID(INFSERV)  TONODE(INFTIE)
          TOFILE(ACMETOPO)
```

This command sends a file named MYTOPO from library QSYS to TIE. It is held in a mailbox for user INFSERV on system INFTIE. When it is received by the user, it is named ACMETOPO.

Top

Error messages

None

Top

Send User Message (SNDUSRMSG)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: No

Parameters
Examples
Error messages

The Send User Message (SNDUSRMSG) command is used by a program to send a message to a message queue and optionally receive a reply to that message. The message sent using this command can be either an impromptu or a predefined message and can be sent to a display station user in an interactive job or a specific message queue. For inquiry messages, a CL variable can be specified to receive the reply value, and the program using this command will wait for a response.

This command uses a combination of parameters available on the Send Program Message (SNDPGMMSG) and Receive Message (RCVMSG) commands to allow a program to send and receive messages by using a single command. Also, the SNDUSRMSG command provides validity checking and uppercase translation for replies to inquiry messages.

Restrictions:

1. The SNDUSRMSG command allows a message of up to 512 characters of first-level message text to be sent. However, if the message is sent to an external message queue (*EXT) in an interactive job, only 76 characters are shown on the Display Program Messages display. If the message is sent to a user's, work station's, or system operator's message queue, the Display Message (DSPMSG) command allows all 512 characters to be displayed.

Top

Parameters

Keyword	Description	Choices	Notes
MSG	Message text, or	*Character value*	Optional, Positional 1
MSGID	Message identifier	*Name*	Optional
MSGF	Message file	*Qualified object name*	Optional
	Qualifier 1: Message file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGDTA	Message data field values	*Character value*	Optional
VALUES	Valid reply values	Values (up to 20 repetitions): *Character value*, ***NONE**	Optional
DFT	Default reply value	*Character value*, ***MSGDFT**	Optional
MSGTYPE	Message type	***INQ**, *INFO	Optional
TOMSGQ	To message queue	Single values: *, *EXT, *SYSOPR Other values: *Qualified object name*	Optional
	Qualifier 1: To message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TOUSR	To user profile	*Name*, *SYSOPR, *REQUESTER	Optional
MSGRPY	CL var for message reply	*Character value*	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
TRNTBL	Translate table	Single values: *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Translate table	*Name*, **QSYSTRNTBL**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CCSID	Coded character set ID	1-65535, *HEX, ***JOB**	Optional

Top

Message text, or (MSG)

Specifies the message text of an immediate message that is sent by the program. A maximum of 512 characters can be specified. The text string must be enclosed in apostrophes if special characters (including blanks) are used. If this parameter is specified, values cannot be specified for the **Message identifier (MSGID)** parameter, **Message file (MSGF)** parameter, or **Message data field values (MSGDTA)** parameter.

'text' Specify the text of the immediate message to be sent.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSG parameter is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Top

Message identifier (MSGID)

Specifies the message identifier of a predefined message sent by the program to a message queue. If a value is specified for this parameter, a value must also be specified for the **Message file (MSGF)** parameter, and a value cannot be specified for the **Message text, or (MSG)** parameter.

identifier
Specify the message identifier of the message to be sent.

Top

Message file (MSGF)

Specifies the message file that contains the predefined message to be sent. This parameter is valid only if a value is specified for the **Message identifier (MSGID)** parameter.

Qualifier 1: Message file

name Specify the name of the message file which contains the predefined message to be sent.

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 message file. If no current library entry exists in the library list, QGPL is used.

name Specify the library in which the message file is located.

Top

Message data field values (MSGDTA)

Specifies the character string or the CL variable that contains a character string used as the message data in the predefined message. A character string that is specified contains one or more substitution values that are used in place of the substitution variables that were defined in the message's text when the message was defined.

***NONE**
No message data is specified for the predefined message.

message-data
Specify the character string that gives the substitution values in the specified predefined message that is sent, or specify the name of the variable that contains the character string.

Coded Character Set Identifier (CCSID) Considerations

The text supplied for the MSGDTA parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID of the job running this command unless a coded character set identifier is supplied in the CCSID parameter. All other text supplied for the MSGDTA parameter is assumed to be 65535 and is not converted. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. For more information about the *CCHAR type field, see the Add Message Description (ADDMSGD) command.

Top

Valid reply values (VALUES)

Specifies a list of valid replies to an inquiry message sent by this command. No more than 20 replies can be specified in the list. If the reply to the inquiry message does not match one of the specified values, an error message is sent to the reply's sender and the inquiry message is sent again.

***NONE**
No replies to inquiry messages are specified. Any reply to an inquiry message is valid.

allowable-values
Specify no more than 20 values that are compared to replies received for inquiry messages sent by this command. The maximum length of each value is 32 characters. If this value is specified, the **CL var for message reply (MSGRPY)** parameter must also be specified.

Top

Default reply value (DFT)

Specifies the value used as the reply to an inquiry message (sent by this command) if the inquiry message is sent to a message queue that is in default delivery mode, or for any other reason the default reply is sent.

***MSGDFT**
The default value defined in the message description of the message ID (specified for the MSGID parameter) is used. If no message ID is specified, the default value is *N.

'default-reply-value'
Specify the reply (enclosed in apostrophes) used as the default reply. This value can only be specified if the **CL var for message reply (MSGRPY)** parameter is specified.

Top

Message type (MSGTYPE)

Specifies the type of message to be sent.

***INQ** An inquiry message is sent and the message queue receiving the message must reply to it.

***INFO**
An informational message is sent.

Top

To message queue (TOMSGQ)

Specifies the name of the message queue to which the message is to be sent. This parameter cannot be used if a value is specified for the **To user profile (TOUSR)** parameter.

Single values

***** In an interactive job, the message is to be sent to the external message queue (*EXT). In a batch job, the message is to be sent to the system operator (message queue QSYSOPR in library QSYS).

***SYSOPR**
The message is to be sent to the system operator (message queue QSYSOPR in library QSYS).

***EXT** The message is to be sent to the job's external message queue. For batch job inquiry messages, the default reply is always received.

Qualifier 1: To message queue

name Specify the name of the message queue that is to receive the message being sent.

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 message queue. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the message queue is located.

Top

To user profile (TOUSR)

Specifies that the message is to be sent to the message queue specified in the user profile for the user named on this parameter. This parameter cannot be used if a value is specified for the **To message queue (TOMSGQ)** parameter.

***SYSOPR**
The message is to be sent to the system operator user profile message queue, QSYS/QSYSOPR.

***REQUESTER**
The message is to be sent to the user profile message queue for interactive jobs or to the system operator's message queue (QSYS/QSYSOPR) for batch jobs.

name Specify the user profile name of the user to whom the message is to be sent.

Top

CL var for message reply (MSGRPY)

Specifies the CL character variable (of up to 132 characters) that contains the reply received in response to an inquiry message. This parameter is valid only if *INQ is specified for the **Message type (MSGTYPE)** parameter.

Top

Translate table (TRNTBL)

Specifies the name of the translation table that is used if the reply value is being translated.

Qualifier 1: Translate table

QSYSTRNTBL
The translation table named QSYSTRNTBL is used to translate the reply value.

The IBM-supplied translation table QSYSTRNTBL table translates, for the English language only, all lowercase characters in the range of X'81' to X'A9' to uppercase characters. All other characters are not translated.

To use a different translation table, use the Create Table (CRTTBL) command and specify that particular table for this parameter.

***NONE**
The reply is not being translated.

name Specify the name of the translation table that is used to translate the message reply.

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 translation table. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the translation table is located.

Top

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that the message text supplied for the MSG (message) or MSGDTA (message data) parameters is in. If a message identifier is specified, the text supplied by the MSGDTA (message data) parameter that corresponds to the *CCHAR type field is assumed to be in the CCSID supplied by this parameter. The data supplied that does not correspond to the *CCHAR type field is assumed to be 65535 and is not converted. For more information about the *CCHAR type field see the Add Message Description (ADDMSGD) command.

If no message identifier is specified, the text supplied by the MSG (message) parameter is assumed to be in the CCSID supplied by this parameter. For more information about the message handler and its use of CCSIDs, see the Globalization topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

***JOB** The text supplied by the MSGDTA (message data) or MSG (message) parameter is assumed to be in the CCSID of the job running this command.

***HEX** The text supplied by the MSGDTA (message data) or MSG (message) parameter is not converted. CCSID 65535 is used.

coded-character-set-identifier
Specify a valid CCSID in which you want your text specified for the MSG (message) or MSGDTA (message data) parameter to be considered in. Valid values range from 1 through 65535. This command validates the CCSID. See the Globalization information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for a list of CCSID values.

Top

Examples

Example 1: Message Requiring Specific Reply

```
SNDUSRMSG   MSG('Data verified. Update master files (Y,N)?')
            TOMSGQ(*)  VALUES(Y N)  DFT(N)  MSGRPY(&REPLY)
```

This command sends an inquiry message to the display station operator (if it is used in an interactive job) or to the system operator (if it is used in a batch job). The valid replies are Y and N, and any other reply is rejected. The reply is returned in the variable &REPLY. The default translation table, QSYSTRNTBL, is used to translate the reply to uppercase characters.

Example 2: Message Requiring Any Reply

```
SNDUSRMSG   MSG('Enter any response when ready to continue.')
            TOMSGQ(WS01)
```

This command sends an inquiry message to a specific message queue. Any reply is valid. Because the purpose of this example is simply to wait, no CL variable is provided to receive the reply.

Example 3: Sending an Information Message

```
SNDUSRMSG   MSGID(USR0150)  TOUSR(FRED)  MSGF(QGPL/USRMSGF)
            MSGDTA(&ACCTNO)  MSGTYPE(*INFO)
```

This command sends a predefined message as an information message to the message queue specified in the user profile of the specified user (FRED). The message data provided is included in the message.

Top

Error messages

***ESCAPE Messages**

CPF247E
CCSID &1 is not valid.

CPF2559
Error occurred in SNDUSRMSG command.

Top

Display Mounted FS Information (STATFS)

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

Parameters
Examples
Error messages

The Display Mounted File System Information (STATFS) command displays information about a mounted file system.

This command can also be issued using the following alternative command name:

- DSPMFSINF

For more information about integrated file system commands, see the Integrated file system information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Restrictions:

1. The user must have execute (*X) authority to each directory in the path.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Path name*	Required, Positional 1
OUTPUT	Output	*, *PRINT	Optional

Top

Object (OBJ)

Specifies the path name of an object that is within the mounted file system whose statistics are to be displayed. Any object in the mounted file system can be specified. For example, it can be a directory (*DIR) or a stream file (*STMF).

For more information on specifying path names, refer to ″Object naming rules″ in the CL concepts and reference topic in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

This is a required parameter.

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.

© Copyright IBM Corp. 1998, 2006

***PRINT**
The output is printed with the job's spooled output.

Top

Examples

The alternative command name for STATFS is DSPMFSINF. The following examples use the alternative command name, but STATFS can be replaced directly for DSPMFSINF in all of them.

Example 1: Displaying Statistics of a Mounted File System

```
DSPMFSINF   OBJ('/jsmith/file1')
```

This command displays the statistics for the mounted file system that contains */jsmith/file1*.

Example 2: Displaying QSYS.LIB File System Statistics

```
DSPMFSINF   OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
```

This command displays the statistics for the QSYS.LIB file system that contains *FILE object MYFILE in library MYLIB.

Top

Error messages

***ESCAPE Messages**

CPFA0A9
Object not found. Object is &1.

CPFA09C
Not authorized to object. Object is &1.

Top

Start Agent Services (STRAGTSRV)

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

Parameters
Examples
Error messages

The Start Agent Services (STRAGTSRV) command starts all of the agent services on this system that are defined in a preferences file. Another preferences file is also used to help define how the services are started. The preferences files are stream files named *ableplatform.preferences* and *able.preferences* that are located in the directory specified for the **Preferences file directory (PREFDIR)** parameter.

You can use the End Agent Services (ENDAGTSRV) command to end the agent services started by this command.

Restrictions:

- You must have all object (*ALLOBJ) and job control (*JOBCTL) special authorities to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
PREFDIR	Preferences file directory	*Path name*, ***DFT**	Optional
HOMEDIR	Home directory	*Path name*, ***CURRENT**	Optional
CLASSPATH	Additional classpath	*Path name*, ***NONE**	Optional
SBMJOBUSER	User profile for SBMJOB	Single values: ***CURRENT** Other values (up to 10 repetitions): *Element list*	Optional
	Element 1: Pool identifier	*Character value*	
	Element 2: User profile	*Name*	

Top

Preferences file directory (PREFDIR)

Specifies the directory that contains the preferences files that define the agent services to be started and how they are started. The preferences files must be named *ableplatform.preferences* and *able.preferences*.

***DFT** Use the preferences files located in /QIBM/ProdData/OS400/able/.

path-name
Specify the directory that contains the preferences files to be used to start the agent services.

Top

Home directory (HOMEDIR)

Specifies the directory to be used as the home directory of the agent services.

© Copyright IBM Corp. 1998, 2006

***CURRENT**
Use the home directory currently in effect for the job running this command. This is the home directory defined for the current user profile of the job running the STRAGTSRV command.

path-name
Specify the path name of the directory to be used as the home directory for the Java Virtual Machines (JVMs) which are used by the agents running in these services. The services use this directory to find needed resources.

Top

Additional classpath (CLASSPATH)

Specifies the additional classpath to be appended to the classpath setting of each Java Virtual Machine (JVM) to be started to run the agent services.

***NONE**
No additional classpath will be appended.

path-name
Specify the path name of the additional classpath to be appended to the classpath setting of each JVM.

Top

User profile for SBMJOB (SBMJOBUSER)

Specifies the user profile to be used for the Java Virtual Machine (JVM) jobs that will run the agent services. You can specify that a particular JVM run under a different profile than the current user profile of the job running the STRAGTSRV command.

You can specify an alternate user profile for up to 10 JVM jobs.

Single values

***CURRENT**
All JVM jobs will run under the same user profile as the current user profile of the job running this command.

Element 1: Pool identifier

character-value
Specify an identifier that matches a JVM that is defined in the preferences file specified for the **Preferences file directory (PREFDIR)** parameter.

Element 2: User profile

name Specify the name of the user profile to be used on the Submit Job (SBMJOB) commands for the JVMs. The JVM jobs will run under the specified user profile instead of the current user profile of the job running this command.

Top

Examples

Example 1: Starting with Shipped Default Values

```
STRAGTSRV
```

This command starts all of the agent services defined in the default ableplatform.preferences file in '/qibm/prodData/OS400/able/' in a way defined in the default able.preferences file in '/qibm/prodData/OS400/able/'. The home directory of each agent service will be set to the Home Directory of the current user. All jobs submitted to run the Java Virtual Machines will run under the current profile.

Example 2: Starting with User-Specified Values

```
STRAGTSRV  PREFDIR('/qibm/userData/OS400/able/test/')
           HOMEDIR('/qibm/userData/OS400/able/test/')
           CLASSPATH('/qibm/userData/OS400/able/test/test.jar')
           SBMJOBUSER((POOL1 TESTPROF1) (POOL3 TESTPROF3))
```

This command starts all of the agent services defined in the ableplatform.preferences file in '/qibm/userData/OS400/able/test/' in a way defined in the default able.preferences file in '/qibm/userData/OS400/able/test/'. The home directory of each agent service will be set to '/qibm/userData/OS400/able/test/'. The Java Virtual Machines will have thier classpaths set to include the test.jar Java Archive file in '/qibm/userData/OS400/able/test/'. The jobs submitted to run the Java Virtual Machines for POOL1 and POOL3 will run under the users TESTPROF1 and TESTPROF3 respectivley, all other JVMs will run under the current user profile.

Top

Error messages

***ESCAPE Messages**

CPF1890
*ALLOBJ authority required for requested operation.

CPF90FF
*JOBCTL special authority required to do requested operation.

Error messages from submitted job:

This command submits a batch job which, in turn, submits additional batch jobs where the agent services will run. The following error messages could be signaled from this batch job:

CPF4B01
Agent Service not submitted. Reason code &1.

CPF4B02
The Java Virtual Machine(JVM) for the agent service was not started. Reason code &1

CPF4B03
Java Virtual Machine(JVM) exception has occurred.

Top

Start ASP Balance (STRASPBAL)

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

Parameters
Examples
Error messages

The Start ASP Balance (STRASPBAL) command allows the user to start the auxiliary storage pool (ASP) balancing function for one or more ASPs. Four types of ASP balancing can be started:

1. **Capacity balancing** - The data on all the units within the ASP will be balanced so each unit has an equal percentage of used and unused space. This is useful when new units are added to an ASP. Instead of having several units with the majority of the data and several new ones with no data, the user can spread the data evenly across all the units.
2. **Usage balancing** - The 'low' use data on each 'low' utilized unit in the ASP is redistributed to balance future use of the arm utilization of each unit within the specified ASP. *USAGE balancing cannot be done until the trace controlled by the Trace ASP Balance (TRCASPBAL) command has collected usage statistics. The TRCASPBAL command starts the trace function that will identify the 'high' and 'low' use data on each unit. After the usage balance activity has run to completion, the trace information will be cleared. Usage balancing is useful when the ASP contains large capacity disk units.
3. **Hierarchical Storage Management (HSM) balancing** - The 'high' use and 'low' use data on each unit in the ASP is redistributed so that the 'high' use data resides on high performance units and the 'low' use data resides on low performance units. The ASP being balanced in this manner must contain the correct mixture of slow and fast units to perform this type of balance. This allows the 'low' use data on the high performance units to be moved to low performance units. Data that is 'high' use that resides on slow units will be moved to the fast disk units. HSM balancing cannot be done until the trace controlled by the Trace ASP Balance (TRCASPBAL) command has collected usage statistics. The TRCASPBAL command starts the trace function that will identify the 'high' and 'low' use data on each unit. After the HSM balance activity has run to completion, the trace information will be cleared. HSM balancing is useful when the ASP contains compressed disk units.
4. **Move data from units** - This option can be used to reduce the down time associated with removing a disk unit. A unit that is scheduled for removal can be marked to end allocations by specifying UNIT(unit-number) and TYPE(*ENDALC). This will keep new allocations away from this unit. For all units marked *ENDALC, specifying TYPE(*MOVDTA) will move data from the marked units to other units in the same ASP. To resume allocations for units marked *ENDALC, specify UNIT(unit-number) and TYPE(*RSMALC). New allocations will once again be allowed to this unit. The Check ASP Balance (CHKASPBAL) command can be used to determine which units are currently marked *ENDALC.

The user may specify a time limit that the function is to run for each ASP being balanced or the balance can be set to run to completion. If the balance function needs to be ended, use the End ASP Balance (ENDASPBAL) command. A message will be sent to the system history (QHST) log when the balancing function is started for each ASP. A message will also be sent to the QHST log when the balancing function completes or is ended.

If the balance function is run for a few hours and then stopped, it will continue from where it left off when the balance function restarts. This allows the balancing to be run during off hours over a several day period.

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.

© Copyright IBM Corp. 1998, 2006

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Balance type	*CAPACITY, *USAGE, *HSM, *MOVDTA, *ENDALC, *RSMALC	Optional, Positional 2
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): *Name*	Optional
UNIT	Storage unit	Values (up to 300 repetitions): *Integer*	Optional
TIMLMT	Time limit	1-9999, *NOMAX	Optional, Positional 3

Top

Balance type (TYPE)

Specifies the type of auxiliary storage pool (ASP) balancing operation to be done.

Note: A value must always be specified for this parameter.

***CAPACITY**
The capacity balance function will be started for the specified ASP (ASP or ASPDEV parameter).

***USAGE**
The usage balance function will be started for the specified ASP (ASP or ASPDEV parameter) .

***HSM** The hierarchical storage management balance function will be started for the specified ASP (ASP or ASPDEV parameter).

***MOVDTA**
Data will be moved off all units marked *ENDALC to other units in the same ASP. The Check ASP Balance (CHKASPBAL) command can be used to determine which units are marked *ENDALC.

***ENDALC**
New allocations will no longer go to the specific units (UNIT parameter). However, the system will use these units for new allocations to avoid 'Out of storage' conditions.

***RSMALC**
Resume allocations to the specified units (UNIT parameter). If the unit has a much lower percentage used than other units in the ASP, run the STRASPBAL command again specifying TYPE(*CAPACITY) for the ASP that contains the unit. This will keep new allocations balanced across the units in the ASP, instead of going mostly to this unit.

Top

Auxiliary storage pool ID (ASP)

Specifies the auxiliary storage pool (ASP) for which the ASP balancing function is to be started.

Note: A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter if *CAPACITY, *USAGE, or *HSM is specified for the **Balance type (TYPE)** parameter.

Single values

***ALL** ASP balancing will be started 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 balancing is to be started.

Top

ASP device (ASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device for which the ASP balancing is to be started.

Note: A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter if *CAPACITY, *USAGE, or *HSM is specified for the **Balance type (TYPE)** parameter.

Single values

***ALLAVL**
ASP balancing will be started 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

Storage unit (UNIT)

Specifies the unit number for which new allocations are to be ended (if *ENDALC is specified for the **Balance type (TYPE)** parameter) or for which new allocations are to resume (if *RSMALC is specified for the TYPE parameter). Specifying TYPE(*MOVDTA) will start moving data off the units marked *ENDALC.

Note: A value must be specified for this parameter if *ENDALC or *RSMALC is specified for the TYPE parameter.

integer
Specify the unit number for which new allocations are to either end (if *ENDALC is specified) or resume (if *RSMALC is specified). Up to 300 unit numbers may be specified.

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. For example, if ASP(*ALL) is specified and the machine has four ASPs configured and TIMLMT(60) is specified, four balance functions are started and each can run 60 minutes. If the balancing of any ASP has not completed after 60 minutes, it will be forced to end. This allows you to do ASP balancing incrementally. The balance function will not run across an IPL.

Note: A value must be specified for this parameter if *CAPACITY, *USAGE, *HSM, or *MOVDTA is specified for the **Balance type (TYPE)** parameter.

***NOMAX**
There is no time limit for the balance function. For a large ASP the balance function can take a long time to complete. If you start the balance function with *NOMAX and you want to force the function to end, you can use the End ASP Balance (ENDASPBAL) command.

1-9999 Specify the number of minutes that the balance function will be allowed to run.

Top

Examples

Example 1: Starting ASP Balancing for ASP 1

```
STRASPBAL   ASP(1) TIMLMT(*NOMAX) TYPE(*CAPACITY)
```

This command starts the *CAPACITY ASP balance function for ASP 1. The balance function will run until the capacity for each of the units has been balanced.

Example 2: Starting ASP Balancing with a Time Limit

```
STRASPBAL   ASP(*ALL)  TIMLMT(60)  TYPE(*USAGE)
```

This command starts the *USAGE ASP balance function for the system ASP and each configured basic ASP for which the TRCASPBAL command has been run. Each balance function will have a time limit of sixty minutes. After sixty minutes, any balance functions which have not completed will be ended.

Example 3: Starting ASP Balancing for an ASP Device

```
STRASPBAL   ASPDEV(MYASP1)  TIMLMT(*NOMAX)  TYPE(*CAPACITY)
```

This command starts the *CAPACITY ASP balance function for the ASP device named MYASP1. The balance function will run until complete.

Example 4: Preparing to Remove Units

```
STRASPBAL   UNIT(11 12 13)  TYPE(*ENDALC)
STRASPBAL   TYPE(*MOVDTA)
```

The first command marks units 11, 12 and 13 to no longer receive new allocations. The second command begins to move data off the marked units. It is recommended that the *MOVDTA ASP balancing function be done during an off-peak time.

Top

Error messages

***ESCAPE Messages**

CPF18AB
ASP balancing for ASP &1 already started.

CPF18AE
ASP &1 does not contain trace data.

CPF18B1
Trace function currently running for ASP &1.

CPF18AD
ASP &1 must contain more than a single unit.

CPF18AF
ASP &1 does not contain mixed unit types.

CPF18B3
Balance type not valid for ASP &1.

CPF1890
*ALLOBJ authority required for requested operation.

CPF9829
Auxiliary storage pool &1 not found.

Top

Start CHT Server (STRCHTSVR)

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

Parameters
Examples
Error messages

The Start Clustered Hash Table Server (STRCHTSVR) command is used to define a clustered hash table server on each cluster node specified in the NODE parameter. Successful completion of this command results in a job being started on each node defined in the cluster. Each node specified on the NODE parameter will be in the clustered hash table servers domain.

The clustered hash table server enables sharing and replicating data between cluster nodes using the Clustered Hash Table APIs. The data is stored within the clustered hash table server in non-persistent storage.

You can use the End Clustered Hash Table Server (ENDCHTSVR) command to end the clustered hash table server.

Restrictions:

- Cluster Resource Service must be active on the local node.
- All nodes specified in the NODE parameter must have Cluster Resource Services active.
- Requesting user profile must exist on all nodes specified in the NODE parameter.
- Requesting user must have change (*CHANGE) authority to the authorization list, if specified, on the current node as well as all nodes specified for the Node (NODE) parameter to perform the start.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server	*Communications name*	Required, Positional 1
RSPTIMO	Request response timeout	1-300, **60**	Optional
AUTL	Authorization list	*Name*, ***SERVER**	Optional
NODE	Node	Values (up to 20 repetitions): *Communications name*, ***LOCAL**	Optional

Top

Server (SERVER)

Specifies the clustered hash table server to be started.

This is a required parameter.

name Specify the name of the clustered hash table server to be started.

Top

© Copyright IBM Corp. 1998, 2006

Request response timeout (RSPTIMO)

Specifies the amount of time (in seconds) that the clustered hash table server has to complete a retrieve request. An error will be returned to the requestor of the retrieve if the clustered hash table server does not get a response before the specified length of time. This parameter is ignored when the current cluster version is 3 or greater.

60 The clustered hash table server starting on the nodes specified on the NODE parameter will wait 60 seconds for a response.

1-300 Specify the number of seconds the clustered hash table server waits for a response before returning an error.

Top

Authorization list (AUTL)

Specifies an authorization list for the clustered hash table server. This defines the list of users authorized to start, end and connect to the clustered hash table server. An authorization list by the same name must exist on all nodes in the NODE parameter. The same authorization list name must be specified when starting a clustered hash table server that is already active on other nodes in the cluster.

***SERVER**
Use the same authorization list as the clustered hash table server being started if it is already active on one of the nodes in the cluster. If the server does not exist in the cluster and this value is specified then no special authority will be needed to start, end or connect to the server. This value must be specified if the current cluster version is 2 or less.

name Specify the name of the authorization list which defines the list of users authorized to the clustered hash table server. Users must have change (*CHANGE) authority to the named authorization list to start and end the clustered hash table server. Users must have use (*USE) authority to the named authorization list to connect to the clustered hash table server. Management of the authorization list is the users responsibility.

Top

Node (NODE)

Specifies the list of cluster nodes that compose the clustered hash table domain. A server job will be started on each of the cluster nodes specified. If the clustered hash table server already exists in the cluster, the cluster nodes specified will be added to the clustered hash table domain and a job will be started on the node specified. Nodes in this list must be unique.

The nodes must be active in the cluster.

***LOCAL**
A clustered hash table server job will be started on the local node only. *LOCAL can be specified only once.

name Specify the name of each cluster node that defines the clustered hash table domain. Up to 20 cluster nodes can be specified.

Top

Examples

Example 1: Starting a Local Clustered Hash Table Server

```
STRCHTSVR   SERVER(CHTSVR1)
```

This command starts clustered hash table server CHTSVR1 only on the local node. There is no authority restriction on the clustered hash table server.

Example 2: Starting a Clustered Hash Table Server on Multiple Nodes

```
STRCHTSVR   SERVER(CHTSVR2)  AUTL(AUTHLIST)  NODE(FRED BARNEY)
```

This command starts clustered hash table server CHTSVR2 on nodes FRED and BARNEY. The clustered hash table server has access restricted by authorization list AUTHLIST.

Top

Error messages

***ESCAPE Messages**

CPFBD02
Start clustered hash table server failed.

Top

Start Cleanup (STRCLNUP)

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

Parameters
Examples
Error messages

The Start Cleanup (STRCLNUP) command starts the cleanup operation, if allowed. Cleanup is allowed if *YES is specified for the **Allow cleanup (ALWCLNUP)** parameter of the Change Cleanup (CHGCLNUP) command.

A batch job is submitted to the job queue specified on the Change Cleanup (CHGCLNUP) command if cleanup is allowed. This cleanup control job submits individual batch jobs to the same job queue each day These batch jobs do the actual cleanup of the items specified on the CHGCLNUP command.

More information is in the Basic System Operation information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Restriction: You must have job control (*JOBCTL) special authority and have at least *USE authority to the QPGMR user profile to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	***SCHED**, *IMMED	Optional, Positional 1

Top

Option (OPTION)

Specifies when the cleanup operation is started.

***SCHED**
The cleanup operation is started as scheduled for the **Time cleanup starts each day (STRTIME)** parameter of the Change Cleanup (CHGCLNUP) command or as scheduled on the Change Cleanup Options display.

***IMMED**
The cleanup operation starts immediately.

Top

© Copyright IBM Corp. 1998, 2006

Examples

Example 1: Starting Cleanup Operation as Scheduled

```
STRCLNUP
```

This command starts the cleanup operation as specified.

Example 2: Starting Cleanup Operation as Specified on CHGCLNUP Command

```
STRCLNUP   OPTION(*SCHED)
```

This command starts the cleanup operation as specified on the STRTIME parameter of the Change Cleanup (CHGCLNUP) command or on the Change Cleanup Options panel.

Example 3: Starting Cleanup Operation Immediately

```
STRCLNUP   OPTION(*IMMED)
```

This command starts the cleanup operation immediately.

Top

Error messages

***ESCAPE Messages**

CPF1E2A
Unexpected error in QSYSSCD job.

CPF1E2B
Power scheduler and cleanup options not found.

CPF1E3A
Not authorized to start cleanup.

CPF1E3C
Job queue &2/&1 not found.

CPF1E3D
Library &1 for JOBQ parameter not found.

CPF1E33
Cleanup options or power schedule in use by another user.

CPF1E34
Error occurred starting &1 job.

CPF1E36
Cleanup has not been started.

CPF1E37
Cleanup has already been started or is scheduled to run.

CPF1E38
Cleanup not allowed to run.

CPF1E39
Not authorized to job queue used for cleanup.

CPF1E99
Unexpected error occurred.

Top

Start Cluster Node (STRCLUNOD)

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

Parameters
Examples
Error messages

The Start Cluster Node (STRCLUNOD) command is used to start Cluster Resource Services on a node in the cluster. If Cluster Resource Services is successfully started on the node specified, the status of the node will be set to Active.

As part of starting the node, Cluster Resource Services checks the allow add to cluster (ALWADDCLU) network attribute to see whether the node being started should be part of the cluster and whether to validate the cluster request through the use of X.509 digital certificates. If validation is required, the requesting node and the node being added must have the following installed on the systems:

1. Operating System option 34 (Digital Certificate Manager)
2. Cryptographic Access Provider Product (AC2 or AC3)

If the cluster is partitioned, this command may be used to start nodes in the partition running this command.

The potential node version of the node being started must be equal to the current cluster version or up to one level higher than the current cluster version. The potential node version and the current cluster version can be retrieved by using the Display Cluster Information (DSPCLUINF) command.

If the node being started is in a device domain, then Operating System option 41, HA Switchable Resources, must be installed and a valid license key must exist on that node.

Restrictions:

1. You must have input/output system configuration (*IOSYSCFG) special authority to run this command.
2. This command cannot be called from a cluster resource group exit program.
3. The node being started must exist in the cluster membership list.
4. If all nodes have a status of New, this command must be invoked on the node on which the cluster was originally created.
5. The node to be started must be IP reachable (TCP/IP is active and the INETD server is started).
6. The first time a node is started, this command must be invoked on a node that is active.

Top

Parameters

Keyword	Description	Choices	Notes
CLUSTER	Cluster	*Name*	Required, Positional 1
NODE	Node identifier	*Name*	Required, Positional 2

Top

© Copyright IBM Corp. 1998, 2006

Cluster (CLUSTER)

Specifies the cluster that contains the node to be started.

This is a required parameter.

name Specify the name of the cluster.

Top

Node identifier (NODE)

Specifies the node identifier to be started.

This is a required parameter.

name Specify the node to be started.

Top

Examples

```
STRCLUNOD   CLUSTER(MYCLUSTER)  NODE(NODE01)
```

This command starts Cluster Resource Services on node NODE01 for cluster MYCLUSTER.

Top

Error messages

***ESCAPE Messages**

CPF1999
Errors occurred on command.

Top

Start Communications Server (STRCMNSVR)

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

Parameters
Examples
Error messages

The Start Communications Server (STRCMNSVR) command is used to start the target display station pass-through server. The target display station pass-through server processes display station pass-through, iSeries Access work station function (WSF), and other 5250 emulation programs on programmable workstations.

Restrictions You must have job control (*JOBCTL) special authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
NBRPASTHR	Number of servers	1-100, ***SYSVAL**	Optional

Top

Number of servers (NBRPASTHR)

Specifies the number of target display station pass-through server jobs to be started.

The possible values are:

***SYSVAL**
Specifies that the system value (QPASTHRSVR) is to be used to determine the number of servers.

number-pass-through-servers
Specify the number of servers. This will override the current system value (QPASTHRSVR). Valid values range from 1 to 100.

Top

Examples

Example 1: Starting Target Display Station Pass-through Server

```
STRCMNSVR
```

This command starts the target display station pass-through server. System value QPASTHRSVR is used to determine how many server jobs are started.

Top

Error messages

None

© Copyright IBM Corp. 1998, 2006

Top

Start Communications Trace (STRCMNTRC)

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

Parameters
Examples
Error messages

The Start Communications Trace (STRCMNTRC) command initiates a communications trace for a specified line, a network interface or a network server description.

A communications trace continues until:

- The End Communications Trace (ENDCMNTRC) command is run.
- The Communications Trace function of the Start System Service Tools (STRSST) command is used to end the trace.
- A physical line problem causes the trace to end.
- TRCFULL(*STOPTRC) is specified and the buffer becomes full.
- Automatically by the watch for trace event functionality.

Restrictions:

- The user must have *USE authority to the line, network interface, or network server to be traced.
- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service Trace function of Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- The following user profiles have authority to this command:
  - QSECOFR
  - 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 Operating System through iSeries 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

© Copyright IBM Corp. 1998, 2006

Parameters

Keyword	Description	Choices	Notes
CFGOBJ	Configuration object	*Name*	Required, Positional 1
CFGTYPE	Type	*LIN, *NWI, *NWS	Required, Positional 2
MAXSTG	Buffer size	*Integer*, *MIN, *MAX, **128K**, 256K, 2M, 4M, 6M, 8M, 16M, 32M, 64M, 128M, 256M, 512M, 1G	Optional
DTADIR	Data direction	*SND, *RCV, ***BOTH**	Optional
TRCFULL	Trace full	***WRAP**, *STOPTRC	Optional
USRDTA	Number of user bytes to trace	Single values: ***CALC**, *MAX Other values: *Element list*	Optional
	Element 1: Beginning bytes	*Decimal number*	
	Element 2: Ending bytes	*Decimal number*, *CALC	
CMNTRCOPTS	Communications trace options	***ALLDTA**, *RMTCTL, *RMTMAC, *RMTSAP, *LCLSAP, *IPPCLNUM, *RMTIPADR	Optional
DDITRCOPTS	DDI trace options	*ALLDTA, *RMTCTL, *RMTMAC, *RMTSAP, *LCLSAP, *IPPCLNUM, *RMTIPADR	Optional
RMTCTL	Remote controller	*Name*	Optional
RMTMAC	Remote MAC address	*Hexadecimal value*	Optional
RMTSAP	Remote SAP	*Hexadecimal value*	Optional
LCLSAP	Local SAP	*Hexadecimal value*	Optional
IPPCLNUM	IP protocol number	0-255, *ICMP, *IGMP, *TCP, *EGP, *IGP, *UDP	Optional
RMTIPADR	Remote IP address	*Character value*	Optional
LMITRCOPTS	LMI trace options	***ALLDTA**, *NOLMI, *LMIONLY	Optional
NWSTRCOPTS	NWS trace options	***NETBIOS**, *INTERNAL, *TCPIP	Optional
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**	
WCHTIMO	Length of time to watch	1-43200, ***NOMAX**	Optional

Keyword	Description	Choices	Notes
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
TEXT	Trace description	*Character value*, ***BLANK**	Optional

Top

Configuration object (CFGOBJ)

Specifies the configuration object to be traced. The object is either a line description, a network interface description, or a network server description.

name Specify the name of the configuration object to be traced.

Top

Type (CFGTYPE)

Specifies the type of configuration description to trace.

***LIN** The configuration object is a line description.

***NWI** The configuration object is a network interface description.

***NWS** The configuration object is a network server description.

Top

Buffer size (MAXSTG)

Specifies the trace buffer size.

128K A trace buffer of 128 kilobytes is used.

***MIN** The minimum trace buffer size is used.

***MAX** The maximum trace buffer size is used.

buffer-size
Specify the trace buffer size. Valid buffer sizes may be specified either as number of kilobytes, or as one of the following special values which has a one-letter suffix of 'K' for kilobytes, 'M' for megabytes, or 'G' for gigabytes: 128K, 256K, 2M, 4M, 6M, 8M, 16M, 32M, 64M, 128M, 256M, 512M, 1G. The minimum trace buffer size is 128 kilobytes.

Top

Data direction (DTADIR)

Specifies the communication data to trace.

Note: For network server description traces, this parameter is ignored and *BOTH is used.

***BOTH**
Data sent and received by the system is traced.

***SND** Data sent by the system is traced.

***RCV** Data received by the system is traced.

Top

Trace full (TRCFULL)

Specifies the action the system takes when the trace buffer is full of data.

***WRAP**
The trace continues and overwrites the data in the buffer.

***STOPTRC**
The trace stops.

Top

Number of user bytes to trace (USRDTA)

Specifies the amount of beginning and ending user data to trace.

Note: For network server description traces and binary synchronous lines, this parameter is ignored and *CALC is used.

Single values

***CALC**
The system determines the number of beginning and ending bytes to be traced. For LAN lines, this is the first 100 bytes. For other line types, the whole frame is traced.

***MAX** Trace as much of frames as possible. For non-LAN, *MAX will be the equivalent of *CALC.

Element 1: Beginning bytes

decimal-number
Specify the number of bytes of beginning user data to be traced.

Element 2: Ending bytes

***CALC**
The system determines the number of ending bytes to be traced.

decimal-number
Specify the number of bytes of ending user data to be traced.

Top

Communications trace options (CMNTRCOPTS)

Specifies the type of data to be traced.

***ALLDTA**
All data is traced. No filtering is specified.

***RMTCTL**
The data traveling to and from a remote controller is traced.

***RMTMAC**
The data traveling to and from a remote medium access control (MAC) address is traced.

***RMTSAP**
The data traveling to and from a remote service access point (SAP) is traced.

***LCLSAP**
The data traveling to and from a local service access point (SAP) is traced.

***IPPCLNUM**
The data within an Internet Protocol (IP) number is traced.

***RMTIPADR**
The data traveling to and from a remote IP address is traced.

Top

DDI trace options (DDITRCOPTS)

The DDITRCOPTS parameter is supported for upward compatibility of CL programs which contain the STRCMNTRC command. The CMNTRCOPTS parameter provides all of the same function as DDITRCOPTS and should be used instead of DDITRCOPTS.

***ALLDTA**
All data is traced. No filtering is specified.

***RMTCTL**
The data traveling to and from a remote controller is traced.

***RMTMAC**
The data traveling to and from a remote medium access control (MAC) address is traced.

***RMTSAP**
The data traveling to and from a remote service access point (SAP) is traced.

***LCLSAP**
The data traveling to and from a local service access point (SAP) is traced.

***IPPCLNUM**
The data within an Internet Protocol (IP) number is traced.

***RMTIPADR**
The data traveling to and from a remote IP address is traced.

Top

Remote controller (RMTCTL)

Specifies the remote controller receiving and sending the data to be traced.

name Specify the name of the remote controller.

Top

Remote MAC address (RMTMAC)

Specifies the remote medium access control address receiving and sending the data to be traced.

hexadecimal-value
Specify the remote medium access control address.

Top

Remote SAP (RMTSAP)

Specifies the remote service access point receiving and sending the data to be traced.

hexadecimal-value
Specify the remote service access point.

Top

Local SAP (LCLSAP)

Specifies the local service access point receiving and sending the data to be traced.

hexadecimal-value
Specify the local service access point.

Top

IP protocol number (IPPCLNUM)

Specifies the Internet Protocol (IP) number to be traced.

***ICMP6**
The Internet control message group is traced.

***IGMP**
The Internet group management group is traced.

***TCP** The transmission control group is traced.

***EGP** The exterior gateway protocol group is traced.

***IGP** A private interior gateway group is traced.

***UDP** The user datagram group is traced.

0-255 Specify the Internet Protocol (IP) number to trace.

Top

Remote IP address (RMTIPADR)

Specifies the remote Internet Protocol (IP) address to be traced.

character-value
Specify the remote IP address to be traced.

Top

LMI trace options (LMITRCOPTS)

Specifies the type of data to be placed in the trace buffer.

***ALLDTA**
All data, including the local management interface (LMI), is placed in the trace buffer.

***NOLMI**
All data, except LMI data, is placed in the trace buffer.

***LMIONLY**
Only LMI data is placed in the trace buffer.

Top

NWS trace options (NWSTRCOPTS)

Specifies the type of data to be placed in the trace buffer.

***NETBIOS**
All NetBIOS data is placed in the trace buffer.

***INTERNAL**
The communications processor operating system data is placed in the trace buffer.

***TCPIP**
All TCP/IP data for network server description applications is placed in the trace buffer.

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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server 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.

Top

Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

***NOMAX**
There is no time limit for watching for a particular message or LIC log entry.

1-43200
Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

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

```
1   Trace option setting   Input    Char(10)
2   Reserved               Input    Char(10)
3   Error detected         Output   Char(10)
4   Comparison data        Input    Char(*)
```

Allowed values for the ″Trace option setting″ parameter are:

***ON** The watch for trace facility is starting when the collection of trace information is started.

***MSGID**
A match on a message id specified on WCHMSG parameter occurred.

***LICLOG**
A match on a LIC log specified on the WCHLICLOG parameter occurred.

***CMPDATA**
The major and minor code of a LIC log matched, but the comparison data did not.

***INTVAL**
The time interval specified on TRCPGMITV parameter is elapsed.

***WCHTIMO**
The length of time to watch specified on WCHTIMO parameter is elapsed.

The ″Reserved″ parameter must be set to blanks.

Allowed values for the ″Error detected″ parameter are:

***CONTINUE**
The trace and the watch for trace event facility will continue running.

***STOP**
The trace and the watch for trace event facility will be ended.

***ERROR**
Error detected by customer trace program.

Allowed values for the ″Comparison data″ parameter when *MSGID is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(7)      Message ID
11   B     CHAR(9)      Reserved
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      Message comparison data
```

Allowed values for the ″Comparison data″ parameter when *LICLOG or *CMPDATA is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(4)      LIC Log major code
 8   8     CHAR(4)      LIC Log minor code
12   C     CHAR(8)      LIC Log identifier
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      LIC log comparison data
```

Allowed values for the ″Comparison data″ parameter when *ON, *INTVAL or *WCHTIMO is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information (always 4).
```

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.iseries.ibm.com/infocenter .

Single values

***NONE**
No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

Qualifier 1: Trace program

name Specify the name of the trace exit program.

Qualifier 2: Library

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

name Specify the name of the library where the user exit program is located.

Top

Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

***NONE**
No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

1-9999 Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

Trace description (TEXT)

Specifies the text that briefly describes the object.

***BLANK**
Text is not specified.

character-value
Specify up to 20 characters of text.

Top

Examples

Example 1: Start a Communications Trace for a Line Description

```
STRCMNTRC   CFGOBJ(*QESLINE)  CFGTYPE(*LIN)
```

This command starts a communications trace of line description QESLINE.

Example 2: Start a Trace and Watch for a Message to End the Trace

```
STRCMNTRC   CFGOBJ(LINE001)  CFGTYPE(*LIN)  WCHMSG((MCH2804))
            WCHMSGQ((*SYSOPR) (*JOBLOG))
            WCHJOB((*ALL/MYUSER/MYJOBNAME))
            TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts a communications trace of line description LINE001. 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 3: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
STRCMNTRC   CFGOBJ(LINE001)  CFGTYPE(*LIN)
            WCHLICLOG(('99??' 9932 MYJOBNAME))
            WCHTIMO(*NOMAX)
```

This command starts a communications trace of line description LINE001. 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 ENDCMNTRC command is issued manually.

Top

Error messages

***ESCAPE Messages**

CPF2601
Line description &1 not found.

CPF2634
Not authorized to object &1.

CPF39AA
Trace &1 type &2 already exists

CPF39AB
Beginning or ending bytes exceeds maximum value

CPF39AC
Total of beginning and ending bytes exceeds maximum value

CPF39AD
&1 type &2 cannot be traced

CPF39A6
Storage could not be allocated

CPF39A7
Trace storage not available in communications processor

CPF39A8
Not authorized to communications trace service tool

CPF39A9
Error occurred during communications trace function

CPF39BD
Network interface description &1 not found

CPF39BF
Remote IP address not valid.

CPF39B6
Communications trace function cannot be performed

CPF39C0
Controller description &1 not found.

CPF39C1
Controller description &1 not valid.

CPF39C2
Number of user bytes to trace must be *CALC.

CPF39F1
Trace buffer size too large.

CPF39F2
Cannot allocate library &1

CPF98A2
Not authorized to &1 command.

Top

Start Commitment Control (STRCMTCTL)

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

Parameters
Examples
Error messages

The Start Commitment Control (STRCMTCTL) command is used to establish either a job level or activation group level commitment definition. The job's current name space determines which IASP the commitment definition is created in, and only files in that same ASP can be opened under commitment control for this commitment definition.

This command also specifies the level of record locking that occurs for the commitment definition to be started. Also, a notify object can be specified.

Before a commitment definition is established, the user must ensure that all database files that are to be opened under commitment control for a single commitment transaction are journaled. If only the after images are being journaled, the system implicitly begins journaling both the before and the after images for the duration of the changes being made to files opened under this commitment definition.

A default journal can be specified. Entries that describe all journals and systems involved in a commitment control operation can be placed in this journal.

More information on the use of journal management is in the ″Journal management″ article in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter. More information on commitment control is in the ″Commitment control″ article in the Information Center.

Restrictions:

1. The user must have object operational and add authority to the object named on the NFYOBJ parameter, if an object is specified.
2. The user must have object operational and add authority to the object named on the DFTJRN parameter, if an object is specified.

Top

Parameters

Keyword	Description	Choices	Notes
LCKLVL	Lock level	*CHG, *CS, *ALL	Required, Positional 1
NFYOBJ	Notify object	Single values: ***NONE** Other values: *Element list*	Optional, Positional 2
	Element 1: Object	*Qualified object name*	
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Object type	Single values: *MSGQ, *DTAARA Other values: *Element list*	
	Element 1: (*MSGQ *DTAARA or *FILE)	*FILE	
	Element 2: Member, if *FILE	*Name*, ***FIRST**	
CMTSCOPE	Commitment definition scope	***ACTGRP**, *JOB	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
TEXT	Text 'description'	*Character value*, ***DFTTEXT**	Optional
DFTJRN	Journal	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Journal	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OMTJRNE	Journal entries to be omitted	***NONE**, *LUWID	Optional

Top

Lock level (LCKLVL)

Specifies the default level of record locking that occurs for the commitment definition to be started.

This is a required parameter.

***CHG** Every record read for update (for a file opened under commitment control) is locked. If a record is changed, added, or deleted, that record remains locked until the transaction is committed or rolled back. Records that are accessed for update operations but are released without being changed are unlocked.

***CS** Every record accessed for files opened under commitment control is locked. A record that is read, but not changed or deleted, is unlocked when a different record is read. Records that are changed, added, or deleted are locked until the transaction is committed or rolled back.

***ALL** Every record accessed for files opened under commitment control is locked until the transaction is committed or rolled back.

Top

Notify object (NFYOBJ)

Specifies the name and type of the object where notification is sent regarding the status of a transaction for a commitment definition. The commitment identifier of the last successful commit operation is sent to the notify object only for the following conditions:

- For a job level commitment definition, if any of the following are true:
  - A system failure occurs
  - The job ends with uncommitted changes
  - The job ends with a nonzero completion code
- For an activation group level commitment definition, if any of the following are true:
  - A system failure occurs
  - The job ends with uncommitted changes
  - The job ends with a nonzero completion code
  - The activation group ends abnormally
  - The activation group ends with uncommitted changes and the uncommitted changes are rolled back

For a system failure, the commitment identifier is placed in the notify object after the next successful initial program load (IPL). For a job that ends with uncommitted changes or with a nonzero completion code, the commitment identifier is placed in the notify object during end job processing. For an activation group that ends with uncommitted changes or ends abnormally, the notification text is placed in the notify object during activation group end processing.

A commitment identifier (specified for the **Commit identification (CMTID)** parameter on the Commit (COMMIT) command) can be specified on each commit operation performed for a commitment definition. If more than one job is concurrently using commitment control or there is more than one commitment definition being used concurrently within a single job, then each commitment definition for each job should use a unique notify object or the specified commit identifier should contain unique text such that the text identifies a single commitment definition for a single job. If *NONE is specified for the CMTID parameter of the Commit (COMMIT) command, this entry is ignored.

***NONE**
No notification is sent after an abnormal system or process end.

object-name
Specify the name (library-name/object-name) of the object to receive notification of the last transaction that is successfully committed. You must have correct authority for the object specified.

The possible library values are:

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

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

library-name
Specify the library where the object is located.

The possible object type values are:

***MSGQ**
The text identifying the last commitment boundary is placed on the specified message queue.

***DTAARA**
The text identifying the last commitment boundary is placed in the specified data area. The data area specified must be of type character, and unique to this job. The text is padded or truncated to fit the data area.

***FILE** The text identifying the last commitment boundary is added to the specified physical file.

The possible physical file member values are:

***FIRST**
The first member of the physical file receives the notification.

member-name
Specify the name of the member of the physical file that receives the notification.

Top

Commitment definition scope (CMTSCOPE)

Specifies the scope for the commitment definition to be started.

***ACTGRP**
An activation-group-level commitment definition is started for the activation group associated with the program issuing the command.

***JOB** The job-level commitment definition is started for the job.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the commitment definition to be started. More information on this parameter is in the CL Reference book, Appendix A.

***DFTTEXT**
The system is to provide a default text description for the commitment definition.

'description'
Specify no more than 50 characters of text, enclosed in apostrophes.

Top

Journal (DFTJRN)

Specifies the default journal. The default journal contains entries identifying each of the resources involved in a unit of work. Entries can also be placed when each unit of work starts or ends due to a commit or rollback operation, depending on the OMTJRNE parameter value.

See the Backup and Recovery book, SC41-5304 for information on how the system performs the rollback operation under commitment control.

The default journal can be used when adding a resource through the Add Committable Resource (QTNADDCR) Application Program Interface (API). If the special value *DFTJRN is specified for the journal name when calling the API, the name specified on this DFTJRN parameter is used.

***NONE**
No default journal is specified.

The name of the default journal can be qualified by one of the following values:

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

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

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

journal-name
Specify the name of the default journal.

Top

Journal entries to be omitted (OMTJRNE)

Specifies the journal entries to omit from the default journal. If *NONE is specified on the DFTJRN parameter, this parameter is ignored.

***NONE**
No journal entries are omitted.

***LUWID**
The journal entry that contains the Logical Unit of Work Identifier (LUWID) and all the resources involved in the logical unit of work are omitted if the logical unit of work is committed or rolled back successfully. If an error occurs while committing or rolling back the logical unit of work, the entry will always be sent regardless of this value.

Top

Examples

Example 1: Defining Activation Group Level Commitment Control

```
STRCMTCTL   LCKLVL(*CHG)  CMTSCOPE(*ACTGRP)  TEXT('Blue Commit Group')
```

This command described by the user as the Blue Commit Group starts the activation group level commitment for the activation group associated with the program issuing the command.

Only records that are updated, inserted, or deleted are locked until the transaction is ended by a commit or rollback operation. No identification for the commitment boundary is sent after the initial program load (IPL) following an abnormal system end, after an abnormal end to an activation group for the job, or when the job or activation group ends either with uncommitted changes or with a nonzero completion code.

Example 2: Defining Job Level Commitment Control

```
STRCMTCTL   LCKLVL(*ALL)  NFYOBJ(RCVLIB/MYFILE *FILE IDSAVE)
            CMTSCOPE(*JOB)  DFTJRN(MGWLIB/MYJRN)
```

This command starts the job level commitment definition. All records accessed in files opened under commitment control are locked until the commitment transaction is ended by a commit or rollback operation. If a commitment transaction ends in a manner that a notify object is to be updated with the commitment identifier of the last successful commit operation, the notify object to be updated is member IDSAVE of file MYFILE in the library RCVLIB. When a commit or rollback is done, an entry that lists information about all the resources involved in the logical unit of work is put into journal MYJRN in library MGWLIB.

Top

Error messages

***ESCAPE Messages**

CPF8351
Commitment control already active.

CPF8352
Attribute in notify object &1 type *&4 not valid.

CPF8360
Not enough storage for commitment control operation.

CPF8366
Commitment definition &2 not created. Reason code &1.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &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.

CPF9815
Member &5 file &2 in library &3 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

Top

Start Copy Screen (STRCPYSCN)

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

Parameters
Examples
Error messages

The Start Copy Screen (STRCPYSCN) command allows you to copy the screens of another display station on your display station to observe what is happening and diagnose problems.

If the STRCPYSCN command is used to copy displays from a source device that has the wide-display feature to an output device with a regular-width display, the command is accepted, but wide-display images are not shown and an informational message is sent to the target work station indicating that the display was not shown.

If the STRCPYSCN command is used to copy displays from a source device that supports graphic DBCS characters, the command is accepted and character information is shown, but graphic DBCS characters appear as single byte. No message is sent.

If the output device is not the requesting device, then the output device cannot be signed on. If the output device is signed on, a message is sent to the requester indicating that the device is not available for copying. If the source device is signed off after display copy has begun, the function automatically ends.

Note: The copy display function can be ended by the target device.

Restrictions:

- Permission must be given from the user of the source work station.
- When a request is made to begin display image copying, a break message is sent to the user of the source work station to inform the user that the displays are going to be copied. The user must reply to this message before any displays are copied.
- *REQUESTER is not valid for the SRCDEV or OUTDEV parameters when the command is submitted to batch.

Top

Parameters

Keyword	Description	Choices	Notes
SRCDEV	Source device	*Name*, *REQUESTER	Required, Positional 1
OUTDEV	Output device	*Name*, *REQUESTER, *NONE	Required, Positional 2
JOBQ	Job queue	*Qualified object name*	Optional
	Qualifier 1: Job queue	*Name*, **QCTL**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTFILE	File to receive output	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	

Top

Source device (SRCDEV)

Specifies the display station that is used as the source for the screen images to be copied.

This is a required parameter.

***REQUESTER**
Specifies that the screens are to be copied from the display station that issued this command.

name Specify the name of the display station (other than the one that issued this command) whose screens are to be copied.

Top

Output device (OUTDEV)

Specifies the output device for the copying process.

This is a required parameter.

***REQUESTER**
Specifies that screens are to be copied to the work station from which this command is issued. *REQUESTER cannot be specified here if it is also specified for the **Source device (SRCDEV)** parameter.

***NONE**
Specifies that the copied screens do not go to a display station. If *NONE is specified here, then a value must be specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the display station (other than the one that issued this command) that will display the copied screens.

Top

Job queue (JOBQ)

Specifies the job queue used to submit the job which shows the screens from the source device on the target device when the requesting device is not the target device. When *REQUESTER is specified on the **Output device (OUTDEV)** parameter, this parameter is ignored, since it defaults to the values for the target display station and then a submit job is not necessary.

Qualifier 1: Job queue

QCTL Job queue QCTL is to be used.

name Specify the name of the job queue where the job which will process the copied screens is to be submitted.

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 job queue. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the job queue is located.

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 QASCCPY in system library QSYS with a format name of QSCCPY1 is used as a model.

Top

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

Element 2: Replace or add records

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

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

Top

Examples

```
STRCPYSCN   SRCDEV(WS2)  OUTDEV(*REQUESTER)
```

This command sends an inquiry message to the user of work station, WS2. The message indicates that the display station displays are about to be copied to another display station. If the user of that display does not wish this to happen, then a cancel (C) reply prevents the operation from beginning. To allow the operation to begin, the user responds with a go (G) reply to the message.

Top

Error messages

***ESCAPE Messages**

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

CPF7AF4
Library QTEMP is not valid for OUTFILE keyword.

CPF7AF5
From device cannot be used with to device.

CPF7AF6
Device &1 not available.

CPF7AF7
Device name &1 not correct.

CPF9845
Error occurred while opening file &1.

CPF9860
Error occurred during output file processing.

***NOTIFY Messages**

CPI7AF9
Screen image not displayed.

Top

Start Cluster Resource Group (STRCRG)

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

Parameters
Examples
Error messages

The Start Cluster Resource Group (STRCRG) command enables resiliency for the specified cluster resource group on the specified recovery domain nodes. This command can be used to change the cluster resource group from a status of Indoubt or Inactive to Active.

This command will do the following for all cluster resource group types:

1. Set the cluster resource group status to Start Pending.
2. For primary-backup model cluster resource groups if the current recovery domain has more than one backup node and some backup nodes are not active, the recovery domain may be reordered so that all active backup nodes are ordered before inactive backup nodes. If the inactive backup nodes are already ordered after active backup nodes or if there are no inactive backup nodes, this command will not change the recovery domain.
3. Call the exit program on all active nodes in the recovery domain with an action code of Start, if an exit program is specified for the cluster resource group.
4. Set the cluster resource group status to Active if the exit program is successful on all active nodes in the recovery domain.
5. If the exit program is unsuccessful on any active node in the recovery domain when processing the Undo action code, the status of the cluster resource group will be set to Indoubt on all nodes.

This command will do the following for application cluster resource groups:

1. Verify the takeover IP address has been configured on all nodes in the recovery domain except replicates.
2. Start the TCP/IP interface on the primary node.
3. If the exit program is unsuccessful on any active node in the recovery domain, the exit program job on the primary node will be cancelled and the Takeover IP address will be ended on the primary node.

   For an application cluster resource group, the exit program on the primary node is not expected to complete. The status of the cluster resource group will be set to Active when the exit program job has been started on the primary and the exit program has completed successfully on all other nodes in the recovery domain.

This command will do the following for device cluster resource groups:

1. The cluster resource group must have at least one configuration object entry.
2. The configuration objects specified for the cluster resource group must exist on all active nodes in the recovery domain and the resource name specified in a configuration object must be the same on all active nodes in the recovery domain.
3. If a data base name has been specified for a configuration object, it must be the same on all active nodes in the recovery domain.
4. If a server takeover IP address is specified, it must exist on all nodes in the recovery domain.
5. The primary node must be the current owner of all input/output processors or high-speed link I/O bridges for the devices in the cluster resource group.
6. The IOP or high-speed link I/O bridge controlling the devices specified in the cluster resource group must be accessible by all active nodes in the cluster resource group's recovery domain.

© Copyright IBM Corp. 1998, 2006

7. Starting the cluster resource group will not vary on the configuration objects or start the server takeover IP address.
8. All members of an auxiliary storage pool group must be configured in the cluster resource group.
9. A value of *PRIMARY for the device's configuration object online' attribute can be specified only for a secondary auxiliary storage pool.

This command will do the following for peer cluster resource groups:

1. All nodes that are designated with a peer role will be an active access point upon successful completion of this command.

Restrictions:

1. You must have input/output system configuration (*IOSYSCFG) special authority to run this command.
2. This command cannot be called from a cluster resource group exit program.
3. Cluster Resource Services must be active on the node running the command.
4. For primary-backup module cluster resource groups, the status of the node currently assigned the role of primary must have a membership status of Active.
5. For peer model cluster resource groups:
   - At least one node must be designated as a peer node or the cluster resource group cannot be started.
   - At least one peer node must have a membership status of Active.
6. The cluster resource group status must be either Inactive or Indoubt.
7. For a device cluster resource group:
   a. The cluster resource group must have at least one configuration object entry.
   b. The configuration objects specified for the cluster resource group must exist on all active nodes in the recovery domain and the resource name specified in a configuration object must be the same on all active nodes in the recovery domain.
   c. The primary node must be the current owner of all input/output processors or high-speed link I/O bridges for the devices in the cluster resource group.
   d. Hardware configuration must be complete so that the physical hardware has been associated with the configuration object. In particular for auxiliary storage pools, the disk units must have been assigned.
   e. The input/output processor or high-speed link I/O bridge controlling the devices specified in the cluster resource group must be accessible by all active nodes in the cluster resource group's recovery domain.

Top

Parameters

Keyword	Description	Choices	Notes
CLUSTER	Cluster	*Name*	Required, Positional 1
CRG	Cluster resource group	*Name*	Required, Positional 2
EXITPGMDTA	Exit program data	*Character value*, ***SAME**	Optional

Top

Cluster (CLUSTER)

Specifies the cluster containing the cluster resource group.

This is a required parameter.

name Specify the name of the cluster.

Top

Cluster resource group (CRG)

Specifies the cluster resource group to be started.

This is a required parameter.

name Specify the name of the cluster resource group.

Top

Exit program data (EXITPGMDTA)

Specifies up to 256 bytes of data that is passed to the cluster resource group exit program when it is called. This parameter may contain any scalar data except pointers. For example, it can be used to provide state information. This data will be stored with the specified cluster resource group and copied to all nodes in the recovery domain. Pointers in this area will not resolve correctly on all nodes and should not be placed in the data. The data specified will replace the existing exit program data stored with the cluster resource group. If blanks are specified, then the exit program data stored with the cluster resource group will be cleared. This parameter must be set to *SAME if no exit program is specified for the cluster resource group.

***SAME**
The exit program data stored with the cluster resource group specified will be passed to the exit program.

character-value
Specify the data to be passed to the exit program.

Top

Examples

```
STRCRG   CLUSTER(MYCLUSTER)  CRG(MYCRG)
         EXITPGMDTA('important information')
```

This command enables resiliency of the cluster resource group MYCRG in the cluster MYCLUSTER. The exit program, when called, is passed 'important information' in the exit program data field. The cluster resource group will be ACTIVE as a result of this command.

Top

Error messages

***ESCAPE Messages**

CPF1999
Errors occurred on command.

Top

Start Debug (STRDBG)

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

Parameters
Examples
Error messages

The Start Debug (STRDBG) command puts a job into debug mode and, optionally, adds as many as 20 programs and 20 service programs and 20 class files to debug mode. It also specifies certain attributes of the debugging session. For example, it can specify whether database files in production libraries can be updated while in debug mode.

The graphical system debugger will be launched instead of the traditional system debugger if the STRDBG command is issued by a user registered for graphical debugging.

Debug can operate in three environments at the same time. They are OPM (Original Program Model), ILE (Integrated Language Environment), and JAVA. Some parameters for this command are applicable for all three environments. Some parameters for this command are applicable only for OPM, ILE, or JAVA. There will be a statement in each parameter description stating the environment in which the parameter is applicable.

The Change Debug (CHGDBG) command can be used later in the job to change the attributes of the debug mode. Also, OPM programs can be added to or removed from the debugging session if they are specified in the Add Program (ADDPGM) or Remove Program (RMVPGM) commands. OPM programs added with the ADDPGM command will be in the OPM debug environment.

When one job is servicing another job, and STRDBG is entered, all debug commands are valid for the job being serviced. If that job is held on a job queue, no further debug commands may be entered until that job is allowed to run. When the job starts, an initial breakpoint screen is displayed. From this screen, additional debug commands may be entered. To service another job, see the STRSRVJOB (Start Service Job) command. More information about debugging one job from another job is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Restrictions:

- You cannot use this command in debug mode. To end debug mode, refer to the End Debug (ENDDBG) command.
- You cannot use this command if you are servicing another job, and that job is held, suspended, or ending.
- This command is shipped with public *EXCLUDE authority and the QPGMR, QSRV, and QSRVBAS user profiles have private authorities to use this command.
- You must have either *CHANGE authority to the program, or *USE authority to the program and *SERVICE special authority.

Top

© Copyright IBM Corp. 1998, 2006

Parameters

Keyword	Description	Choices	Notes
PGM	Program	Single values: ***NONE** Other values (up to 20 repetitions): *Qualified object name*	Optional, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DFTPGM	Default program	*Name*, ***PGM**, *NONE	Optional
MAXTRC	Maximum trace statements	*Integer*, **200**	Optional
TRCFULL	Trace full	***STOPTRC**, *WRAP	Optional
UPDPROD	Update production files	***NO**, *YES	Optional
OPMSRC	OPM source level debug	***NO**, *YES	Optional
SRVPGM	Service program	Single values: ***NONE** Other values (up to 20 repetitions): *Qualified object name*	Optional
	Qualifier 1: Service program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CLASS	Class file	Values (up to 20 repetitions): *Path name*, ***NONE**	Optional
DSPMODSRC	Display module source	***PGMDEP**, *NO, *YES	Optional
SRCDBGPGM	Source debug program	Single values: ***SYSDFT** Other values: *Qualified object name*	Optional
	Qualifier 1: Source debug program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
UNMONPGM	Unmonitored message program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Unmonitored message program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Program (PGM)

Specifies up to 20 programs to debug in the job. Before a program can be debugged, its name must be specified on this parameter, in the Add Program (ADDPGM) command, or the Display Module Source (DSPMODSRC) command.

This parameter is applicable in both OPM and ILE environments.

Single values

***NONE**
No program names are specified at the start of the debugging session. The Add Program (ADDPGM) command or the Display Module Source (DSPMODSRC) command can be used to add programs later.

Other values (up to 20 repetitions)

Qualifier 1: Program

name Specify the name of the program to debug. You cannot debug two programs that have the same name at the same time in the OPM debug environment. In the ILE debug environment, two programs with the same name can be debugged if they are in different libraries.

Qualifier 2: Library

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

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

name Specify the library where the program is located.

Top

Default program (DFTPGM)

Specifies the original program model (OPM) program to use as the default program during debug mode. The program specified here is used as the default program for any of the other debug commands for which the user specified a value of *DFTPGM for the **Program (PGM)** parameter. That is, if a default program was previously specified, this parameter can change it.

This parameter is applicable only in the OPM environment.

***PGM** The program named on the PGM parameter of this command is to be the default program for the job's debugging session. If there is more than one program name specified on the PGM parameter, the first OPM program named in the list that is added to the OPM debug environment is the default program. If *NONE is specified on the PGM parameter, or is the default, *NONE is also assumed when this value (*PGM) is specified.

***NONE**
No program is specified as the default program; if a program was specified as a default program, it is no longer the default program. If the job has no default program, *DFTPGM cannot be specified on the PGM parameter of any other debug commands.

name Specify the name of the program to use as the default program during debug mode. The same name (in qualified form) must also be specified for the PGM parameter of this command. You cannot specify a bound program on this parameter.

Note: If OPMSRC(*YES) is specified, the OPM program may be added to either the OPM debug environment or the ILE debug environment. If added to the ILE debug environment, then it cannot be a default program for the OPM debug environment.

Top

Maximum trace statements (MAXTRC)

Specifies the maximum number of trace statements that the system puts into the job's trace file before either stopping tracing or wrapping around (overlaying) on the trace file. When the trace file contains the maximum specified, the system performs the actions specified by the value supplied for the **Trace full (TRCFULL)** parameter.

Note: Instruction stepping can be performed on a program being debugged in an interactive environment by setting the maximum number of trace statements to **1** and the value for the TRCFULL parameter to *STOPTRC.

This parameter is applicable only in the OPM environment.

200 Two hundred trace statements can be put into the file before tracing is stopped or wrapping occurs.

integer
Specify the maximum number of trace statements that can be in the trace file.

Top

Trace full (TRCFULL)

Specifies what happens when the job's trace file is full (that is, it contains the maximum number of trace statements specified by the value supplied for the **Maximum trace statements (MAXTRC)** parameter.

This parameter is applicable only in the OPM environment.

***STOPTRC**
In a batch environment, tracing stops but the program continues processing. In an interactive environment, control is given to the user when a breakpoint occurs. If the user continues processing, a breakpoint occurs before processing each subsequent statement within the range of statements being traced, and the trace file is extended to contain the new entry.

***WRAP**
The trace file is overlaid with new trace statements as they occur, wrapping from the beginning of the file. The program will complete processing with no message to indicate that wrapping has occurred. The trace file never has more than the maximum specified statements, and they are the most recently recorded statements.

Top

Update production files (UPDPROD)

Specifies whether or not database files in a production library can be opened for updating records, or for adding new records, while the job is in debug mode. If not, the files must be copied into a test library before trying to run a program that uses the files.

This parameter is applicable in both OPM and ILE environments.

***NO** Database files in production libraries cannot be updated while the job is in debug mode. Database files can be opened for reading only. This protects database files from unwanted updates while a program is being debugged. The exception to this is starting debug mode after a production library is already opened. If this value is specified, some commands, which may include function keys, menu options, or pulldowns, may not work properly.

***YES** Database files in production libraries can be updated while the job is in debug mode.

Top

OPM source level debug (OPMSRC)

Specifies that original program model (OPM) programs are debugged using the system source debug support (same as ILE source debug).

This parameter is valid for OPM CL, OPM RPG and OPM COBOL programs that were created with OPTION(*SRCDBG) with CRTCLPGM, CRTRPGPGM and CRTCBLPGM commands. Additionally, this parameter is valid for OPM CL, OPM RPG and OPM COBOL programs that were created with OPTION(*LSTDBG) with the CRTCLPGM, CRTRPGPGM, CRTCBLPGM, CRTSQLRPG, CRTSQLCBL, and CRTRPTRPG commands. If the OPM program is not CL, RPG or COBOL and was not compiled with a valid debug option, then this parameter is ignored.

This parameter is applicable in both OPM and ILE environment.

ILE functions will now operate on OPM programs. For example, source level debug.

***NO** OPM debug functions are used for OPM programs.

***YES** ILE debug functions are used for OPM programs.

Top

Service program (SRVPGM)

Specifies up to 20 service programs to debug in the job. Before a service program can be debugged, its name must be specified on this parameter or the Display Module Source (DSPMODSRC) command.

This parameter is applicable only in the ILE environment.

Single values

***NONE**
No service program names are specified at the start of the debugging session. The Display Module Source (DSPMODSRC) command can be used to add service programs later.

Other values (up to 20 repetitions)

Qualifier 1: Service program

name Specify the name of the service program to debug.

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 service program. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the service program is located.

Top

Class file (CLASS)

Specifies up to 20 class files to debug in the job. Before a class file can be debugged, its name must be specified on this parameter or the Display Module Source (DSPMODSRC) command.

This parameter is applicable only in the ILE environment.

***NONE**
No class files are specified at the start of the debugging session. The Display Module Source (DSPMODSRC) command can be used to add class files later.

path-name
Specify the path names of the class file to debug.

Top

Display module source (DSPMODSRC)

Specifies whether the first display of the source debug program is shown when this command is processed and no errors occur.

This parameter is applicable only in the ILE environment.

***PGMDEP**
The showing of the source debug program display is dependent on the programs specified on the **Program (PGM)** parameter. If any of the programs are ILE programs, the display is shown. If any of the programs are OPM programs with source debug data, and OPMSRC(*YES) is specified, the display is shown. If all of the programs are OPM programs and OPMSRC(*NO) is specified, the display is not shown.

***NO** The first display of the source debug program is not shown.

***YES** The first display of the source debug program is shown.

Top

Source debug program (SRCDBGPGM)

Specifies the source debug program to be used. See the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter for an explanation of this parameter.

This parameter is applicable only in the ILE environment.

Single values

***SYSDFT**
The system source debug program is used.

Qualifier 1: Source debug program

name Specify the name of the program to be used to debug programs.

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

Unmonitored message program (UNMONPGM)

Specifies the qualified name of the user-supplied program called when a message that is not monitored occurs in the job being debugged. When the program specified is called, it is passed parameters that identify the program name, the recursion level, the high-level language statement identifier, the machine instruction number at which the breakpoint occurred, the message that was not monitored, the message data, the length of the message data, and the message reference key.

This parameter is applicable only in the OPM environment.

The passed parameters have the following formats:

1. Program name (10 bytes). Specifies the name of the program in which the breakpoint was reached.
2. Recursion level (5 bytes). Specifies the recursion level number of the program in which the breakpoint was reached. This value is a 1- to 5-digit number padded on the right with blanks.
3. Statement Identifier (10 bytes). Specifies the high-level language program statement identifier that was reached. This is the statement identifier specified on the Add Breakpoint (ADDBKP) command. If a machine instruction number is used to specify the breakpoint, this parameter contains a slash (/) followed by a 4-digit hexadecimal machine instruction number.
4. Instruction number (5 bytes). Specifies the machine instruction number that corresponds to the high-level language statement at which the breakpoint was reached. No slash appears in front of the machine instruction number. The value consists of 1 to 4 hexadecimal characters representing the MI instruction number, followed by one or more blanks. If a machine instruction number is passed on the third parameter, the numbers in the third and fourth parameters are the same.
5. Message ID (7 bytes). Specifies the ID of the message that was not monitored.
6. Message data (256 bytes). Specifies the first 256 bytes of message data sent with the message not monitored.
7. Message data length (5 bytes). Specifies the length of the message data sent with the message not monitored.
8. Message MRK (4 bytes). Specifies the message reference key (MRK) of the message not monitored.

All the parameter values are left-adjusted and padded on the right with blanks. When control returns to the program with the message that was not monitored, processing continues.

Single values

***NONE**
No program is called when a message that is not monitored occurs.

Qualifier 1: Unmonitored message program

name Specify the name of the user-supplied program called when a message that is not monitored occurs in the job being debugged. After the program runs, control is returned to the interrupted 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.

name Specify the name of the library to be searched.

Top

Examples

Example 1: Start Debug for Program

```
STRDBG   PGM(TESTLIB/PAYROLL)  UPDPROD(*NO)
```

This command starts debug mode to debug the program PAYROLL, which is in the test library TESTLIB. If tracing is used, up to 200 trace statements can be stored in the trace before tracing stops. If program PAYROLL is a bound program, the Display Module Source display will be shown, giving the source for the module that contains the program entry point. Any database files updated by the PAYROLL program must be in a test library.

Example 2: Start Debug for Class File

```
STRDBG   CLASS('financial.payapplet')
```

This command starts debug mode to debug the class **payapplet** in the **financial** package.

Top

Error messages

***ESCAPE Messages**

CPF1999

Errors occurred on command.

Top

Start Debug Server (STRDBGSVR)

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

Parameters
Examples
Error messages

The Start Debug Server (STRDBGSVR) command starts the debug server router function. Only one debug server router can be active at a time. Once started, the debug server router remains active until it is ended using the ENDDBGSVR command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
STRDBGSVR
```

This command starts the debug server router function.

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start Database Monitor (STRDBMON)

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

Parameters
Examples
Error messages

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

For each monitor started using the STRDBMON commmand, the system generates a monitor ID that can be used to uniquely identify each individual monitor. The monitor ID can be used on the ENDDBMON command to uniquely identify which monitor is to be ended. The monitor ID is returned in the informational message CPI436A which is generated for each occurrence of the STRDBMON command. The monitor ID can also be found in column QQC101 of the QQQ3018 database monitor record.

Restrictions:

1. There are two types of monitors. A private monitor is a monitor over one, specific job (or the current job). A public monitor is a monitor which collects data across mulitple jobs. Only one (1) monitor can be started on a specific job at a time (i.e. only one private monitor can be active over any specific job). For example, STRDBMON JOB(*) followed by another STRDBMON JOB(*) within the same job is not allowed. There can be a maximum of ten (10) public monitors active at any one time. For example, STRDBMON JOB(*ALL) followed by another STRDBMON JOB(*ALL) is allowed providing the maximum number of public monitors does not exceed 10. You may have 10 public monitors and 1 private monitor active at the same time for any specific job.
2. If multiple monitors specify the same output file, only one copy of the database statistic records will be written to the specified output file for each job. For example, STRDBMON OUTFILE(LIB/TABLE1) JOB(*) and STRDBMON OUTFILE(LIB/TABLE1) JOB(*ALL) both use the same output file. For the current job, you will not get two copies of the database statistic records, one copy for the private monitor and one copy for the public monitor. You will get only one copy of the database statistic records.
3. QTEMP cannot be specified as the library on the OUTFILE parameter unless JOB(*) was also specified.
4. This command is conditionally threadsafe. For multithreaded jobs, this command is not threadsafe and may fail when the OUTFILE parameter is a distributed file or is a Distributed Data Management (DDM) file of type *SNA.
5. Any public monitor requires the file specified for the OUTFILE parameter to be in a library that resides in the system ASP.

Top

Parameters

Keyword	Description	Choices	Notes
OUTFILE	File to receive output	*Qualified object name*	Required, Positional 1
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member to receive output	*Name*, ***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
JOB	Job name	Single values: ***** Other values: *Qualified job name*	Optional
	Qualifier 1: Job name	*Generic name, name*, *ALL	
	Qualifier 2: User	*Generic name, name*, *ALL	
	Qualifier 3: Number	000000-999999, *ALL	
TYPE	Type of records	***BASIC**, *DETAIL, *SUMMARY	Optional
FRCRCD	Force record write	0-32767, ***CALC**	Optional
RUNTHLD	Run time threshold	0-2147483647, ***NONE**	Optional
INCSYSSQL	Include system SQL	***NO**, *YES, *INI	Optional
FTRFILE	Filter by database file	Single values: ***NONE** Other values (up to 10 repetitions): *Qualified object name*	Optional
	Qualifier 1: Filter by database file	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Generic name, name*	
FTRUSER	Filter by user profile	*Generic name, name*, ***NONE**, *CURRENT	Optional
FTRINTNETA	Filter by internet address	*Character value*, ***NONE**	Optional
COMMENT	Comment	*Character value*, ***BLANK**	Optional

Top

File to receive output (OUTFILE)

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

This is a required parameter.

Qualifier 1: File to receive output

name Specify the name of the file.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Top

Output member options (OUTMBR)

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

Element 1: Member to receive output

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

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

Element 2: Replace or add records

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

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

Top

Job name (JOB)

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

Single values

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

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

Qualifier 1: Job name

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

generic-name
Specify the generic name of the jobs whose database monitor are to be started. All jobs matching the specified generic name, including jobs waiting on job queues, will be monitored.

Qualifier 2: User

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

generic-name
Specify the generic name of the user whose jobs are to be monitored. All jobs matching the specified generic name, including jobs waiting on job queues, will be monitored.

Qualifier 3: Number

000000-999999
Specify the number of the job whose database monitor is to be started.

Top

Type of records (TYPE)

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

***BASIC**
Only the basic database monitor records are collected.

***DETAIL**
Both basic and detail database monitor records are collected. The detail database monitor record (QQQ3019) contains a count of the number of synchronous and asynchronous reads and writes to the database, as well as other database counts.

***SUMMARY**
Only the basic database monitor records are collected.

Top

Force record write (FRCRCD)

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

***CALC**
The system will calculate the number of records to be held in the buffer.

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

Top

Run time threshold (RUNTHLD)

Specifies a filtering threshold based on the estimated run time of the SQL statement. Monitor records will be created only for those SQL statements whose estimated run time meets or exceeds the specified run time threshold. If the estimated run time of the SQL statement is less than the specified threshold then no monitor records will be created for that SQL statement.

***NONE**
A run time threshold is not specified. All monitor records will be created.

runtime-threshold
Specify the run time threshold, in seconds. Monitor records will be created for all SQL statements whose estimated run time meets or exceeds this value.

Top

Include system SQL (INCSYSSQL)

Specifies whether or not monitor records will be created for system-generated SQL statements. Monitor records will always be created for user-specified SQL statements. This option determines if monitor records will also be created for SQL statements generated internally by the system.

***NO** No monitor records will be created for system-generated SQL statements. Monitor records will only be created for user-specified SQL statements.

***YES** Monitor records will be created for both user-specified and system-generated SQL statements.

***INI** Monitor records will be generated based on the value of the SQL_DBMON_OUTPUT option in the current INI file. A value of *USER or *DEFAULT creates monitor records for just user-specified SQL statements. A value of *SYSTEM creates monitor records for just system-generated SQL statements. A value of *ALL creates monitor records for both user-specified and system-generated SQL statements. If no INI file exists, then a default value of *NO will be used for the INCSYSSQL option.

Top

Filter by database file (FTRFILE)

Specifies a filter based on the name of the file and library used within the SQL statement. Monitor records will be created only for those SQL statements that use the qualified file. The specified file name can be either the 10-byte short name or the 256-byte long name.

Single values

***NONE**
No filtering by file is specified.

Qualifier 1: Data base file

***ALL** Monitor records will be created for any SQL statement that uses any file in the specified library. If none of the files used in the SQL statement come from the specified library, no monitor monitor records will be created for the SQL statement.

name Monitor records will be created only for those SQL statements that use the specified file. Monitor records will not be created for any SQL statements that do not use the specified file.

generic-name
Monitor records will be created only for those SQL statements that use a file that matches the generic prefix. If none of the files used in the SQL statement match the specified prefix, no monitor monitor records will be created for the SQL statement.

Qualifier 2: Library

name Monitor records will be created only for those SQL statements that use a file from the specified library. Monitor records will not be created if none of the files used in the SQL statement come from the specified library.

generic-name
Monitor records will be created only for those SQL statements that use a file from a library that matches the generic prefix. If none of the files used in the SQL statement come from the generic library, no monitor monitor records will be created for the SQL statement.

Top

Filter by user profile (FTRUSER)

Specifies a filter based on a user profile name. Monitor records will be created only for those SQL statements that are executed by the specified user. Monitor records will not be created for SQL statements executed by a different user.

***NONE**
Filtering by user is not specified.

***CURRENT**
Monitor records will be created only for those SQL statements that are executed by the user who is invoking the STRDBMON command. Monitor records will not be created for SQL statements executed by a different user.

user-name
Monitor records will be created only for those SQL statements that are executed by the specified user. Monitor records will not be created for SQL statements executed by a different user.

generic-user-name
Monitor records will be created only for those SQL statements that are executed by a user whose name starts with the specified prefix. Monitor records will not be created for SQL statements executed by a different user.

Top

Filter by internet address (FTRINTNETA)

Specifies filtering based on the internet address of a remote system. Monitor records will be created only for those SQL statements executed from the specified remote system.

***NONE**
Internet address filtering is not specified.

internet-address
The internet address is specified in the form **nnn.nnn.nnn.nnn**, where **nnn** is a decimal number ranging from 0 through 255, without the leading zeros. (An internet address having all binary ones or zeros in the bits of the network or host identifier portions of the address is not valid.)

Top

Comment (COMMENT)

User-specified description that is associated with the database monitor. The description is stored in the monitor record that has a record ID of 3018.

***BLANK**
Text is not specified.

character-value
Specify up to 100 characters of text.

Top

Examples

Example 1: Starting Database Monitoring For All Jobs

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

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

Example 2: Starting Database Monitoring For a Specific Job

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

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

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

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

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

Example 4: Starting Database Monitoring For All Jobs That Begin With 'QZDA'

```
STRDBMON   OUTFILE(LIB41/DBMONFILE)  JOB(*ALL/*ALL/QZDA*)
```

This command starts database monitoring for all jobs that whose job name begins with 'QZDA'. The performance statistics (monitor records) are added to member DBMONFILE (since OUTMBR was not specified) in file DBMONFILE in library LIB41. This library may exist in more than one independent auxiliary storage pool (ASP); the library in the name space of the originator's job will always be used.

Example 5: Starting Database Monitoring For All Jobs and Filtering SQL Statements That Run Over 10 Seconds

```
STRDBMON   OUTFILE(LIB41/DBMONFILE)  JOB(*ALL)  RUNTHLD(10)
```

This command starts database monitoring for all jobs. Monitor records are created only for those SQL statements whose estimated run time meets or exceeds 10 seconds.

Example 6: Starting Database Monitoring For the Current Job and Filtering Over a Specific File

```
STRDBMON   OUTFILE(LIB41/DBMONFILE) JOB(*)
           FTRFILE(LIB41/TABLE1)
```

This command starts database monitoring for the current job. Monitor records are created only for those SQL statements that use file LIB41/TABLE1.

Example 7: Starting Database Monitoring For the Current Job and the Current User

```
STRDBMON   OUTFILE(LIB41/DBMONFILE)  JOB(*)  FTRUSER(*CURRENT)
```

This command starts database monitoring for the current job. Monitor records are created only for those SQL statements that are executed by the current user.

Example 8: Starting Database Monitoring For Jobs Beginning With 'QZDA' and Filtering Over Run Time and File

```
STRDBMON   OUTFILE(LIB41/DBMONFILE)  JOB(*ALL/*ALL/QZDA*)
           RUNTHLD(10)  FTRUSER(DEVLPR1)  FTRFILE(LIB41/TTT*)
```

This command starts database monitoring for all jobs whose job name begins with 'QZDA'. Monitor records are created only for those SQL statements that meet all of the following conditions:

- The estimated run time, as calculated by the query optimizer, meets or exceeds 10 seconds
- Was executed by user 'DEVLPR1'.
- Use any file whose name begins with 'TTT' and resides in library LIB41.

Top

Error messages

***ESCAPE Messages**

CPF1321

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

CPF222E
&1 special authority is required.

CPF4269
Not authorized to object &1 in &2 type *&3.

CPF436A
Record format for file &1 in &2 does not match model file.

CPF436B
&1 can not be specified on the OUTFILE parameter.

CPF436C
Job &4 is already being monitored.

CPF436E
Job &1 user &2 job number &3 is not active.

CPF43A2
Address specified on FTRINTNETA parameter is not valid.

***STATUS Messages**

CPI436A
Database monitor started for job &1, monitor ID &2.

Top

Start Data Base Reader (STRDBRDR)

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

Parameters
Examples
Error messages

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

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

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

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

Top

Parameters

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

Top

© Copyright IBM Corp. 1998, 2006

Data base file (FILE)

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

This is a required parameter.

Qualifier 1: Data base file

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

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 file.

name Specify the name of the library where the file is located.

Top

Member (MBR)

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

***FIRST**
The first member in the file is used.

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

Top

Job queue (JOBQ)

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

Qualifier 1: Job queue

QBATCH
The job entry is placed on the QBATCH job queue.

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

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 job queue.

name Specify the name of the library where the job queue is located.

Top

Queue for reader messages (MSGQ)

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

Single values

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

Qualifier 1: Queue for reader messages

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

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

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 message queue. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the message queue is located.

Top

Reader (RDR)

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

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

name Specify the name that identifies the reader to be started.

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF1338
Errors occurred on SBMJOB command.

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

CPF3301
Reader &1 already started.

CPF3307
Job queue &1 in &2 not found.

CPF3330
Necessary resource not available.

CPF3362
Objects in QTEMP not valid for parameter values.

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

CPF3364
File &1 in library &2 not database file or DDM file.

CPF9812
File &1 in library &2 not found.

CPF9815
Member &5 file &2 in library &3 not found.

Top

Start DFU (STRDFU)

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

Parameters
Examples
Error messages

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

Error messages for STRDFU

***ESCAPE Messages**

DFU0005
The command failed.

DFU0018
The Run option is not valid.

Top

Parameters

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

Top

Option (OPTION)

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

The possible values are:

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

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

© Copyright IBM Corp. 1998, 2006

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

Top

DFU program (DFUPGM)

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

The possible values are:

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

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

Top

Data base file (FILE)

Specifies the database file you want to change or display.

The possible values are:

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

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

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

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

Top

Member (MBR)

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

The possible values are:

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

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

Top

Examples

None

Top

Error messages

***ESCAPE Messages**

DFU0005
The command failed.

DFU0018
The Run option is not valid.

Top

Start Directory Shadowing (STRDIRSHD)

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

Parameters
Examples
Error messages

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

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

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

There are no parameters for this command.

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

Top

Parameters

None

Top

Examples

```
STRDIRSHD
```

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

Top

Error messages

***ESCAPE Messages**

CPF89A8
Unable to start job that controls directory shadowing.

Top

© Copyright IBM Corp. 1998, 2006

Start DNS Query (STRDNSQRY)

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

Parameters
Examples
Error messages

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

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

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

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

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

There are two parameters for this command:

1. HOSTNAME
2. DMNNAMSVR

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

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

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

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

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

help (or ?)
: Displays a list of subcommands for the STRDNSQRY (NSLOOKUP) tool.

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

© Copyright IBM Corp. 1998, 2006

lserver NAME

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

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

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

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

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

set debug
Show debugging information.

set nodebug
Do not show debugging information.

set d2 Show exhaustive (verbose) debugging information.

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

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

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

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

set nosearch
Do not use the srchlist option.

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

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

set vc Use TCP for queries instead of UDP.

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

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

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

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

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

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

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

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

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

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

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

A IP Address record. This is the default value.

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

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

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

MX Mail Exchange record.

NS Name server (DNS server) information for the zone

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

SOA Start of Authority record.

TXT Text record.

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

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

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

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

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

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

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

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

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

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

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

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

ls -t TYPE DOMAIN
List all DNS TYPE records for DOMAIN. Substitute a DNS record type for TYPE, and a domain name for DOMAIN. See the 'set type=X' subcommand for a list of DNS record types.

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

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

Top

Parameters

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

Top

Host (HOSTNAME)

Specify the TCP/IP host that you want the Domain Name Service (DNS) server to supply information about. You can use either the host's worded name, or its numerical IP address. For example, a host's name is anybiz.usa.com, and its numerical address is 123.4.56.7. Use either the name to obtain the address, or the address to obtain the name.

***NONE**
No host name is provided. This is the default value for the HOSTNAME parameter and is used if you do not supply a host name or address. An interactive NSLookup session will be started.

host-name
Specify the name of the host to use for the DNS server query.

host-IP-address
Specify the IP address of the host to use for the DNS server query.

Top

Domain Name Server (DMNNAMSVR)

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

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

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

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

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

domain-name-server-name
Specify the name of a DNS server.

domain-name-server-IP-address
Specify the IP address of a DNS server.

Top

Examples

```
STRDNSQRY   HOSTNAME('9.12.234.14')  DMNNAMSVR(*CFG)
```

This command starts a DNS query using the host located at IP address 9.12.234.14 and the DNS server that is currently designated for use by this system.

Top

Error messages

None

Top

Start Disk Reorganization (STRDSKRGZ)

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

Parameters
Examples
Error messages

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

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

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

Top

Parameters

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

Top

Auxiliary storage pool ID (ASP)

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

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

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

Top

ASP device (ASPDEV)

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

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

© Copyright IBM Corp. 1998, 2006

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

Top

Time limit (TIMLMT)

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

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

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

Top

Examples

Example 1: Starting Disk Reorganization for ASP 1

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

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

Example 2: Starting Disk Reorganization with a Time Limit

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

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

Example 3: Starting Disk Reorganization for an ASP Device

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

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

Top

Error messages

***ESCAPE Messages**

CPF1888
Disk reorganization for ASP &1 already started.

CPF1890
*ALLOBJ authority required for requested operation.

CPF1891
Disk reorganization cannot be started on an ASP with only one unit.

Top

Start Education (STREDU)

Where allowed to run:

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

Threadsafe: No

Parameters
Examples
Error messages

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

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
STREDU
```

This command shows the following menus:

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

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start 3270 Display Emulation (STREML3270)

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

Parameters
Examples
Error messages

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

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

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

Top

Parameters

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

© Copyright IBM Corp. 1998, 2006

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

Top

Emulation controller (EMLCTL)

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

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

Top

Emulation device (EMLDEV)

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

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

Top

Emulation location (EMLLOC)

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

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

Top

Display device, batch only (DSPDEV)

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

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

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

Top

Page Up (Roll Down) key (PAGEUP)

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

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

***NONE**
No function is assigned to the Page Up (Roll Down) key. When there are fewer input fields on the display than allowed by the 5250 display device, this key has no function.

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

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

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

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

Top

Page Down (Roll Up) key (PAGEDOWN)

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

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

***NONE**
No function is assigned to the Page Down (Roll Up) key. When there are fewer input fields on the display than allowed by the 5250 display device, the key has no function.

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

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

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

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

Top

Test Request key (TESTREQ)

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

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

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

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

Top

Cursor Select key (CSRSLT)

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

***NONE**
A physical function key is not assigned to emulate the cursor select key. The real Cursor Select key is used.

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

Top

SNA DBCS 3270PC emulation (IGCEMLPC)

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

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

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

Top

Emulation printer device (EMLPRTDEV)

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

***NONE**
SNA DBCS 3270PC printer emulation with device emulation is not used. No printer emulation device is selected.

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

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

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

Top

Timeout wait for host (INZWAIT)

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

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

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

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

Top

Numeric lock keyboard (NUMLCK)

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

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

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

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

Top

Handle nulls (NULLS)

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

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

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

Top

Host signon/logon command (LOGON)

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

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

***NONE**
No text is sent to the host system after 3270 emulation is started.

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

Top

Wait response (WAITRSP)

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

***NO** The emulation device does not wait to send a positive response. It sends the response as soon as the data is received to the workstation display.

***YES** The emulation device waits until the data received is shown on the workstation display to send a positive response.

Top

End emulation conditions (ENDCOND)

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

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

The possible values are:

***NONE**
No additional ways to end 3270 device emulation are requested.

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

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

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

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

Top

Attention emulation menu (ATNEMLMNU)

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

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

The possible values are:

***YES** The Select 3270 Emulation Option for SNA or BSC menu is displayed when you press the Attention key.

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

Top

Function key program (FKEYPGM)

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

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

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

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

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

The possible **program name** values are:

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

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

The possible library values are:

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

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

name Specify the name of the library to be searched.

The possible **function key** values are:

***ALLFKEYS**
All function keys call the specified user-exit program.

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

Top

Keyboard language type (KBDTYPE)

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

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

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

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

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

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

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

keyboard-language-identifier
Specify the keyboard language identifier to be used.

Top

Outgoing translation table (TRNTBLOUT)

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

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

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

The possible library values are:

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

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

name Specify the name of the library to be searched.

Top

Incoming translation table (TRNTBLIN)

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

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

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

The possible library values are:

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

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

name Specify the name of the library to be searched.

Top

Configuration entry (EMLCFGE)

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

The possible values are:

***NONE**
No configuration entry is named and the configuration entry defaults are used.

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

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

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF2619
Table &1 not found.

CPF269A
Library parameter is not set to ″QSYS ″ on call.

CPF269B
T.61 conversion table not found.

CPF85EA
Screen address received from host is larger than screen size.

CPF85EB
3270 device emulation session ended.

CPF85EC
Specifying text on the LOGON parameter is not supported.

CPF85ED
Values other than ENDCOND(*NONE) are not supported.

CPF85E2
3270 display emulation is already active at this job.

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

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

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

CPF8503
Emulation controller &1 not found.

CPF8504
Controller &1 does not support 3270 emulation.

CPF8505
Emulation device &2 not found.

CPF8506
Emulation location &1 not found.

CPF8507
Display emulation cannot open required file.

CPF8508
Host system did not respond.

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

CPF8510
Internal error occurred on device &1.

CPF8511
Emulation ended by errors on device &2.

CPF8512
Emulation ended because device &2 was held.

CPF8513
Emulation ended by errors on device &2.

CPF8514
Error recovery stopped on device &1.

CPF8515
3270 emulation session ended by host.

CPF8516
No match between host and device &2.

CPF8517
Received more than maximum number of fields allowed.

CPF8518
Emulation ended because of internal failure in system.

CPF8519
Function check in 3270 emulation.

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

CPF852B
Program &1 not found.

CPF852C
Not authorized to program &1.

CPF8521
Not authorized to controller &1.

CPF8522
Not authorized to emulation device &2.

CPF8523
*NO on the ATNEMLMNU parameter is not supported.

CPF8524
Emulation cannot open its required display file.

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

CPF8526
No 3270 display emulation devices available.

CPF8527
Emulation device &2 not available.

CPF8528
Device &2 is not a display emulation device.

CPF8530
Not authorized to use any display emulation device.

CPF8533
Display device not specified in a batch job.

CPF8534
Display device &1 is not available.

CPF8535
Display device &1 not found.

CPF8536
Not authorized to display device &1.

CPF8539
&1 keyboard type not supported by 3270 emulation.

CPF8546
No 3270 display emulation devices available.

CPF8547
No 3270 printer emulation devices available.

CPF8550
Emulation ended due to time-out internal failure.

CPF8551
Emulation ended with error code &1.

CPF8552
Emulation ended because of return code.

CPF8553
BSC controller or device not allowed without translation.

CPF8565
Emulation device &1 not found.

CPF8568
Device &1 not printer emulation device.

CPF8569
Not authorized to use any printer emulation device.

CPF8571
No 3270 printer emulation devices available.

CPF8572
Emulation device &1 not available.

Top

Start EPM Environment (STREPMENV)

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

Parameters
Examples
Error messages

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

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

Error messages for STREPMENV

None

Top

Parameters

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

Top

Environment Name (EPMENV)

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

environment-name
Enter a name for the environment.

Top

Environment Program (ROOTPGM)

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

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

© Copyright IBM Corp. 1998, 2006

The possible values for library are:

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

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

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

Top

Examples

None

Top

Error messages

None

Top

Start Font Management Aid (STRFMA)

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

Parameters
Examples
Error messages

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

There are no parameters for this command.

Error messages for STRFMA

None

Top

Parameters

None

Top

Examples

None

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start Host Server (STRHOSTSVR)

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

Parameters
Examples
Error messages

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

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

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

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

- Server jobs QPWFSERVSO and QPWFSERVSS - these jobs run in the QSERVER subsystem or a user-defined subsystem
- Server job QPWFSERVS2 - this job runs in the QSERVER subsystem
- Server job QIWVPPJT - this job runs in the QSYSWRK subsystem
- Server job QTFPJTCP - this job runs in the QSERVER subsystem.

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

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

Restrictions:

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

Error messages for STRHOSTSVR

***ESCAPE Messages**

PWS300D
Unable to start host server daemon jobs.

© Copyright IBM Corp. 1998, 2006

PWS3006
Errors occurred starting server daemon jobs.

Top

Parameters

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

Top

Server type (SERVER)

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

The possible values are:

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

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

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

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

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

***NETPRT**
The network print server daemon is started in the QSYSWRK subsystem. The daemon job is named QNPSERVD. The associated server jobs are named QNPSERVS and QIWVPPJT.

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

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

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

Top

Required protocol (RQDPCL)

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

Single Values

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

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

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

Top

Examples

None

Top

Error messages

***ESCAPE Messages**

PWS300D
Unable to start host server daemon jobs.

PWS3006
Errors occurred starting server daemon jobs.

Top

Start IDDU (STRIDD)

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

Parameters
Examples
Error messages

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

Top

Parameters

None

Top

Examples

```
STRIDD
```

This commands displays the main IDDU menu.

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start IP over SNA Interface (STRIPSIFC)

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

Parameters
Examples
Error messages

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

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

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*	Required, Positional 1

Top

Internet address (INTNETADR)

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

This is a required parameter.

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPFA10F
IP over SNA interface &1 not started.

Top

© Copyright IBM Corp. 1998, 2006

Start ITF (STRITF)

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

Parameters
Examples
Error messages

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

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	*Communications name*	Required, Positional 1

Top

Remote location (RMTLOCNAME)

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

This is a required parameter.

Top

Examples

```
STRITF   CHICAGO
```

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

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start Journal (STRJRN)

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

Parameters
Examples
Error messages

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

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

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

For other ways to start journaling see the following commands:

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

Restrictions:

- The object must not be journaling changes to another journal.
- The maximum number of objects that can be associated with one journal is either 250,000 or 10,000,000. To get 10,000,000, the value of *MAX10M must have been specified for the JRNOBJLMT parameter on either the Create Journal (CRTJRN) command or on the Change Journal (CHGJRN) command. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.
- The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
- Byte stream files that are currently memory mapped, are virtual volume files, or are being used as IXS network storage spaces cannot be journaled.
- Objects that are internally marked as not eligible for journaling cannot be journaled. The system may mark system working directories that are created inside of user directories as not eligible for journaling.
- For data areas, only local external data area objects may be journaled. The special data areas (*LDA, *GDA, *PDA) and DDM data areas cannot be journaled.
- For data queues, only local data queues are supported. DDM data queues cannot be journaled.
- At least one of parameter OBJ or OBJFID must be specified.

© Copyright IBM Corp. 1998, 2006

Top

Parameters

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

Top

Objects (OBJ)

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

Element 1: Name

'object-path-name'

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

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

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

Element 2: Include or omit

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

***INCLUDE**

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

***OMIT**

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

Top

File identifier (OBJFID)

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

file-identifier
Objects identified with the FID are journaled.

Top

Journal (JRN)

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

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

Top

Directory subtree (SUBTREE)

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

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

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

***NONE**
Only the objects that match the selection criteria are processed. The objects within selected directories are not implicitly processed.

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

Top

Name pattern (PATTERN)

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

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

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

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

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

Element 1: Pattern

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

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

Element 2: Include or omit

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

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

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

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

Top

New objects inherit journaling (INHERIT)

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

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

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

Top

Images (IMAGES)

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

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

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

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

Top

Omit journal entry (OMTJRNE)

Specifies the journal entries that are omitted.

***NONE**
No entries are omitted.

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

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

Top

Examples

Example 1: Start Journaling with Omit of Directory

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

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

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

Example 2: Start Journaling with Pattern Matching

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

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

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

Example 3: Start Journaling with Omit by Pattern

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

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

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

Example 4: Start Journaling using File Identifiers

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

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

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

Example 5: Start Journaling on a Set of Data Queues

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

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

Top

Error messages

***ESCAPE Messages**

CPFA0D4
File system error occurred. Error number &1.

CPF700A
&1 of &2 objects have started journaling.

CPF705A
Operation failed due to remote journal.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

Start Journal Access Path (STRJRNAP)

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

Parameters
Examples
Error messages

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

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

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

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

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

For other ways to start journaling see the following commands:

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

Restrictions:

- Before journaling an access path, all physical files over which the access path is built must first be journaled to the same journal that is used to journal the access path. Even if all physical file members for a particular physical file are removed from the access path of a logical file, all physical files must still be journaled to the same journal before journaling the access path.
- The maximum number of objects that can be associated with one journal is either 250,000 or 10,000,000. To get 10,000,000, the value of *MAX10M must have been specified for the JRNOBJLMT parameter on either the Create Journal (CRTJRN) command or on the Change Journal (CHGJRN) command. If the number of objects is larger than this maximum, journaling does not start.
- All access paths to be journaled must specify MAINT(*IMMED) or MAINT(*DLY).
- If only after images are being journaled for the physical file members, the system automatically starts journaling the before and after images for the physical file once journaling is started for any access path built over the physical file. When journaling ends for the access paths, the system automatically stops journaling the before images for the physical file and again only journals the after images.
- Overrides are not applied to files specified on the FILE parameter.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance.

© Copyright IBM Corp. 1998, 2006

To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Top

Parameters

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

Top

Journaled file (FILE)

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

This is a required parameter.

Qualifier 1: Journaled file

file-name
Specify the name of the file.

Qualifier 2: Library

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

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

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

Top

Journal (JRN)

Specifies the journal that receives the file change journal entries.

This is a required parameter.

Qualifier 1: Journal

journal-name
Specify the name of the journal.

Qualifier 2: Library

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

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

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

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF6971
Damage prevents object &1 from being journaled.

CPF6972
Cannot allocate access path for file &1 in &2.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7004
Maximum number of objects journaled to journal &1.

CPF7008
Cannot start or end access path journaling for file &1.

CPF7009
Not all based-on files being journaled to &3.

CPF7011
Not enough storage or resources.

CPF703C
DDL transaction prevents journaling operation.

CPF703D
DDL transaction prevents journaling operation.

CPF703E
DDL transaction prevents journaling operation.

CPF7030
Object of type *&3 already being journaled.

CPF7031
Cannot allocate member &3 file &1 in &2.

CPF7033
Start or end journaling failed for member &3.

CPF7034
Logical damage of file &1 in &2.

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

CPF705A
Operation failed due to remote journal.

CPF7079
Access path journaling for file &1 not started.

CPF708D
Journal receiver found logically damaged.

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

CPF709D
Cannot start journaling object of type *&7.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

Start Journal Object (STRJRNOBJ)

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

Parameters
Examples
Error messages

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

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

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

For other ways to start journaling see the following commands:

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

Restrictions:

- The object must not be journaling changes to another journal.
- The maximum number of objects that can be associated with one journal is either 250,000 or 10,000,000. To get 10,000,000, the value of *MAX10M must have been specified for the JRNOBJLMT parameter on either the Create Journal (CRTJRN) command or on the Change Journal (CHGJRN) command. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.
- The specified journal and object must reside in the same Auxiliary Storage Pool (ASP).
- For data areas, only local external data area objects may be journaled. The special data areas (*LDA, *GDA, and *PDA) and DDM data areas cannot be journaled.
- For data queues, only local data queues are supported. DDM data queues cannot be journaled.

Top

© Copyright IBM Corp. 1998, 2006

Parameters

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

Top

Object (OBJ)

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

This is a required parameter.

Qualifier 1: Object

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

Qualifier 2: Library

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

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

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

Top

Object type (OBJTYPE)

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

This is a required parameter.

***DTAARA**
Changes for data area objects are to be journaled.

***DTAQ**
Changes for data queue objects are to be journaled.

Top

Journal (JRN)

Specifies the journal that receives the journaled changes.

This is a required parameter.

Qualifier 1: Journal

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

Qualifier 2: Library

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

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

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

Top

Images (IMAGES)

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

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

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

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

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF6979
Journal &1 in library &2 is unusable.

CPF700A
&1 of &2 objects have started journaling.

CPF705A
Operation failed due to remote journal.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

Start Journal Physical File (STRJRNPF)

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

Parameters
Examples
Error messages

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

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

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

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

For other ways to start journaling see the following commands:

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

Restrictions:

- The file must not be journaling changes to another journal.
- Overrides are not applied to files specified on the FILE parameter.
- The maximum number of objects that can be associated with one journal is either 250,000 or 10,000,000. To get 10,000,000, the value of *MAX10M must have been specified for the JRNOBJLMT parameter on either the Create Journal (CRTJRN) command or on the Change Journal (CHGJRN) command. If the number of objects is larger than this maximum, journaling does not start.
- The specified journal must be a local journal. Although all object types which can be journaled to a local journal can also have their changes sent to a remote journal, this is accomplished by a two step process. First start journaling to the local journal. Then connect the local journal to a remote instance. To initiate such a connection, use the Add Remote Journal (ADDRMTJRN) command or the Add Remote Journal (QjoAddRemoteJournal) API. For information about remote journaling, see the Journal management topic.
- In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
- If the file has Large Object (LOB) columns, and the total of the LOB columns plus the record size is greater than 15,761,440 bytes, then the file can only be journaled to a journal with RCVSIZOPT(*MAXOPT2) or RCVSIZOPT(*MAXOPT3) specified.

Top

© Copyright IBM Corp. 1998, 2006

Parameters

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

Top

Physical file to be journaled (FILE)

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

This is a required parameter.

Qualifier 1: Physical file to be journaled

file-name
Specify the name of a physical file.

Qualifier 2: Library

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

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

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

Top

Journal (JRN)

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

This is a required parameter.

Qualifier 1: Journal

journal-name
Specify the name of the journal.

Qualifier 2: Library

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

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

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

Top

Record images (IMAGES)

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

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

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

Top

Journal entries to be omitted (OMTJRNE)

Specifies the journal entries that are omitted.

***NONE**
No journal entries are omitted.

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

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF6971
Damage prevents object &1 from being journaled.

CPF6979
Journal &1 in library &2 is unusable.

CPF700D
File &1 in library &2 not journaled.

CPF7002
File &1 in library &2 not a physical file.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7004
Maximum number of objects journaled to journal &1.

CPF7011
Not enough storage or resources.

CPF703C
DDL transaction prevents journaling operation.

CPF703D
DDL transaction prevents journaling operation.

CPF703E
DDL transaction prevents journaling operation.

CPF7030
Object of type *&3 already being journaled.

CPF7031
Cannot allocate member &3 file &1 in &2.

CPF7033
Start or end journaling failed for member &3.

CPF7034
Logical damage of file &1 in &2.

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

CPF704B
Journaling started locally but distributed requests failed.

CPF705A
Operation failed due to remote journal.

CPF708D
Journal receiver found logically damaged.

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

CPF709D
Cannot start journaling object of type *&7.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9873
ASP status is preventing access to object.

CPF9875
Resources exceeded on ASP &1.

Top

Start Job Log Server (STRLOGSVR)

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

Parameters
Examples
Error messages

The Start Job Log Server (STRLOGSVR) command is used to start the job log server which writes job logs for jobs that are in a job log pending state.

The job log server will write a job's job log either to a spooled file, to a printer, or to an outfile, if specified to do so (by using the QMHCTLJL, Control job log API).

Restrictions:

- You must have job control (*JOBCTL) special authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
NBRSVR	Number of servers	1-30, ***CALC**	Optional, Positional 1

Top

Number of servers (NBRSVR)

Specifies the number of job log server jobs to be started. If one or more job log server jobs are already active, this is the number of additional server jobs that will be started. If the number of servers requested would exceed the maximum active, only the difference between the maximum and the current number of active servers will be started. The maximum number of job log servers that can be active at one time is 30.

***CALC**
The number of job log server jobs started will be determined by the system.

1-30 Specify the number of job log servers to start.

Top

Examples

```
STRLOGSVR   NBRSVR(*CALC)
```

This command starts the job log server. The number of server jobs started will be determined by the system.

Top

© Copyright IBM Corp. 1998, 2006

Error messages

***ESCAPE Messages**

CPF134A
Job log server not started.

Top

Start Mode (STRMOD)

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

Parameters
Examples
Error messages

The Start Mode (STRMOD) command starts one or all modes currently in use for an advanced program-to-program communications (APPC) remote location. The user can use STRMOD in either the reset or ended state; it is required only after an End Mode (ENDMOD) command has ended a mode. More information is in the APPC Programming book, SC41-5443.

Restriction: The user must have operational authority for the APPC device to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	*Communications name*	Required, Positional 1
DEV	Device	*Name*, ***LOC**	Optional, Positional 2
MODE	Mode	*Communications name*, ***NETATR**, *ALL	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional

Top

Remote location (RMTLOCNAME)

Specifies the remote location name.

This is a required parameter.

Top

Device (DEV)

Specifies the device description name.

The possible values are:

***LOC** The device description is determined by the system.

device-name
Specify the name of the device description.

Top

© Copyright IBM Corp. 1998, 2006

Mode (MODE)

Specifies the mode that is to be started.

The possible values are:

***NETATR**
The mode in the network attributes is used.

***ALL** All modes currently in use for the remote location are to be started.

- For a device description automatically created by the APPN support or a device description manually created with the APPN parameter specified as *YES, *ALL indicates that any modes that have been used while the remote location was active, but are not currently started, are to be started.
- For a device description manually created with the APPN parameter specified as *NO, *ALL specifies that all configured modes for the specified remote location are to be started.

BLANK
The mode name (consisting of 8 blank characters) is used.

mode-name
Specify a mode name.

Note: SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

Local location (LCLLOCNAME)

Specifies the local location name.

The possible values are:

***LOC** The local location name is determined by the system.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

local-location-name
Specify the name of your location. The local location name is specified if you want to indicate a specific local location name for the remote location.

Top

Remote network identifier (RMTNETID)

Specifies the remote network ID used with the remote location.

The possible values are:

***LOC** The system selects the remote network ID.

***NETATR**
The remote network identifier specified in the network attributes is used.

***NONE**
No remote network identifier (ID) is used.

remote-network-id
Specify the name of the remote network ID.

Top

Examples

```
STRMOD   RMTLOCNAME(APPCRLOC)  DEV(APPCDEV)  MODE(APPCMODE)
         RMTNETID(CHICAGO)
```

This command starts a mode named APPCMODE for a remote location named APPCRLOC, a device named APPCDEV, and a remote network ID of CHICAGO.

Top

Error messages

***ESCAPE Messages**

CPF598B
The &1 command failed for one or more modes.

Top

Start Mail Server Framework (STRMSF)

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

Parameters
Examples
Error messages

The Start Mail Server Framework (STRMSF) command starts the mail server framework jobs in the system work subsystem (QSYSWRK).

Top

Parameters

Keyword	Description	Choices	Notes
MSGOPT	How to process mail messages	***RESUME**, *RESET, *CLEAR	Optional, Positional 1
NBRMSFJOB	Number of MSF jobs	1-99, **3**	Optional, Positional 2

Top

How to process mail messages (MSGOPT)

Specifies how the mail server framework processes existing mail server framework messages.

The possible values are:

***RESUME**
All existing mail server framework messages continue processing from the point the mail server framework previously ended.

***RESET**
All existing mail server framework messages are processed as if they were just created.

***CLEAR**
All existing mail server framework messages are deleted. This option should only be used when a software error is reported with the mail server framework or its associated exit point programs.

Top

Number of MSF jobs (NBRMSFJOB)

Specifies the number of mail server framework jobs to start. This option allows concurrent processing of several mail server framework messages.

The possible values are:

3 Three jobs are started.

number-of-jobs
Specify the number of jobs you want handling mail server framework messages. The valid values range from 1 through 99.

© Copyright IBM Corp. 1998, 2006

Top

Examples

Example 1: Starting One Mail Server Framework Job

```
STRMSF   NBRMSFJOB(1)
```

This command starts one mail server framwork job in a normal manner, processing any mail server framework messages at the point at which processing was interrupted.

Example 2: Restarting Mail Server Framework Jobs

```
STRMSF   NBRMSFJOB(3)  MSGOPT(*RESET)
```

This command starts three mail server framework jobs and any mail server framework messages which were partially handled by previous mail server famework jobs are processed again from the beginning.

Top

Error messages

***ESCAPE Messages**

CPFAFAA
STRMSF did not complete successfully.

CPFAFAD
Mail Server Framework currently active.

CPFAFA0
Errors detected on MSF internal message index.

CPFAFA1
Errors detected on MSF internal message queue.

CPFAFFF
Internal system error in program &1.

Top

Start NFS Server (STRNFSSVR)

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

Parameters
Examples
Error messages

The Start Network File System Server (STRNFSSVR) command starts one or all of the following Network File System (NFS) server daemons. For more information about these daemon jobs, see Network File System book, SC41-5714

SERVER(*ALL) should be used, which will start the daemons in the following order. (This order is the recommended order for starting the Network File System daemons.)

- The Remote Procedure Call (RPC) RPCBind daemon
- The block input/output (I/O) (BIO) daemon
- The server (SVR) daemon
- The mount (MNT) daemon
- The network status monitor (NSM) daemon
- The network lock manager (NLM) daemon

If just one daemon is to be started, be sure the appropriate order for starting NFS daemons and the possible consequences of starting daemons in an order other than that specified above are understood. For more information about starting NFS daemons, see Network File System book, SC41-5714

If the user attempts to start a daemon or daemons that are already running, they will not cause the command to fail, and it will continue to start other daemons that were requested to start. The command will issue diagnostic message CPDA1BA or CPDA1BD if the daemon is already running. However, for best results, end NFS daemons before attempting the STRNFSSVR command.

To determine if an NFS daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following jobs:

```
QNFSRPCD  The RPCBind daemon
QNFSBIOD  The block I/O (BIO) daemon
QNFSNFSD  The NFS server (SVR) daemon
QNFSMNTD  The mount (MNT) daemon
QNFSNSMD  The network status monitor (NSM) daemon
QNFSNLMD  The network lock manager (NLM) daemon
```

Restrictions:

1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
2. The user must be enrolled in the system distribution directory. Use the Add Directory Entry (ADDDIRE) command to enroll the user.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server daemon	*ALL, *RPC, *BIO, *SVR, *MNT, *NSM, *NLM	Required, Positional 1

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
NBRSVR	Number of server daemons	1-20, **1**	Optional, Positional 2
NBRBIO	Number of block I/O daemons	1-20, **1**	Optional
RTVRPCREG	Retrieve RPC registration	***NO**, *YES	Optional
STRJOBTIMO	Timeout for start of daemon	1-3600, **30**, *NOMAX	Optional

Top

Server daemon (SERVER)

Specifies the Network File System (NFS) daemon jobs to be started by this command. The specified daemon should not already be running.

***ALL** All NFS daemons will be started.

***RPC** The NFS RPCBind daemon will be started.

***BIO** Starts NFS block input/output (I/O) daemons. Additional daemons will be started if the number specified on the **Number of block I/O daemons (NBRBIO)** parameter is greater than the number of block I/O daemons already running on the system.

***SVR** Starts NFS server daemons. Additional daemons will be started if the number specified on the **Number of server daemons (NBRSVR)** parameter is greater than the number of server daemons already running on the system.

***MNT** The NFS mount daemon will be started.

***NSM** The NFS network status monitor daemon will be started.

***NLM** The NFS network lock manager daemon will be started.

This is a required parameter.

Top

Number of server daemons (NBRSVR)

Specifies the number of NFS server (*SVR) daemon jobs the user wants to have running. Additional daemons will be started if the number specified on this parameter is greater than the number of server daemons already running on the system. This parameter can only be used if SERVER(*SVR) or SERVER(*ALL) is specified.

1 One NFS server daemon job should be started if there are not already any NFS server daemons running.

1-20 Specify the number of NFS server daemon jobs the user wants to have running.

Top

Number of block I/O daemons (NBRBIO)

Specifies the number of NFS block input/output (I/O) (*BIO) daemon jobs the user wants to have running. Additional daemons will be started if the number specified on this parameter is greater than the number of block I/O daemons already running on the system. This parameter can only be used if SERVER(*BIO) or SERVER(*ALL) is specified.

1 One NFS block I/O daemon job should be started if there are not already any NFS block I/O daemons running.

1-20 Specify the number of NFS block I/O daemon jobs the user wants to have running.

Top

Retrieve RPC registration (RTVRPCREG)

Specifies whether to retrieve previously recorded registration information when the RPCBind daemon is started. If registration information is retrieved, any services already registered with the RPCBind daemon do not have to re-register with the RPCBind daemon. This parameter can only be used if SERVER(*RPC) or (SERVER(*ALL) is specified.

***NO** Do not retrieve registration information.

***YES** Retrieve registration information.

Top

Timeout for start of daemon (STRJOBTIMO)

Specifies the number of seconds to wait for each daemon to successfully start. If a daemon has not started within the timeout value, the command will fail.

30 Default seconds before timeout.

***NOMAX**
Wait forever for daemons to start; do not timeout.

1-3600 Specify a number of seconds to wait for daemons to start before timing out and failing the command. Timeout values less than 30 seconds are rounded up to 30 seconds.

Top

Examples

Example 1: Start All NFS Daemons

```
STRNFSSVR   SERVER(*ALL)  STRJOBTIMO(*NOMAX)
```

This command starts all NFS daemons, and waits forever for them to start. No daemons should be previously running.

Example 2: Start Only One Daemon

```
STRNFSSVR   SERVER(*MNT)
```

This command starts the NFS mount daemon, and waits up to the default of 30 seconds for it to start. The mount daemon should not be already running, and other daemons have been started in the appropriate order.

Top

Error messages

***ESCAPE Messages**

CPFA1B8

*IOSYSCFG authority required to use &1.

Top

Start NetWare Connection (STRNTWCNN)

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

Parameters
Examples
Error messages

The Start NetWare Connection (STRNTWCNN) command starts an authenticated connection to the specified server. This command is used to:

- Start a connection when there is no password in the authentication entry for the server (the password value is *STRNTWCNN).
- Start a connection using authentication information different from that specified in the user profile authentication entry.
- To specify a preferred server for NetWare operations that might otherwise use a different server.

If a password is stored in the authentication entry, this command is not needed; the system will start and end connections as necessary.

Connections can be authorized to be used only by the current job (AUTJOB(*)), or can be authorized to be used by any job running under the specified user profile (AUTJOB(*ANY)), for example, a batch job. Each user profile can only have one connection per job open to a given server at a time.

For NetWare Directory Services trees, connections may be started to each server within the tree. The special value *ALL may be used to start connections to all servers. This could be used, for example, when copying from one server in the tree to another, or copying between partitions of the NDS tree. If multiple connections exist, operations that do not require a specific server will be performed using the first connection started.

The NetWare backup services require that each user perform a separate authentication in addition to a normal login. The connection type (CNNTYPE) parameter allows you to specify that a *SAVRST authentication be performed for this purpose.

Restrictions: To start a connection for another user profile (specified in AUTUSR parameter), you need to have *USE authority to that user profile.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server	*Character value*, *ALL, *ANY	Required, Positional 1
SVRTYPE	Server type	***SERVER**, *NDS	Optional
NDSTREE	NDS tree	*Character value*, ***SERVER**	Optional
AUTUSR	Authorized user profile	*Name*, ***CURRENT**	Optional
NDSCTX	NDS context	*Character value*, ***AUTE**, *USRPRF	Optional
NTWUSER	NetWare user	*Character value*, ***AUTE**, *USRPRF	Optional
PASSWORD	Password	*Character value*, *NONE, *AUTE	Optional
AUTJOB	Authorized job	*, *ANY	Optional
CNNTYPE	Connection type	Single values: *ALL Other values (up to 2 repetitions): ***USER**, *SAVRST	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
CNNIDLTIME	Connection idle time	1-9999, ***NOMAX**	Optional

Top

Server (SERVER)

Specify the server to which the connection is to be started.

***ANY** For *NDS servers, start a connection to any server within the NDS tree.

***ALL** For *NDS servers, start connections to all servers within the NDS tree.

character-value
Specify the name of the server to which the connection is to be started.

Top

Server type (SVRTYPE)

Specifies the type of server.

***SERVER**
Use system information about the specified server to determine the server type. This value cannot be used when *ANY or *ALL is specified for the **Server (SERVER)** parameter.

***NDS** The server is in a NetWare Directory Services tree.

Top

NDS tree (NDSTREE)

For *NDS servers, specifies the name of the NetWare Directory Services (NDS) tree to which the connection is to be started.

***SERVER**
Use system information about the specified server to determine the NDS tree name. This value cannot be used when *ANY or *ALL is specified for the SERVER parameter.

character-value
Specify the name of NDS tree.

Top

Authorized user profile (AUTUSR)

Specifies the user profile authorized to use the connection.

***CURRENT**
The connection can only be used by jobs running under the current user profile.

name Specify the name of the user profile. The connection can only be used by jobs running under the specified user profile.

Top

NDS context (NDSCTX)

Specifies, for *NDS servers, the NetWare Directory Services context in which the NetWare user name is defined.

***AUTE**
Use the authentication entry for this tree in the user profile to determine the NDS directory context.

***USRPRF**
Use the authorized user profile name as the NDS directory context name.

character-value
Specify the name of NDS directory context.

Top

NetWare user (NTWUSER)

Specifies the NetWare user name used to authenticate the user.

***AUTE**
Use the authentication entry for the specified server or NDS tree in the user profile to determine the NetWare user name.

***USRPRF**
Use the authorized user profile name as the NetWare user name.

character-value
Specify the NetWare user name.

Top

Password (PASSWORD)

Specifies the password to be used to verify the authority for the specified user.

***AUTE**
Use the password in the authentication entry for this server or NDS tree in the user profile.

***NONE**
No password is used to verify authority.

character-value
Specify the password to be used.

Top

Authorized job (AUTJOB)

Specifies the job authorized to use the connection. The connection can be used either by the current job, or by any job running under the specified user profile name.

***** The connection can only be used by the current job. The connection will be closed when the job ends.

***ANY** The connection can be used by any job running under the specified user profile. The connection must be ended by an End NetWare Connection (ENDNTWCNN) request.

Notes:

- The QNETWARE file system requires a connection for the current job (AUTJOB(*)).
- If a connection cannot be started using an authentication entry, printing to a NetWare print server requires a connection for any job (AUTJOB(*ANY)).
- Connections for AUTJOB(*) are also ended if the QFPZAUT activation group is reclaimed (RCLACTGRP command).

Top

Connection type (CNNTYPE)

Specifies the types of authentication to be performed. NetWare backup services require a separate authentication in addition to a normal user login.

Single values

***ALL** Both *USER and *SAVRST authentications will be performed.

Other values (up to 2 repetitions)

***USER**
A normal user authentication will be done. This will allow the user to use administration and file system services other than save and restore.

***SAVRST**
Authentication to NetWare backup services will be performed. This option requires that the appropriate NetWare backup NLMs (TSAxxx) be loaded on the server.

Top

Connection idle time (CNNIDLTIME)

Specifies the idle time after which the connection will be closed. The idle time is the time since the most recent request to the server.

***NOMAX**
No idle time limit is specified. The connection will remain open until ended by an End NetWare Connection (ENDNTWCNN) request, or until the job is ended, whichever is sooner.

1-9999 Specify the number of minutes of allowed idle time.

Top

Examples

None

Top

Error messages

***ESCAPE Messages**

FPE0226
Connection not started to server &1.

Top

Start Object Conversion (STROBJCVN)

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

Parameters
Examples
Error messages

The Start Object Conversion (STROBJCVN) command converts user objects from the format used in a previous version, release, and modification level of the operating system to the format required for use in the current operating system. The following objects are converted with this command:

- Programs (original program model (OPM) and Integrated Language Environment (ILE))
- Service programs
- Modules
- Database files (physical and logical) on Replacing-a-release upgrade methods.
- Objects that may contain SQL statements (programs, service programs, or SQL package (*SQLPKG) objects) as well as program and service program objects that are used to implement an external stored procedure. Only the stored SQL statement information is converted to the current operating system format.

The user objects that are not converted with this command are automatically converted when they are first used.

Note: Using objects that are not yet converted will have a one-time performance degradation while the conversion operation is performed.

Restrictions:

- Programs, service programs, and modules must have all observable creation data to be converted. Unobservable creation data cannot be used by this command. Conversion of stored SQL statement information does not require observable creation data.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	*Name*, *ALLUSR, *PTY	Required, Positional 1
OBJTYPE	Object type	***ALL**, *FILE, *ALLPGM, *SQL	Optional, Positional 2
PTY	Priority	1-99, *ALL	Optional, Positional 3

Top

Library (LIB)

Specifies the user library whose objects are to be converted to the format of the current operating system.

This is a required parameter.

© Copyright IBM Corp. 1998, 2006

***ALLUSR**
All user libraries are converted. All libraries with names that do not begin with the letter Q are converted except for the following:

```
#CGULIB     #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 converted:

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

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***PTY** Convert the set of libraries at the priority specified in the PTY parameter. The object conversion priority for a library can be set using the Plan Object Conversion task.

name Specifies the name of the user library whose objects are to be converted.

Note: Although only one library can be specified on this command, repeated calls of this command in a CL program can be used to convert a set of user libraries.

Top

Object type (OBJTYPE)

Specifies which objects in the library should be converted to the format of the current operating system.

***ALL** All *PGM, *SRVPGM, *MODULE, and database files in the specified library are converted to the format of the current operating system. In addition, the stored SQL information in all *PGM, *SRVPGM, and *SQLPKG objects that contain SQL statements, as well as *PGM and *SRVPGM objects that are used to implement external stored procedures, is converted to the format of the current operating system.

***FILE** Only database file member objects in the specified library are converted to the format of the current operating system.

***ALLPGM**
Only *PGM and *SRVPGM objects in the specified library are converted to the format of the current operating system.

***SQL** The stored SQL information in all *PGM, *SRVPGM, and *SQLPKG objects that contain SQL statements, as well as *PGM and *SRVPGM objects that are used to implement external stored procedures, is converted to the format of the current operating system.

Top

Priority (PTY)

Specifies the priority of libraries to be converted to the format of the current operating system. The priority groups are set on the source system during upgrade preparation. This parameter is valid only when *PTY is specified for the **Library (LIB)** parameter.

The priority for a library can be set with the Plan Object Conversion task of the Upgrade Assistant.

Note: You must specify a value for this parameter if *PTY is specified for the LIB parameter.

***ALL** All user libraries are converted to the format of the current operating system. Specifying this value with LIB(*PTY) is the same as specifying LIB(*ALLUSR).

1-99 Specify the priority number of the libraries whose objects are to be converted to the format of the current operating system.

Note: Although only one priority can be specified on this command, repeated calls of this command in a CL program can be used to convert multiple sets of libraries with different priority numbers.

Top

Examples

Example 1: Start Conversion of All Objects

```
STROBJCVN   LIB(LIB1)
```

This command converts all OPM and ILE programs, service programs, modules, and physical and logical database files contained in the LIB1 user library to the format of the current operating system.

In addition, the stored SQL information in all *PGM, *SRVPGM, and *SQLPKG objects that contain SQL statements, as well as *PGM and *SRVPGM objects that are used to implement external stored procedures, is converted to the format of the current operating system.

Example 2: Start Conversion of SQL Objects

```
STROBJCVN   LIB(LIB2)  OPTION(*SQL)
```

This command converts the stored SQL information in all *PGM, *SRVPGM, and *SQLPKG objects in library LIB2 that contain SQL statements, as well as *PGM and *SRVPGM objects in library LIB2 that are used to implement external stored procedures, to the format of the current operating system.

Top

Error messages

***ESCAPE Messages**

CPDA972
Errors occurred while converting object &1 in library &2 type *&3.

CPFA960
Library &1 is not a user library.

CPFA972
Not all eligible objects in library &1 converted.

CPFB0C1
Not all eligible objects in priority &1 converted.

CPF9810

Library &1 not found.

Top

Start Pass-Through (STRPASTHR)

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

Parameters
Examples
Error messages

The Start Pass-Through (STRPASTHR) command allows you to pass through to a target system where you can sign on as if you were attached locally. For information on configuring or operating the pass-through function, refer to the Remote Work Station Support book.

Restriction:

1. This command cannot be entered at a work station with a display that has 12 lines by 80 characters.

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	*Communications name*, *CNNDEV	Required, Positional 1
CNNDEV	APPC device	Single values: ***LOC** Other values (up to 16 repetitions): *Name*	Optional
VRTCTL	Virtual controller	*Name*, ***NONE**	Optional, Positional 2
VRTDEV	Virtual display device	Single values: ***NONE** Other values (up to 32 repetitions): *Name*	Optional, Positional 3
MODE	Mode	*Communications name*, ***NETATR**	Optional
LCLLOCNAME	Local location	*Communications name*, ***LOC**, *NETATR	Optional
RMTNETID	Remote network identifier	*Communications name*, ***LOC**, *NETATR, *NONE	Optional
SRQ10PGM	System request program	Single values: ***SRQMNU** Other values: *Qualified object name*	Optional
	Qualifier 1: System request program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RMTUSER	User profile	*Character value*, ***NONE**, *CURRENT	Optional
RMTPWD	User password	*Character value*, ***NONE**	Optional
RMTINLPGM	Initial program to call	*Name*, ***RMTUSRPRF**, *NONE	Optional
RMTINLMNU	Initial menu	*Name*, ***RMTUSRPRF**, *SIGNOFF	Optional
RMTCURLIB	Current library	*Name*, ***RMTUSRPRF**	Optional
PASTHRSCN	Display option	*Character value*, ***YES**, *NO	Optional

Top

Remote location (RMTLOCNAME)

Specifies one of the following:

- If (*LOC) is specified for the **APPC device** prompt (CNNDEV parameter), the RMTLOCNAME parameter specifies the name of the remote location that is the target of the pass-through session.

© Copyright IBM Corp. 1998, 2006

- If any devices are specified on the CNNDEV parameter, the RMTLOCNAME parameter specifies the first system to do intermediate pass-through routing.

remote-location-name
Specify the remote location name that is the target of the pass-through session or the first system that does intermediate pass-through routing. APPN determines the route to this location.

***CNNDEV**
Use the APPC devices specified with the **APPC device** prompt (CNNDEV parameter).

Top

APPC device (CNNDEV)

Specifies the names of the device descriptions that connect the first system to do pass-through routing with the target system. If *CNNDEV is specified on the Remote location prompt (RMTLOCNAME parameter), the first device specified on this parameter is on the source system. If the name of a remote location is specified on the Remote location prompt (RMTLOCNAME parameter), the first device specified on this parameter is on the system in that remote location. If another system connects the source system to the target system and pass-through must establish the intermediate sessions, then you must specify a list of APPC device descriptions. The APPC device names must be listed in the order that the systems are passed through to get to the target system.

You can enter multiple values for this parameter.

***LOC** The **Remote location** prompt (RMTLOCNAME parameter) specifies the name of the remote location that is the target of the pass-through session.

device-name(s)
Specify the names of the device descriptions that complete the route from the source system to the target system. Up to 16 names can be specified.

Top

Virtual controller (VRTCTL)

Specifies the name of the virtual controller on the remote system that is used to do pass-through jobs. If you specify a virtual controller, one of the virtual display devices attached to it is selected for the pass-through job. A device on the target system is selected based on a comparison of device type and model.

This entry is mutually exclusive with an entry on the **Virtual display device** prompt (VRTDEV parameter); *NONE must be specified when one or more devices are specified on the **Virtual display device** prompt (VRTDEV parameter).

***NONE**
No controller is specified. If *NONE is also specified on the **Virtual display device** prompt (VRTDEV parameter), you are requesting automatic configuration.

virtual-controller-name
Specify the name of the virtual controller description on the target system.

Top

Virtual display device (VRTDEV)

Specifies one or more devices on the target system that are connected to a virtual controller used for the pass-through session. A device on the target system from the list is selected based on a comparison of device type and model.

This entry is mutually exclusive with an entry on the **Virtual controller** prompt (VRTCTL parameter); *NONE must be specified when a controller name is specified on the **Virtual controller** prompt (VRTCTL parameter).

You can enter multiple values for this parameter.

***NONE**
No device names are specified. If *NONE is also specified on the **Virtual controller** prompt (VRTCTL parameter), you are requesting automatic configuration.

virtual-display-device-name
Specify the names of the virtual display device descriptions on the target system. Up to 32 names can be specified.

Top

Mode (MODE)

Specifies the mode name used.

***NETATR**
The mode in the network attributes is used.

mode-name
Specify a mode name.

Top

Local location (LCLLOCNAME)

Specifies the local location name.

***LOC** The local location name is chosen by the system.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

APPN-location-name
Specify the local location name that is associated with the source system. If the local location name is not valid, an escape message is sent.

Top

Remote network identifier (RMTNETID)

Specifies the network ID of the network where the remote location resides.

***LOC** Any remote network ID for the remote location may be used.

***NETATR**
The LCLNETID value specified in the system network attributes is used.

***NONE**
The remote location does not support network identifiers.

APPN-network-identifier
Specify a remote network ID.

Top

System request program (SRQ10PGM)

Specifies that the system request menu is displayed, or specifies the name of the program that starts on the source system when system request option 10 is selected.

The user program can display a menu that allows you to select the system you want to access, and then transfer to a group job that sends the Start Pass-Through (STRPASTHR) command to the desired system. For more information, see the Remote Work Station Support book.

***SRQMNU**
The system request menu on the source system is displayed.

program-name
Specify the program and library names of the program started when system request option 10 is selected.

The possible library values are:

***LIBL** The library list is used to locate the program.

***CURLIB**
The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

library-name
Specify the library where the program is located.

Top

User profile (RMTUSER)

Specifies the user profile for automatic sign-on to the target system. If a profile is specified for this parameter and password security is active on the target system, (*NONE) is not valid for the User password prompt (RMTPWD parameter).

***NONE**
No user profile name is sent, and no automatic sign-on occurs.

***CURRENT**
The user profile of the job using this command is sent. If the target system allows it, and the user profile exists on the target system, and the password specified in the RMTPWD parameter is valid for the profile, the user is automatically signed on. Otherwise, the user is presented with a sign-on display on the target system or a failure message on the source system, depending on the configuration of the target system.

profile-name
Specify a user profile name to use that exists on the target system. If the target system allows it, and if the user profile exists on the target system, the user is automatically signed on. Otherwise, the user is presented with a sign-on display on the target system or a failure message on the

source system, depending on the configuration of the target system. If a profile is specified and password security is active on the target system, a password must be specified, even if the profile specified is the same as the current profile.

Top

User password (RMTPWD)

Specifies the password being sent to the target system.

***NONE**
The system does not send a password. If a profile is specified on the **User profile** prompt (RMTUSER parameter), and password security is active on the target system, this value is not allowed.

password
Specify a password being sent to the target system to verify the sign-on of the user specified in the RMTUSER parameter. This password is encrypted before being sent across the communication line.

Top

Initial program to call (RMTINLPGM)

Specifies the program that is called immediately after you sign on.

***RMTUSRPRF**
The initial program specified in the remote user profile is called immediately after automatic sign-on.

***NONE**
No program is run before the initial menu is shown, even if an initial program is specified in the remote user profile.

program
Specify the name of a program that is called immediately after automatic sign-on.

Top

Initial menu (RMTINLMNU)

Specifies the first menu shown when you are automatically signed on to the target system after the initial program is run.

***RMTUSRPRF**
The initial menu specified in the remote user profile is shown immediately after the initial program is run.

***SIGNOFF**
A menu is not shown after the initial program is run, even if an initial menu is specified in the remote user profile. After the initial program ends, the user is signed off, and the pass-through session ends.

menu Specify the menu that is shown immediately after the initial program is run.

Top

Current library (RMTCURLIB)

Specifies the name of the library that becomes the current library in the library list of the job after automatic sign-on.

***RMTUSRPRF**
The current library specified in the remote user profile becomes the current library in the library list after automatic sign-on.

library
Specify the library that becomes the current library in the library list after automatic sign-on.

Top

Display option (PASTHRSCN)

Specifies whether the pass-through display and associated status messages appear before the pass-through session is established.

***YES** The pass-through display and information messages are shown before the pass-through session is established.

***NO** The pass-through display and information messages are not shown before the pass-through session is established.

Top

Examples

Example 1: Pass-Through to Toronto

```
STRPASTHR   RMTLOCNAME(*CNNDEV)  CNNDEV(DET CHI TOR)  VRTCTL(VWSC)
```

This command specifies starting a pass-through to the Toronto system by going through Detroit and Chicago. More information is in the **Remote Work Station Support** book located in the Information Center.

Example 2: Pass-Through to Detroit

```
STRPASTHR   RMTLOCNAME(DETROIT)  VRTCTL(VWSC)
```

This command specifies a pass-through to the Detroit system. APPN establishes the route to Detroit.

Example 3: Pass-Through to Toronto

```
STRPASTHR   RMTLOCNAME(DETROIT)  CNNDEV(CHI TOR)  VRTCTL(VWSC)
```

This command specifies another way to pass-through to the Toronto system by going through Chicago and Detroit. APPN establishes the route to Detroit.

Example 4: Pass-Through to Detroit

```
STRPASTHR   RMTLOCNAME(DETROIT)  RMTUSER(*CURRENT)
```

This command specifies a pass-through to the DETROIT system and an automatic sign-on using the user profile with the same name as the one currently used on the source system. It also specifies that the DETROIT system automatically configures a virtual device for the pass-through session, since a virtual controller or virtual device was not specified.

Top

Error messages

***ESCAPE Messages**

CPF2702
Device description &1 not found.

CPF2703
Controller description &1 not found.

CPF5383
Mode &7 specified for device &4 not valid.

CPF5536
System cannot automatically select an APPC device description for the remote location.

CPF5546
Class-of-service for device &4 not valid.

CPF8901
Virtual device &1 not varied on.

CPF8902
Virtual device &1 not available.

CPF8903
Device &1 not valid for pass-through.

CPF8904
Pass-through request not accepted.

CPF8905
Pass-through not allowed on this system.

CPF8906
Error during session initialization. Reason code &1.

CPF8907
Communications failure for device &1.

CPF8908
Controller &1 not varied on.

CPF8909
Old software release. Pass-through ended.

CPF8910
Controller &1 not valid for pass-through.

CPF8911
Communications failure. Session was not started.

CPF8912
Pass-through session ended. Reason code &1.

CPF8913
Pass-through ended abnormally.

CPF8916
Cannot select virtual device &1 at system &2.

CPF8917
Not authorized to &1.

CPF8918
Job canceled at system &1.

CPF8919
Device &1 not accessed by system &2.

CPF8920
Pass-through failed. &1 must be varied off and on.

CPF8921
APPC failure. Failure code is &3.

CPF8922
Negative response from device &1 at system &2.

CPF8923
Data stream received at system &1 not valid.

CPF8924
No available virtual controller.

CPF8925
Device &1 not created.

CPF8928
Device &1 could not be changed.

CPF8929
Device &1 could not be varied on.

CPF8931
Location &1 not an APPC location.

CPF8932
Device &1 must be non-networking APPC device.

CPF8933
Route to specified location not found.

CPF8935
Pass-through not allowed to system &1.

CPF8936
Pass-through failed for security reasons.

CPF8937
Automatic sign on not allowed.

CPF8938
Error in QRMTSIGN program. Pass-through failed.

CPF8939
Trying to send too much data.

CPF8940
Cannot automatically select virtual device.

CPF8941
Incorrect internal use of pass-through.

CPF8943
Pass-through not allowed from server TELNET session.

***STATUS Messages**

CPI8901
No matching device on remote system. Function limited.

CPI8902
Pass-through started at system &1.

CPI8903
Virtual device &1 selected at system &2.

CPI8906
Automatic sign-on not allowed.

Top

Start PC Command (STRPCCMD)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The Start PC Command (STRPCCMD) command allows you to run a single application, a DOS command, or an OS/2 command on an attached personal computer.

Note: Do not precede an entry with an asterisk unless that entry is a ″special value″ that is shown (on the display itself or in the help information) with an asterisk.

Error messages for STRPCCMD

None

Top

Parameters

Keyword	Description	Choices	Notes
PCCMD	PC command	*Character value*	Required, Positional 1
PAUSE	Pause	***YES**, *NO	Optional

Top

PC command (PCCMD)

Specifies the valid DOS command, OS/2 command, or personal computer application you want to run. If you are using the STRPCCMD command on a server command line and the command or application name contains special characters, you must enclose the command in apostrophes. Special characters include blanks, commas, and colons.

If you select the option to start a PC command from the iSeries Access Organizer menu, you do not need to enclose the command in apostrophes unless it ends in a colon. The ending colon is ignored unless the entire PC command is enclosed in apostrophes.

This is a required parameter.

Top

Pause (PAUSE)

Specifies whether the computer should pause after running a command.

The possible values are:

© Copyright IBM Corp. 1998, 2006

***YES** The computer will pause after running the command before returning to the iSeries Access Organizer menu.

***NO** The computer returns immediately to the iSeries Access Organizer menu after the command runs.

Top

Examples

None

Top

Error messages

None

Top

Start iSeries Access Organizer (STRPCO)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The Start iSeries Access Organizer (STRPCO) command starts the iSeries Access Organizer on the host system.

Note: Do not precede an entry with an asterisk unless that entry is a ″special value″ that is shown (on the display itself or in the help information) with an asterisk.

Error messages for STRPCO

None

Top

Parameters

Keyword	Description	Choices	Notes
PCTA	Text Assist	*Character value*, ***YES**, *NO	Optional, Positional 1

Top

Text Assist (PCTA)

Specifies whether you are going to use the Personal Computer Text Assist.

The possible values are:

***YES** The Personal Computer Text Assist is going to be used.

***NO** The Personal Computer Text Assist is not going to be used.

Top

Examples

None

Top

Error messages

None

© Copyright IBM Corp. 1998, 2006

Top

Start Performance Explorer (STRPEX)

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

Parameters
Examples
Error messages

The Start Performance Explorer (STRPEX) command starts a new Performance Explorer session or resumes a suspended Performance Explorer session.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
3. The following user profiles have private authorities to use the command:
   - QPGMR
   - QSRV
4. Two threads within the same job will not be allowed to run STRPEX at the same time. The thread that issued STRPEX first will run the command to completion while the second STRPEX waits.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	*Name*	Required, Key, Positional 1
OPTION	Option	***NEW**, *INZONLY, *RESUME	Optional, Positional 2
DFN	Definition	*Name*, ***SELECT**	Optional
FTR	Filter	*Name*, ***NONE**, *SELECT	Optional

Top

Session ID (SSNID)

Specifies an identifier for this Performance Explorer session. This name must be unique within the active sessions of the Performance Explorer tool.

This is a required parameter.

session-identifier
Specify the session identifier for a new or suspended Performance Explorer session.

Top

© Copyright IBM Corp. 1998, 2006

Option (OPTION)

Specifies whether a new session should be started or a session which is currently suspended should be resumed.

This is a required parameter.

***NEW** A new session of the Performance Explorer tool is started. Performance data collection begins as soon as the session has been established.

***INZONLY**
A new Performance Explorer session is started, but once the session is established, it is suspended. This option allows the user to perform the setup for a particular session before the scenario to be monitored is started. To begin data collection, the user must invoke this command again specifying the same session identifier and OPTION(*RESUME).

***RESUME**
A suspended Performance Explorer session is resumed. The session was suspended either by specifying OPTION(*SUSPEND) on the ENDPEX command or by specifying OPTION(*INZONLY) on a previous STRPEX command.

Top

Definition (DFN)

Specifies the name of the Performance Explorer definition to use when starting a new Performance Explorer session. The definition identifies the performance data to be collected. This parameter is required in order to start a new Performance Explorer session. This parameter cannot be specified when resuming a suspended session, since the suspended session already has an associated Performance Explorer definition.

definition-name
Specify the name of the Performance Explorer definition.

Top

Filter (FTR)

Specifies the name of the Performance Explorer filter to use when starting a new Performance Explorer session. The filter determines which events are collected based on the filter values which are compared to the event's actual data. If a filter is not specified, then all events in the definition are collected. This parameter cannot be specified when resuming a suspended session.

***SELECT**
A list of existing filters will be displayed for user selection. This value is valid only if the command is run in an interactive job.

filter-name
Specify the name of the Performance Explorer filter.

Top

Examples

Example 1: Start a New Session

```
STRPEX   SSNID(TESTRUN2)  DFN(NEWDESC)  OPTION(*NEW)
         FTR(MYFILTER)
```

This command starts a new session of the performance explorer using the criteria identified in a definition named NEWDESC and a filter named MYFILTER. The new session name is TESTRUN2.

Example 2: Resume a Suspended Session

```
STRPEX   SSNID(TESTRUN1)  OPTION(*RESUME)
```

This command resumes an already existing session of the performance explorer named TESTRUN1.

Top

Error messages

Error messages for RMVPEXDFN

***ESCAPE Messages**

CPFAF05
STRPEX command was not successful. Reason code is &1. See details for more information.

CPFAF11
Unable to locate program, object, library, or file member as specified in the definition or filter.

Top

Start Performance Collection (STRPFRCOL)

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

Parameters
Examples
Error messages

The Start Performance Collection (STRPFRCOL) command starts the system-level collection of performance data by Collection Services. The properties of the system-level collection are controlled by the Collection Services configuration, which can be changed using the Configure Performance Collection (CFGPFRCOL) command. The data included in the system-level collection is determined by the value specified on the Collection profile parameter.

Performance data collection is conducted by the Collection Services server job (QYPSPFRCOL). If this job is not active, it will be started as a result of this command. If the job is active, any change to the collection profile will take effect immediately and the collection will continue uninterrupted.

QYPSPFRCOL creates a management collection object (*MGTCOL) to store performance data. Data collected will include both the system-level collection profile categories as well as data collected on behalf of client applications (for example, iSeries Navigator monitors or Performance Collector APIs). All data is stored in the current management collection object. This data may be processed using the Create Performance Data (CRTPFRDTA) command to generate the performance database files.

Cycling the performance collection will cause the Collection Services server job to create a new management collection object prior to implementing the specified Collection profile. Note: cycling may result in the loss of one interval of data and will also impact any client applications that are using Collection Services.

Top

Parameters

Keyword	Description	Choices	Notes
COLPRF	Collection profile	***CFG**, *MINIMUM, *STANDARD, *STANDARDP, *ENHCPCPLN, *CUSTOM	Optional, Positional 1
CYCCOL	Cycle collection	***NO**, *YES	Optional, Positional 2

Top

Default collection profile (COLPRF)

Specifies the collection profile. This will determine which categories will be included in the system-level collection. Specifying a profile for this parameter will change the default collection profile in the Collection Services configuration. The default collection profile may also be changed using the Configure Performance Collection (CFGPFRCOL) command.

***CFG** The collection will use the currently configured value. If the configured value has never been modified, the collection profile will be *STANDARDP.

***MINIMUM**
The minimum data collection recommended. Includes the following categories: *SYSBUS, *POOL, *HDWCFG, *SYSCPU, *SYSLVL, *JOBMI, *JOBOS, *DISK, and *IOPBASE.

© Copyright IBM Corp. 1998, 2006

***STANDARD**
The standard profile includes all categories which are typically needed by the Performance Tools for iSeries, with the exception of communications data. It includes all categories in the *MINIMUM profile, as well as the following categories: *POOLTUNE, *SUBSYSTEM, *SNADS, *LCLRSP, *APPN, *SNA, *TCPBASE, *USRTNS, and *LPAR. The category *DOMINO will be included if the product Domino for iSeries has been installed on the system. The category *HTTP will be included if the product IBM HTTP Server for iSeries (powered by Apache) has been installed on the system.

***STANDARDP**
The standard plus profile includes all categories which are typically needed by the Performance Tools for iSeries, including communications data. It includes all categories in the *STANDARD profile, as well as the following categories: *IPCS, *CMNBASE, *CMNSTN, *CMNSAP, *TCPIFC, and *DPS.

***ENHCPCPLN**
The enhanced capacity planner profile includes all categories in the *STANDARDP profile, with the addition of *INTPEX.

***CUSTOM**
The custom profile includes categories as defined by the user. This profile must be defined using the Collection Services function in iSeries Navigator.

Top

Cycle collection (CYCCOL)

Specifies whether the collection should be cycled. Cycling the collection will cause data to be collected in a new management collection object (*MGTCOL).

***NO** Data will continue to be collected in the current collection object.

***YES** Data will be collected in a new collection object.

Top

Examples

Example 1: Starting the Performance Collection

```
STRPFRCOL
```

This command will cause the Collection Services server job (QYPSPFRCOL) to start the system-level collection of performance data using the currently configured default collection profile.

Example 2: Starting with Collection Profile and Cycle Collection

```
STRPFRCOL COLPRF(*MINIMUM) CYCCOL(*YES)
```

This command will cause the Collection Services server job (QYPSPFRCOL) to cycle and begin collecting performance data in a new management collection object (*MGTCOL) using the *MINIMUM collection profile. Note: this command will also change the configured default collection profile to *MINIMUM.

Top

Error messages

***ESCAPE Messages**

CPF3CF2
Error(s) occurred during running of &1 API.

CPFB94A
Collector communications error. Reason code &1.

Top

Start Performance Trace (STRPFRTRC)

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

Parameters
Examples
Error messages

The Start Performance Trace (STRPFRTRC) command is a simplified interface to the TRCINT command for collecting Multiprogramming level (MPL) and Transaction trace data. This command collects the same performance trace data as was collected in previous releases by the Start Performance Monitor (STRPFRMON) command.

The trace started by this command creates and uses trace table QPM_STRPFRTRC.If the trace table exists, any existing data will be deleted before this trace begins.

The trace can be stopped and the data can be written to a data base file by using the End Performance Trace (ENDPFRTRC) command.

Restrictions:

- This command is shipped with public *EXCLUDE authority.
- The following user profiles have private authorities to use the command:
  - QSRV

Top

Parameters

Keyword	Description	Choices	Notes
SIZE	Trace table size	Single values: ***CALC**, *MAX Other values: *Element list*	Optional, Positional 1
	Element 1: Number of units	1-998000	
	Element 2: Unit of measure	***KB**, *MB	
OMTTRCPNT	Omit trace points	***NONE**, *RSCMGT	Optional
JOBTYPE	Job types	Single values: *NONE, *ALL Other values (up to 12 repetitions): ***DFT**, *ASJ, *BCH, *EVK, *INT, *MRT, *RDR, *SBS, *SYS, *WTR, *PDJ, *PJ, *BCI	Optional
JOBTRCITV	Job trace interval	0.1-9.9, **0.5**	Optional

Top

Trace table size (SIZE)

Specifies the size of the trace table.

Note: The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). Refer to the Trace Internal (TRCINT) command SIZE parameter for additional information regarding the setting of trace table size.

Specifying a size of less than 16 megabytes is not recommended.

© Copyright IBM Corp. 1998, 2006

Single values

***CALC**
The minimum trace table size is determined based on the processor group of your system.

***MAX** The trace table is set to the maximum size of 258048 megabytes.

Element 1: Number of units

Specify the size of the trace table.

1-998000
Specify the size of the trace table in kilobytes or megabytes.

Element 2: Unit of measure

Specify whether the value specified for the first element of this parameter should be treated as number of kilobytes or number of megabytes.

***KB** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

***MB** The trace table size is specified in megabytes. The valid range is 1 through 258048.

Top

Omit trace points (OMTTRCPNT)

Specifies trace points whose trace records are to be excluded.

***NONE**
No trace points are to be excluded.

***RSCMGT**
Resource management trace points (seize/lock conflict data) will be excluded from the trace.

Top

Job types (JOBTYPE)

Specifies the types of jobs for which trace data is to be collected for use in the batch job trace report.

Note: The value *DFT includes the values *ASJ, *BCH, *EVK, *MRT, *PDJ, *PJ and *BCI. The value *BCH includes the values *EVK, *MRT, *PDJ, *PJ, and *BCI.

Single values

***NONE**
No jobs are to be traced.

***ALL** All job types are to be traced.

Other values (up to 12 repetitions)

***DFT** Batch and autostart jobs are traced.

***ASJ** Autostart jobs are traced.

***BCH** Batch jobs are traced.

***EVK** Jobs started by a procedure start request are traced.

***INT** Interactive jobs are traced.

***MRT** Multiple requester terminal jobs are traced.

***RDR** Reader jobs are traced.

***SBS** Subsystem monitor jobs are traced.

***SYS** System jobs are traced.

***WRT** Writer jobs are traced.

***PDJ** Print driver jobs are traced.

***PJ** Prestart jobs are traced.

***BCI** Batch immediate jobs are traced.

Top

Job trace interval (JOBTRCITV)

Specifies the time interval (in CPU seconds) to be used between each collection of the job trace data.

0.5 A time slice interval value of 0.5 CPU seconds is used.

0.1-9.9 Specify the trace interval to be used, in CPU seconds.

Top

Examples

```
STRPFRTRC   SIZE(*CALC)
```

This command starts the collection of performance trace data. The trace table size may be adjusted to the calculated minimum. This example will result in the same trace table size and data as would STRPFRMON TRACE(*ALL) DMPTRC(*NO).

Top

Error messages

***ESCAPE Messages**

CPF0A2A
Performance trace already started

Also refer to the TRCINT command for other messages.

Top

Start Program Export List (STRPGMEXP)

Parameters
Examples
Error messages

The Start Program Export List (STRPGMEXP) binder definition statement starts a list of exports in a service program export block.

Top

Parameters

Keyword	Description	Choices	Notes
PGMLVL	Program level	***CURRENT**, *PRV	Optional, Positional 1
LVLCHK	Signature level check	***YES**, *NO	Optional, Positional 2
SIGNATURE	Signature	*Character value*, ***GEN**	Optional, Positional 3

Top

Program level (PGMLVL)

Specifies the service program level for this export block.

***CURRENT**
This service program export block contains exports for the current service program.

***PRV** This service program export block contains exports for a previous version of the service program.

Top

Signature level check (LVLCHK)

Specifies whether a level check is performed on the export block.

***YES** The service program export block is level checked by generating a nonzero signature.

***NO** The service program export block is not level checked. A zero signature is generated.

Top

Signature (SIGNATURE)

Specifies the signature for the export block.

***GEN** If *YES is specified for the **Signature level check (LVLCHK)** parameter, the system generates a nonzero signature for the export block. If *NO is specified for the LVLCHK parameter, this value is required and the system generates a zero signature for the export block.

© Copyright IBM Corp. 1998, 2006

hexadecimal-character-value

The signature value for the export block is set to the specified string of hexadecimal digits and is not generated by the system. If the specified value is less than 32 hexadecimal digits in length, the system pads it on the left with hexadecimal zeros to 32 digits. If the specified value is greater than 32 hexadecimal digits in length, the system truncates it on the right to 32 digits.

character-value

The signature value for the export block is set to the EBCDIC character codes of the given signature-value and is not generated by the system. If the signature-value is less than 16 characters in length, the system pads the signature-value on the right with spaces to 16 characters. If the signature-value is greater than 16 characters in length, the system truncates it on the right to 16 characters.

Top

Examples

```
STRPGMEXP   PGMLVL(*CURRENT)  LVLCHK(*YES)  SIGNATURE(*GEN)
```

This binder definition statement marks the beginning of a list of exported variables or procedures for a service program. This service program export block contains exports for the current service program. Level checking will be performed using a signature generated by the operating system.

Top

Error messages

None

Top

Start Programmer Menu (STRPGMMNU)

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

Parameters
Examples
Error messages

The Start Programmer Menu (STRPGMMNU) command shows the programmer menu. This command can be used instead of the CALL QPGMMENU function, and allows you to pass parameters to specify and control the data which appears in the associated fields on the programmer menu.

NOTES:

1. A user exit program can be called instead of submitting a job when option 3 is selected.
2. The first four parameters control the defaults that appear when the menu is first displayed.

More information about using the Programmer Menu is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Top

Parameters

Keyword	Description	Choices	Notes
SRCFILE	Source file	*Name*, ***DFT**	Optional, Positional 1
SRCLIB	Source library	*Name*, ***LIBL**, *CURLIB	Optional, Positional 2
OBJLIB	Object library	*Name*, ***DFT**, *CURLIB	Optional, Positional 3
JOBD	Job description	*Name*, ***USRPRF**	Optional, Positional 4
ALWUSRCHG	Allow changes	***YES**, *NO	Optional
EXITPGM	Option 3 exit program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Option 3 exit program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DLTOPT	Delete option	***DLT**, *PROMPT, *NODLT	Optional

Top

Source file (SRCFILE)

Specifies an existing source file that contains source file members to be updated or to which new source file members are to be added.

***DFT** This is the default for the type being specified on the menu. This field is blank when shown on the display station.

name Specify the name of the source file to be updated.

Top

© Copyright IBM Corp. 1998, 2006

Source library (SRCLIB)

Specifies the library that is searched for the source file.

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

***CURLIB**
The current library for the job is used to locate the source file. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the source file is located.

Top

Object library (OBJLIB)

Specifies the library that is to contain the object.

***DFT** Blanks appear for this field. The library used depends on the menu option you selected.

***CURLIB**
The current library for the job is the library that is to contain the object.

name Specify the name of the library that is to contain the created object.

Top

Job description (JOBD)

Specifies the job description used with the job being submitted.

***USRPRF**
The job description defined in the user profile of the user running the STRPGMMNU command is used for the job.

name Specify the name of the job description used for the job. The job description is found through the library list being used by the job.

Top

Allow changes (ALWUSRCHG)

Specifies whether the menu display fields you specified on the previous parameters in this command can be changed by the user.

***YES** Values on the display can be changed.

***NO** The display fields cannot be changed.

Top

Option 3 exit program (EXITPGM)

Specifies the user-written program that is called as an exit program in place of submitting a batch job when menu option 3 is selected. When the exit program is called, it receives parameters that are sent by the programmer menu. More information about the EXITPGM parameter is in the CL information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Single values

***NONE**
No user-written program is called; a batch job is submitted. When *NONE is specified, *DLT must be specified for the **Delete option (DLTOPT)** parameter.

Qualifier 1: Option 3 exit program

name Specify the name of the program called when option 3 is selected, instead of submitting the create command as a batch job. When a value is specified on this parameter, the text that appears on the menu for option 3 shows the name and library of the exit program.

Qualifier 2: Library

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

***CURLIB**
The current library for the job is used to locate the program. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the program is located.

Top

Delete option (DLTOPT)

Specifies the action to be taken when:

- A program name is specified on the **Option 3 exit program (EXITPGM)** parameter.
- Option 3 of the programmer menu is selected.
- An object of the name and type to be created already exists in the library specified on the menu.

Regardless of the value specified, the system passes a parameter (from among the parameters passed from the Programmer Menu) to the exit program that specifies whether the object exists.

***DLT** This value must be specified if *NONE is specified on the EXITPGM parameter. If an exit program is specified for the EXITPGM parameter, and the object specified to be created with option 3 exists, and the Enter key is pressed, a message is shown; press the F11 key to proceed. When the F11 key is pressed, the system deletes or replaces the object before calling the program specified by the exit program. This is the normal Programmer Menu function when an exit program is not specified.

NOTES:

1. When the *DLT value is specified, the object is deleted or replaced before the job is submitted or the user exit program is called.
2. If the source type is one of the following, the object is replaced rather than deleted (an exit program must be used to delete instead of replace):

```
BAS       C         CBL
CBL36     CLP       DSPF
DSPF36    FTN       ICFF
MNU36     MSGF36    PAS
PLI       PRTF      RPG
RPG36     RPT36
```

***PROMPT**
The system does not delete or replace the object, but you are prompted for approval to delete the object. If the object exists, and the Enter key is pressed, a message is displayed. Press the F11 key to proceed; the system does not delete the object. The user confirms whether the object is deleted or replaced, yet the deletion is still controlled by the exit program.

***NODLT**

The user exit program is called regardless of the presence of the object.

Table 1. **Figure: Table 1. Actions Taken by the System When an Exit Program Is Called**

DLTOBJ Specified	F11 Required	Object Deleted	Value Passed to Exit Program
If object existed when option 3 was selected			
*DLT	Yes	Yes	1
*PROMPT	Yes	No	0
*NODLT	No	No	0
If object did not exist when option 3 was selected			
*DLT	No	---	2
*PROMPT	No	---	2
*NODLT	No	---	2

Top

Examples

Example 1: Displaying Programmer Menu

```
STRPGMMNU
```

This command displays the Programmer Menu with defaults for all parameters. This has the same result as entering CALL QPGMMENU.

Example 2: Preventing Values from Being Changed

```
STRPGMMNU   SRCFILE(YOURFILE)  SRCLIB(YOURLIB)  OBJLIB(YOURLIB)
            JOB(YOURJOBD)  ALWUSRCHG(*NO)
```

This command prevents the values on the menu from being changed from those specified on the command.

Example 3: Calling an Exit Program

```
STRPGMMNU   EXITPGM(OPT3PGM)  DLTOPT(*PROMPT)
```

This command calls user exit program OPT3PGM instead of submitting a batch job when option 3 is specified. If the object already exists, DLTOPT(*PROMPT) requires the user to press the F11 key; however, the object is not deleted.

Example 4: Receiving Parameters

The following portion of a CL program is an example of how these parameters would be received by a user exit program. If the specified type is one of those listed, the object is not deleted. The create command with REPLACE(*YES) specified is passed to the exit program. The value passed to the exit program is 0.

```
PGM   PARM(&OPTION &PARM &TYPE &PARM2 &SRCFILE  +
           &SRCLIB &OBJLIB &JOBD &RQSLEN &RQSDTA512  +
           &F4 &F11 &EXIST)
/* The following values are passed in exactly as  */
/* they appear on the Programmer Menu.            */
DCL   VAR(&OPTION)   TYPE(*CHAR)  LEN(2)
DCL   VAR(&PARM)     TYPE(*CHAR)  LEN(10)
DCL   VAR(&TYPE)     TYPE(*CHAR)  LEN(10)
DCL   VAR(&PARM2)    TYPE(*CHAR)  LEN(21)
DCL   VAR(&SRCFILE)  TYPE(*CHAR)  LEN(10)
DCL   VAR(&SRCLIB)   TYPE(*CHAR)  LEN(10)
DCL   VAR(&OBJLIB)   TYPE(*CHAR)  LEN(10)
DCL   VAR(&JOBD)     TYPE(*CHAR)  LEN(10)
/* The following values are derived by QPGMMENU     */
/* from the information entered to the above fields */
/* and the F keys.                                  */
/* NUMBER OF BYTES OF REQUEST DATA */
DCL   VAR(&RQSLEN)  TYPE(*DEC)  LEN(3 0)
/* DATA FOR RRQSDTA PARAMETER OF SBMJOB COMMAND. */
DCL   VAR(&RQSDTA512)  TYPE(*CHAR)  LEN(512)
/* F4 WAS PRESSED, '1', OTHERWISE '0'.   */
DCL   VAR(&F4)  TYPE(*CHAR)  LEN(1)
/* F11 WAS PRESSED, '1', OTHERWISE '0'.     */
DCL   VAR(&F11)  TYPE(*CHAR)  LEN(1)
/* OBJECT EXISTS- '0 OBJECT WAS DELETED- '1'
OR OBJECT DID NOT EXIST -'2'*/
DCL   VAR(&EXIST)  TYPE(*CHAR)  LEN(1)
```

Additional information, along with examples of the STRPGMMNU command with the EXITPGM parameter, can be found in the **CL Programming** book in the Information Center.

Top

Error messages

None

Top

Start Program Profiling (STRPGMPRF)

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

Parameters
Examples
Error messages

The Start Program Profiling (STRPGMPRF) command starts collection of profiling information of ILE programs or service programs that have been enabled to collect profiling data by specifying PRFDTA(*COL) on the Change Program (CHGPGM) or Change Service Program (CHGSRVPGM) command, or when the modules were created by a compiler, or by specifying PRFDTA(*COL) on the Change Module (CHGMOD) CL command. All programs compiled or changed with this option that are active will have profiling information updated until an End Program Profiling (ENDPGMPRF) command is issued.

The profiling information is added to existing profiling information. If this is not desired, the profiling data can be cleared by specifying PRFDTA(*CLR) on the CHGPGM or CHGSRVPGM command.

Restrictions:

- This command is shipped with no public (*EXCLUDE) authority, and QPGMR user profile having use (*USE) authority to the command.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
STRPGMPRF
```

This command start the collection of program profiling information.

Top

Error messages

***ESCAPE Messages**

CPF5CAA
Unexpected error occurred during program profiling.

Top

© Copyright IBM Corp. 1998, 2006

Start Prestart Jobs (STRPJ)

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

Parameters
Examples
Error messages

The Start Prestart Jobs (STRPJ) command starts jobs for a prestart job entry in an active subsystem when there are no currently active prestart jobs for the prestart job entry.

This command is valid after an ENDPJ command is complete, or when all prestart jobs have been ended by the system due to an error or were never started during subsystem start up due to STRJOBS (*NO) on the ADDPJE command. The number of jobs started is determined by the INLJOBS value on the prestart job entry.

Restrictions:

1. To use this command, you must have:
   - job control (*JOBCTL) special authority.
   - use (*USE) authority to the subsystem description.
   - execute (*EXECUTE) authority to the library that contains the program.

Top

Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	*Name*	Required, Positional 1
PGM	Program	*Qualified object name*	Required, Positional 2
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

Top

Subsystem (SBS)

Specifies the name of the active subsystem that contains the prestart job entry.

This is a required parameter.

name Specify the name of the active subsystem that contains the prestart job entry.

Top

© Copyright IBM Corp. 1998, 2006

Program (PGM)

Specifies the name of the program that identifies the prestart job entry. This program name is used to match an incoming request.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program that identifies the prestart job entry.

Qualifier 2: Library

***LIBL** All libraries in the thread's library list are searched until a match is found.

***CURLIB**
The current library for the thread is used to locate the object. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the program's library.

Top

Examples

```
STRPJ   SBS(SBS1)  PGM(PJLIB/PJPGM)
```

This command starts prestart jobs for prestart job entry PJPGM in subsystem SBS1. Subsystem SBS1 must be active when this command is issued. The number of jobs started is the number specified in the INLJOBS value of prestart job entry PJPGM. The subsystem starts program PJPGM in library PJLIB.

Top

Error messages

***ESCAPE Messages**

CPF0921
Start Prestart Jobs command not allowed now.

CPF1226
Start prestart jobs failed.

CPF1227
No authority has been granted to use command.

CPF1317
No response from subsystem for job &3/&2/&1.

CPF1351
Function check occurred in subsystem for job &3/&2/&1.

CPF1834
Prestart job entry for program &1 in &2 does not exist.

CPF1835
Not authorized to subsystem description.

CPF9810
Library &1 not found.

CPF9811
Program &1 in library &2 not found.

CPF9820
Not authorized to use library &1.

CPF9821
Not authorized to program &1 in library &2.

Top

Start Printer Emulation (STRPRTEML)

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

Parameters
Examples
Error messages

The Start Printer Emulation (STRPRTEML) command starts 3270 printer emulation using a binary synchronous communications (BSC) or Systems Network Architecture (SNA) emulation printer device and a printer device file. The STRPRTEML command is used to print host system (System/370 type) information on an iSeries system. It is used when the user is working on an iSeries system and the information is on a System/370 type system.

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

Top

Parameters

Keyword	Description	Choices	Notes
EMLCTL	Emulation controller, or	*Name*	Optional, Positional 1
EMLDEV	Emulation device, or	*Name*	Optional, Positional 2
EMLLOC	Emulation location	*Communications name*	Optional, Positional 3
PRTDEV	Print device	*Name*	Optional
JOB	Job name	*Name*, ***EMLDEV**	Optional
ENDBKTEJT	End Bracket eject	***NO**, *YES	Optional
PRTFILE	Printer file	*Qualified object name*	Optional
	Qualifier 1: Printer file	*Name*, **QPEMPRTF**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TIMOUTEJT	Timeout wait eject	Single values: ***NONE**, *IMMED Other values: *Element list*	Optional
	Element 1: Minutes	0-99	
	Element 2: Seconds	0-59, **0**	
DFRPRTOUT	Defer printing spool output	***PRTFILE**, *YES, *NO	Optional
SPOOL	Spool output	***PRTFILE**, *YES, *NO	Optional
OPNPRTF	Open printer file	***IMMED**, *RCVDTA	Optional
NUMCOL	Print positions per line	1-378, ***PRTFILE**	Optional
NUMLIN	Lines per page	1-255, ***PRTFILE**	Optional
LPI	Lines per inch	***PRTFILE**, 6, 3, 4, 7.5, 7,5, 8, 9	Optional
MSGQ	Message queue	Single values: ***DSPDEV**, *NONE Other values: *Qualified object name*	Optional
	Qualifier 1: Message queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
CHRSET	Language character set	***SYSVAL**, AGB, AGE, AGI, AGM, ALI, ALM, BGB, BGE, BLI, BLM, BRB, BRE, CAB, CAE, CAI, CAM, CLB, CLE, CYB, CSB, CSE, DMB, DMI, DME, DMM, ESB, FAB, FAI, FAE, FAM, FNB, FNI, FQB, FQI, FNE, FNM, ICE, GKB, GNB, GNE, HIB, HNB, HNE, ICB, ICI, ICM, INB, INI, IRB, ITB, ITE, ITI, ITM, JEB, JEI, JKB, JPB, JPE, JUB, KAB, KOB, LTB, LAE, LVB, MKB, MKE, NCB, NCE, NEB, NEI, NEE, NEM, NWB, NWE, NWI, NWM, PKE, PLB, PLE, PRB, PRI, PRE, PRM, RCB, RMB, RME, ROB, ROE, RUB, RUE, SFI, SFM, SGM, SGI, SKB, SKE, SPB, SPE, SPI, SPM, SQB, SQE, SSB, SSI, SWB, SWI, SSE, SSM, SWE, SWM, TAB, THB, THE, TKB, TKE, TRB, TRE, UAE, UKB, UKI, USB, USI, UKE, UKM, USE, USM, VNE, YGI, YGM, *TRNTBL	Optional
SBMJOB	Submit job	***YES**, *NO	Optional
JOBD	Job description	*Qualified object name*	Optional
	Qualifier 1: Job description	*Name*, **QBATCH**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TRNTBLOUT	Outgoing translation table	Single values: ***CHRSET** Other values: *Qualified object name*	Optional
	Qualifier 1: Outgoing translation table	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
JOBQ	Job queue	Single values: ***JOBD** Other values: *Qualified object name*	Optional
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ENDCOND	End emulation conditions	Single values: ***NONE** Other values (up to 3 repetitions): *DACTLU, *ENDBKT, *UNBIND	Optional
FORMFEED	Acknowledge form feed	***YES**, *NO	Optional
EMLCFGE	Configuration entry	*Name*, QEMDFTCFGE, ***NONE**	Optional

Top

Emulation controller (EMLCTL)

Specifies the name of a BSC controller description or SNA controller description that has attached 3270 printer emulation device descriptions. When this parameter is specified, the printer emulation job uses a 3270 printer emulation device attached to this controller description. The requester must be authorized to the controller and at least one device, and the device must be available.

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

Top

Emulation device (EMLDEV)

Specifies the name of a BSC or an SNA printer emulation device (EMLDEV(3284, 3286, 3287, 3288, or 3289)) that is used by the printer emulation job to do a type 3270 printer emulation. The user must be authorized to the device, and the device must be available.

Top

Emulation location (EMLLOC)

Specifies the emulation remote location name that describes the location of the 3270 printer emulation devices. This name is defined during device description configuration and refers to the remote location where communication takes place. When this parameter is specified, an available printer emulation device is selected from those referred to by the location. At least one printer emulation device referred to by the location must be available, and the job running emulation must be authorized to use the device.

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

Top

Print device (PRTDEV)

Specifies the name of the printer used with this printer device file to print the output. This parameter is ignored if the printer data is spooled. If the output is not spooled and the printer device is being used at the time the job is initiated, the emulation session ends.

Note: If a printer name is not specified, the output is sent to the printer device specified on the **Device (DEV)** parameter of the Create Printer File (CRTPRTF) command. This printer device can be displayed by using the Display File Description (DSPFD) command.

Top

Job name (JOB)

Specifies a job name for the printer emulation job. If the EMLDEV parameter is not specified, and a batch job is to be submitted (*YES is specified for the **Submit job (SBMJOB)** parameter), a job name must be specified. This parameter is ignored when *NO is specified for the SBMJOB parameter.

***EMLDEV**
The job name is the same as the printer emulation device name.

job-name
Specify a name for this printer emulation job.

Top

End Bracket eject (ENDBKTEJT)

Specifies whether SNA printer emulation should force out the emulation output when an SNA End Bracket (EB) is received from the host system. Emulation output is forced out by closing and then reopening the emulation printer file specified in the PRTFILE parameter. When the emulation printer output is ejected, a page eject is performed. The parameter uses the default value of *NO for BSC printer emulation.

***NO** The emulation output is not forced out when SNA printer emulation receives an End Bracket.

***YES** The emulation output is forced out when SNA printer emulation receives an End Bracket. This is done only if the open printer file contains host system data.

Top

Printer file (PRTFILE)

Specifies the printer device file that prints data received from the host system. The printer device file can be spooled or not spooled.

QPEMPRTF
The standard printer file (which specifies SPOOL(*YES)) shipped with the emulation program is used as the printer device file.

printer-device-file-name
Specify the name and library of a user-defined printer device file.

The possible library values are:

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

***CURLIB**
The current library for the job is used to locate the printer device file. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the library where the printer device file is located.

Top

Timeout wait eject (TIMOUTEJT)

Specifies whether printer emulation forces out the emulation output when a time-out has occurred while waiting for host system data. The emulation output is forced out only if the open printer file contains host system data. The wait interval can be specified in number of minutes or seconds. If a value is specified for both minutes and seconds, then these values will be added together.

Note: When the emulation printer output is ejected, a page eject is also performed.

***NONE**
The emulation output is not forced out based on a specified timeout period.

***IMMED**
The emulation output is forced out immediately.

minutes-seconds
Specify a time-out wait interval in minutes or seconds or both.

Minutes
Valid values range from 0 through 99.

Seconds
Valid values range from 0 through 59.

Top

Defer printing spool output (DFRPRTOUT)

Specifies whether spooled output is printed immediately or is delayed. This parameter is ignored when *NO is specified for the Spool output (SPOOL) parameter.

***PRTFILE**
The SCHEDULE value for the printer file controls how the spooled output is printed.

***YES** Spooled output is printed when the spooled file is closed.

***NO** Spooled output can be printed before the spooled file is closed. The printed output does not contain all the data sent by the host system until the spooled file is closed. If the printer is not using spooling, this parameter is ignored.

Note: Once the printer starts printing output from 3270 device emulation, spooled output from other jobs sharing the printer does not print until the spooled file that is currently printing is complete.

Top

Spool output (SPOOL)

Specifies whether the output data for the printer device file is spooled. If *NO is specified on this parameter, output is sent to the printer specified on the **Print device (PRTDEV)** parameter. If the output is spooled, it is sent to the output queue specified on the **Spooled output queue (OUTQ)** parameter of the Create Printer File (CRTPRTF) command for the printer file specified on the PRTFILE parameter. The Output Queue value can be displayed by using the Display File Description (DSPFD) command and specifying the printer file name.

Note: If *JOB is specified on the OUTQ parameter, output is sent to the output queue specified on the OUTQ parameter of the job description specified on the JOBD parameter. This value can be displayed by using the Display Job Description (DSPJOBD) command and specifying the job description name.

***PRTFILE**
The value specified on the **Spool output (SPOOL)** parameter of the Create Printer File (CRTPRTF) command determines whether spooling is performed.

***YES** The data is spooled.

***NO** The data is not spooled; it is sent directly to the device and is printed as the output becomes available.

Top

Open printer file (OPNPRTF)

Specifies when the printer file is opened during the SNA 3270 printer emulation session. If the printer data is not spooled, then the printer will be allocated to your job when the printer file is opened. If the printer data is spooled, then the spool writer is allocated to your job after the printer file is opened depending on the value of the DFRPRTOUT parameter.

This parameter is not allowed if specified for BSC 3270 printer emulation.

***IMMED**
The printer file is opened immediately after starting the 3270 printer emulation session.

***RCVDTA**
The printer file is opened after first receiving print data from the host system.

Top

Print positions per line (NUMCOL)

Specifies the number of columns in a line when creating the printed output.

***PRTFILE**
The printer file PAGESIZE(width) value contains the number of columns per line. This value is used if *IMMED is specified on the open printer file (OPNPRTF) parameter, or *RCVDTA is specified on the OPNPRTF parameter and the maximum print positions (MPP) value is not sent from the host system. Otherwise, the MPP value sent from the host system is used.

number-of-columns
Specify the number of columns per line in the printed output. Valid values range from 1 through 378.

Top

Lines per page (NUMLIN)

Specifies the number of lines per page when creating the printed output.

***PRTFILE**
The printer file PAGESIZE(length) value contains the number of lines per page. This value is used if *IMMED is specified on the open printer file (OPNPRTF) parameter, or *RCVDTA is specified on the OPNPRTF parameter and the maximum page length (MPL) value is not sent from the host system. Otherwise, the MPL value sent from the host system is used.

number-of-lines
Specify the number of lines per page in printed output. Valid values range from 1 through 255.

Top

Lines per inch (LPI)

Specifies the number of lines per inch when creating the printed output.

***PRTFILE**
The printer file LPI value contains the number of lines per inch. This value is used if *IMMED is specified on the open printer file (OPNPRTF) parameter, or *RCVDTA is specified on the OPNPRTF parameter and the set line density (SDL) value is not sent from the host system. Otherwise, the SDL value sent from the host system is used.

lines-per-inch
Specify the number of lines per inch in the printed output. Valid values are 3, 4, 6, 7.5, 7,5, 8 and 9. Values 3, 7.5 and 7,5 are valid only for double-byte character set (DBCS) printer devices.

Top

Message queue (MSGQ)

Specifies the message queue to which operational messages for this device are sent.

***DSPDEV**
The current display station message queue is used.

***NONE**
No messages are sent to message queues other than the job log for the printer emulation job.

The possible library values are:

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

***CURLIB**
The current library for the job is used to locate the message queue. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the library where the message queue is located.

message-queue-name
Specify the name and library of the message queue where messages are sent when the printer emulation job is running.

Top

Language character set (CHRSET)

Specifies the 3-character country or region keyboard language identifier which represents a specific full character identifier (CHRID - comprised of a character set and code page) that is used by the printer being emulated.

***SYSVAL**
The current QKBDTYPE system value is used.

country-keyboard-identifier
Specify the country or region keyboard language identifier to be used.

***TRNTBL**
Allows a user-defined translation table to be used. The character translation is defined in the translation table specified by the **Outgoing translation table (TRNTBLOUT)** parameter. The CHRID associated with the QKBDTYPE system value will be used by the printer being emulated.

Top

Submit job (SBMJOB)

Specifies whether the printer emulation should be done as a separate job or as part of this job.

***YES** A specific job is submitted to do the printer emulation. The job attributes are determined from the job description specified by the **Job description (JOBD)** parameter. The job uses your user profile.

***NO** Printer emulation is done in the current job.

Top

Job description (JOBD)

Specifies the job description for the job that is being submitted for 3270 printer emulation. This parameter is ignored when *NO is specified for the **Submit job (SBMJOB)** parameter.

QBATCH
The job description QBATCH is used for the job.

The possible library values are:

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

***CURLIB**
The current library for the job is used to locate the job description name. If no library is specified as the current library for the job, QGPL is used.

library-name
Specify the library where the job description name is located.

job-description-name
Specify the name and library of the job description associated with the job.

Top

Outgoing translation table (TRNTBLOUT)

Specifies the outgoing translation table used to translate characters sent from the host system to 3270 Emulation.

***CHRSET**
Specify that translation is done when data is sent from the host system using the character set specified on the **Language character set (CHRSET)** parameter.

The possible library values are:

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

***CURLIB**
The current library for the job is used to locate the translation table. If no current library entry exists in the library list, QGPL is used.

library-name
Specify the library where the translation table is located.

table-name
Specifies the name and library of the table which is used for outgoing translation.

Top

Job queue (JOBQ)

Specifies the name of the job queue in which this 3270 printer emulation job is placed. This parameter is ignored when *NO is specified for the **Submit job (SBMJOB)** parameter.

***JOBD**
The submitted job is placed in the job queue associated with the job description specified in the (JOBD) parameter.

job-queue-name
Specify the name and library of the job queue to contain the submitted job.

The possible library values are:

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

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

library-name
Specify the library where the job queue is located.

Top

End emulation conditions (ENDCOND)

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

This parameter is not allowed if specified for BSC 3270 printer emulation.

***NONE**
No additional ways to end 3270 printer emulation are requested.

***DACTLU**
The 3270 printer emulation session ends if it receives an SNA DACTLU from the host system.

***ENDBKT**
The 3270 printer emulation session ends if it receives an SNA end bracket from the host system. Please consider the following before selecting this end condition:

- This end condition should be used only when you need to print one host system file for the duration of the session. An end bracket may occur after printing the first file, and the 3270 session ends before a second file can print.

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

- This end condition should be used only when you need to print one host system file for the duration of the session. An UNBIND may occur after printing the first file, and the 3270 session will end before a second file can print.

Top

Acknowledge form feed (FORMFEED)

Specifies whether to acknowledge a form-feed instruction located in the first character position of the first print line for a 3270 Information Display System data-stream compatibility (DSC) LU3 printer.

This parameter is ignored for an SNA character string (SCS) LU1 printer.

***YES** The form-feed instruction is acknowledged. The print position advances to a new page.

***NO** The form-feed instruction is ignored. The print position does not advance to a new page.

Top

Configuration entry (EMLCFGE)

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

***NONE**
No configuration entry is named and the configuration entry defaults are used.

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

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

Top

Examples

Example 1: Printing Data to Standard Emulation Printer File

```
STRPRTEML   EMLDEV(HOSTPRT4)
```

This command starts a batch job by accepting data from the HOSTPRT4 device and prints the data to the standard emulation printer file (QPEMPRTF). The job is named HOSTPRT4 and runs until the job is canceled. Messages are sent to the current work station message queue.

Example 2: Emulating a Printer in the Current Job

```
STRPRTEML   EMLDEV(HOSTPRT5)  SBMJOB(*NO)
```

This command does printer emulation in the current job by accepting data from the HOSTPRT5 device, and writing the data to the standard emulation printer device file (QPEMPRTF). The request is active until it ends through the End Printer Emulation (ENDPRTEML) command, or until the job is canceled.

Example 3: Printing Output Immediately

```
STRPRTEML   EMLCTL(EMLCTL1)  TIMOUTEJT(10)
            DFRPRTOUT(*NO)  NUMLIN(96)
```

This command starts a batch job by accepting data from the device and printing the data in printer file QPEMPRTF. If a timeout of 10 minutes occurs, printer emulation forces out the emulation output. The output prints immediately; the maximum number of lines per page is 96.

Top

Error messages

***ESCAPE Messages**

CPF85EB
3270 device emulation session ended.

CPF85ED
Values other than ENDCOND(*NONE) are not supported.

CPF85EE
*RCVDTA on the OPNPRTF parameter is not supported.

CPF8510
Internal error occurred on device &1.

CPF8511
Emulation ended by errors on device &2.

CPF8512
Emulation ended because device &2 was held.

CPF8513
Emulation ended by errors on device &2.

CPF8514
Error recovery stopped on device &1.

CPF8515
3270 emulation session ended by host.

CPF8516
No match between host and device &2.

CPF8517
Received more than maximum number of fields allowed.

CPF8518
Emulation ended because of internal failure in system.

CPF8519
Function check in 3270 emulation.

CPF8561
Printer emulation not started.

CPF8564
Printer emulation job &3/&2/&1 ended.

CPF8570
Translate of 3270 printer data stream failed.

CPF8579
Cannot open printer file &1 in library &2.

CPF8580
File &1 is not a printer file or has been overridden.

CPF8582
Cannot open printer file &1 again in library &2.

CPF8583
Printer emulation cannot open required file.

Top

Start Printer Writer (STRPRTWTR)

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

Parameters
Examples
Error messages

The Start Printer Writer (STRPRTWTR) command starts a spooling writer to the specified printer. The writer, which is a system job, takes spooled files from an output queue and produces (writes) the output on the printer device. This command specifies the name of the printer, the names of the output and message queues used, and the name of the writer.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, its own device, and only one type of writer (print, remote, or diskette) can be active to a single output queue. A writer that has been started can be actively writing output or waiting for a file entry to be put on the output queue. The writer can be changed, held, or canceled if the Change Writer (CHGWTR), Hold Writer (HLDWTR), or End Writer (ENDWTR) command is used. Because each writer runs independently of the job that started it, you can continue doing other work on the system after you start a writer.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Printer	*Name*, *ALL, *SYSVAL	Required, Positional 1
OUTQ	Output queue	Single values: ***DEV** Other values: *Qualified object name*	Optional, Positional 2
	Qualifier 1: Output queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGQ	Queue for writer messages	Single values: ***DEVD**, *REQUESTER Other values: *Qualified object name*	Optional, Positional 4
	Qualifier 1: Queue for writer messages	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
FORMTYPE	Form type options	*Element list*	Optional
	Element 1: Form type	*Character value*, ***ALL**, *STD, *FORMS	
	Element 2: Message option	***INQMSG**, *MSG, *NOMSG, *INFOMSG	
FILESEP	File separators	0-9, ***FILE**	Optional
SEPDRAWER	Drawer for separators	1-255, ***DEVD**, *FILE	Optional
WTR	Writer	*Name*, ***DEV**	Optional, Positional 3
AUTOEND	Auto-end options	*Element list*	Optional, Positional 5
	Element 1: Automatically end writer	***NO**, *YES	
	Element 2: If yes, when to end	***NORDYF**, *FILEEND	
ALWDRTPRT	Allow direct print	***NO**, *YES	Optional
ALIGN	Align page	*FILE, ***WTR**, *FIRST	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
INIT	Initialize printer	***WTR**, *FIRST, *ALL	Optional
FILE	Spooled file	*Name*, ***NONE**, *LAST	Optional
JOB	Job name	Single values: ***** Other values: *Qualified job name*	Optional
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, ***ONLY**, *LAST, *ANY	Optional
JOBSYSNAME	Job system name	*Name*, ***ONLY**, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: ***ONLY**, *LAST Other values: *Element list*	Optional
	Element 1: Creation date	*Date*	
	Element 2: Creation time	*Time*, ***ONLY**, *LAST	
PAGE	Starting page	*Integer*, ***BEGIN**	Optional

Top

Printer (DEV)

Specifies the printer device used to print the spooled file. The device must be available before the writer can be started.

This is a required parameter.

***ALL** Start a printer writer for every printer configured on the system.

***SYSVAL**
Start a printer writer for the system default printer.

name Specify the name that the printer device being started is identified by.

Top

Output queue (OUTQ)

Specifies the output queue from which the writer processes spooled files.

Single values

***DEV** Use the default output queue associated with the printer specified on the **Printer (DEV)** parameter.

Qualifier 1: Output queue

name Specify the name of the output queue that the writer processes spooled files from.

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 output queue. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the output queue is located.

Top

Queue for writer messages (MSGQ)

Specifies the message queue that is used by this user.

Single values

***DEVD**
The messages are sent to the message queue specified in the printer's device description.

***REQUESTER**
The messages are sent to the workstation message queue of the workstation of the user who started the process. If this value is specified for a batch job, *DEVD is used.

Qualifier 1: Queue for writer messages

name Specify the name of the message queue where writer messages should be sent.

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 message queue. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the message queue is located.

Top

Form type options (FORMTYPE)

Specifies which form type should be selected to print. This parameter specifies that only the files with this form type are processed now. All other files are left on the output queue as available. If you wish to change the type of form to be printed after the writer is started, you should use the Change Writer (CHGWTR) command.

Note: The form load message is issued when the spooled file to be printed has a form type different from the form type of the last spooled file that was printed on the device. The last form type printed is kept from the last STRPRTWTR, CHGWTR, or VRYCFG command issued.

Consider the following example:

1. The last spooled file printed on printer PRT01 had the form type *STD.
2. The user changes the form type on PRT01 to XYZ using the following command:
   ```
   CHGWTR  PRT01  FORMTYPE(XYZ)
   ```
3. No spooled file with the form type XYZ is printed on PRT01.
4. The user then sends a spooled file with the form type *STD to PRT01. The form load message is not issued, despite the intervening CHGWTR command, because the last spooled file printed on PRT01 had the same form type as the spooled file being printed.

   The form load message would be issued if a spooled file with the form type XYZ were actually printed on PRT01.

Element 1: Form type

***ALL** All available files on the output queue will be processed regardless of their form type.

***FORMS**
All available files on the output queue with the same form type are processed as a group before the writer moves on to the next form type. The writer first chooses the first available file on the queue. After the first file is complete, all files with the same form type will be processed. The writer again chooses the first available file on the queue and repeats the process for that form type.

***STD** The writer processes spooled files with a form type of *STD.

form-type
Specify the type of form for which you want spooled files processed.

Element 2: Message option

***INQMSG**
An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer.

***INFOMSG**
An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

***MSG** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the printer and an informational message is sent when no spooled files requiring this form type remain in the output queue.

***NOMSG**
Neither an inquiry message nor an informational message is sent to the message queue.

Top

File separators (FILESEP)

Specifies how to control the number of file separator pages that are printed before each file.

***FILE** The number of separators that was specified for each individual file is used.

0-9 Specify the number of separator pages to print. Whenever you respond to the change form type message indicating that a new form type has been put on the printer, the writer issues a message inquiring how many file separator pages should be printed with the new form type.

Top

Drawer for separators (SEPDRAWER)

Specifies which paper drawer is selected for printing job and file separators.

***DEVD**
The value stored in the device description for the printer is used.

***FILE** The separator pages are printed from the same drawer as the spooled file.

1 The separator pages are printed from drawer 1.

2 The separator pages are printed from drawer 2.

3 The separator pages are printed from drawer 3.

1-255 Specify a number to indicate the drawer from which the separator pages are printed.

Note: For some printers, SEPDRAWER(3) implies an envelope drawer.

Top

Writer (WTR)

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

***DEV** The name of the writer is the same as the name of the printer device specified on the **Printer (DEV)** parameter.

name Specify the name by which the writer being started is identified.

Top

Auto-end options (AUTOEND)

Specifies whether the writer ends automatically.

Element 1: Automatically end writer

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

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

Element 2: If yes, when to end

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

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

Top

Allow direct print (ALWDRTPRT)

Specifies whether the printer writer allows files to be printed directly to the printer. A file printed directly to the printer is created by specifying SPOOL(*NO) for a printer file. When direct printing is allowed, the non-spooled printer file is printed immediately if the printer is available or, if the printer is busy, the non-spooled printer file waits until the printer is available. The maximum wait is the length of time specified on the WAITFILE parameter on the printer file, after which the job is automatically canceled. The user can cancel a non-spooled printer file only with an End Job (ENDJOB) command.

***NO** The printer does not allow non-spooled printer files to be printed to the device.

***YES** The printer can be used to print spooled and non-spooled output. See the Create, Change, or Override Printer File (CRTPRTF, CHGPRTF, or OVRPRTF) command to set the value of the WAITFILE parameter.

Note: Nonspooled files wait up to 30 seconds regardless of whether the value specified on the WAITFILE parameter is less than 30 seconds.

Top

Align page (ALIGN)

Specifies how to control the forms alignment.

***WTR** The writer keeps track of the output that is printed and issues a forms alignment message whenever it determines that forms may need to be aligned.

***FILE** The forms alignment message is issued for every file that has *YES specified for the **Align page** prompt (ALIGN parameter). This option should be taken whenever the automatic forms alignment control provided by the writer does not provide the desired results.

***FIRST**
The forms alignment message is issued only for the first file printed. No alignment messages are issued when subsequent errors occur on the printer.

Top

Initialize printer (INIT)

Specifies how often to initialize the printer device.

Note: This parameter is ignored if TRANSFORM(*YES) or a user data transform program was specified on the printer device description.

***WTR** The writer initializes the printer device when necessary.

***FIRST**
The writer initializes the printer device only before the first file is printed, or after a device error occurs.

***ALL** The writer initializes the printer device before each file and each copy of the file is printed.

Top

Spooled file (FILE)

Specifies the first (or only) spooled file to print on the printer. If several files are available on the output queue, the next file produced is the first one available with the highest priority.

***NONE**
No spooled file name is specified; the first spooled file that becomes available on the output queue is processed first.

***LAST**
The spooled file which was being printed when the writer ended will start to print when the writer is restarted.

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

Top

Job name (JOB)

Specifies the name of the job that created the spooled file. This parameter is valid only if a spooled file name is specified on the **Spooled file (FILE)** parameter.

Single values

***** The job from which this Start Printer Writer (STRPRTWTR) command was issued is the job that created the spooled file.

Qualifier 1: Job name

name Specify the name of the job that created the spooled file. If no other job qualifiers are given, all of the jobs currently in the system are searched for the simple name of the job.

Qualifier 2: User

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

Qualifier 3: Number

000000-999999
Specify the system-assigned job number.

Top

Spooled file number (SPLNBR)

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

***ONLY**
Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

***LAST**
The spooled file with the highest number and the specified file name is used.

***ANY** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

1-999999
Specify the number of the specified file from the job on the specified output queue that is to be processed first.

Top

Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

***ONLY**
There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file create date and time.

***CURRENT**
The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

***ANY** The job system name is not used to determine which spooled file is used. Use this value when the spooled file create date and time parameter is to take precedence over the job system name when selecting a spooled file.

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 one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

***LAST**
The spooled file with the latest create date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

Element 1: Creation date

date Specify the date the spooled file was created.

Element 2: Creation time

***ONLY**
There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.

***LAST**
The spooled file with the latest create time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.

time Specify the time the spooled file was created.

Top

Starting page (PAGE)

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

***BEGIN**
The restart page of the spooled file is the first page to print.

Note: If this value has not been changed using the Change Spooled File Attribute (CHGSPLFA) command, the file will print in the same manner as it would if the value were *BEGIN.

integer
Specify the number of the first page to print. This number must be within the page range of the file to be valid.

Top

Examples

```
STRPRTWTR   DEV(QSYSPRT)  OUTQ(QPRINTS)  WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named QPRINTS and prints the output on the printer named QSYSPRT. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

Top

Error messages

***ESCAPE Messages**

CPF0906
A duplicate job named &3/&2/&1 was found.

CPF1338
Errors occurred on SBMJOB command.

CPF1764
Writer already started for device &1.

CPF1842
Cannot access system value &1.

CPF2115
Object &1 in &2 type *&3 damaged.

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

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

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

CPF3309
No files named &1 are active.

CPF3310
Writer &1 already started.

CPF3330
Necessary resource not available.

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

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

CPF3343
Duplicate job names found.

CPF3347
Device &1 not found.

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

CPF3362
Objects in QTEMP not valid for parameter values.

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

CPF3369
Device &1 not printer device.

CPF336B
Not found or not authorized to driver exit program &1 in library &2.

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

CPF346A
Transform exit program &1 in library &2 not found or user is not authorized.

CPF346B
Driver exit program &1 in library &2 not found or user is not authorized.

CPF3463
Output queue for device &1 not found.

CPF3464
Not authorized to output queue &1 in library &2.

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

Top

Start Query Management Proc (STRQMPRC)

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

Parameters
Examples
Error messages

The Start Query Management Procedure (STRQMPRC) command allows you to run a query management procedure that was saved as a member in a source file.

Top

Parameters

Keyword	Description	Choices	Notes
SRCMBR	Source member	*Name*	Required, Positional 1
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QQMPRCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RDB	Relational database	*Simple name*, ***NONE**, *CURRENT	Optional
RDBCNNMTH	Connection Method	***DUW**, *RUW	Optional
USER	User	*Name*, ***CURRENT**	Optional
PASSWORD	Password	*Character value*, *NONE	Optional
NAMING	Naming convention	***SYS**, *SQL, *SAA	Optional
ALWQRYDFN	Allow information from QRYDFN	***NO**, *YES, *ONLY	Optional
CMDSRCFILE	Command source file	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Command source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
CMDSRCMBR	Source member	*Name*, ***FIRST**	Optional
ALWDSPLAY	Display screens	***YES**, *NO	Optional

Top

Source member (SRCMBR)

Specifies the source file member that contains the query management procedure to be run.

This is a required parameter.

name Specify the name of the member.

Top

© Copyright IBM Corp. 1998, 2006

Source file (SRCFILE)

Specifies the source file that contains the query management procedure to be run.

This is a required parameter.

Qualifier 1: Source file

QQMPRCSRC
IBM-supplied source file QQMPRCSRC is used.

name Specify the name of the source file.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Top

Relational database (RDB)

Specifies the relation database that is accessed during the processing of this command.

***NONE**
The local database is accessed. If the user is connected to a remote database, the connection is reset to local and remains local until completion of this command.

***CURRENT**
The relational database to which the user is currently connected is accessed.

name Specify the name of the relational database that is accessed. The database must have an entry in the relation database directory.

Top

Connection Method (RDBCNNMTH)

Specifies the connection method to use.

***DUW**
Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

Top

User (USER)

Specifies the user name sent to the remote system when starting the conversation.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name being used for the application requester job.

Top

Password (PASSWORD)

Specifies the password to be used on the remote system.

***NONE**
No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

character-value
Specify the password of the user name specified on the USER parameter.

Top

Naming convention (NAMING)

Specifies the naming convention used for naming objects.

***SYS** The system naming convention is used (name/object-name).

***SQL** The SQL naming convention is used (database-name.object-name). If NAMING(*SQL) is specified, CMDSRCFILE(*LIBL) cannot be specified or allowed as a default value for locating any of the objects specified on other parameters on this command.

***SAA** The SQL naming convention is used (database-name.object-name). If NAMING(*SAA) is specified, CMDSRCFILE(*LIBL) cannot be specified or allowed as a default value for locating any of the objects specified on other parameters on this command.

Top

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query or form information is taken from a query definition (QRYDFN) object when no query management query (QMQRY) or query management form (QMFORM) object can be found using the specified object name. Any information that has to be derived in this way is discarded when the common programming interface (CPI) command in the procedure is completed. No query management objects are created.

***NO** The information is not taken from a QRYDFN object.

***YES** The information is taken from a QRYDFN object when the specified QMQRY or QMFORM object is not found.

***ONLY**
Information can be derived only from a QRYDFN object. Query management objects are ignored.

Top

Command source file (CMDSRCFILE)

Specifies the command source file that query management uses to run a command procedure. A command procedure can only contain query management set commands which can set application variables as well as query management variables that start with the 'DSQ' value.

The supported DSQ variables are:

- DSQCMTLV
- DSQCONFIRM
- DSQOAUTH
- DSQSDBNM
- DSQSRUN

Single values

***NONE**
A command source file is not used. The CMDSRCMBR parameter is ignored.

Qualifier 1: Command source file

name Specify the name of the command source file.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Top

Source member (CMDSRCMBR)

Specifies the command source member that query management uses to run a command procedure. A command procedure can only contain query management set commands which set variables that start with the 'DSQ' value.

***FIRST**
The first member is used.

name Specify the name of the command source member.

Top

Display screens (ALWDSPLAY)

Specifies the display mode used. The query management session is set to interactive mode if ALWDSPLAY(*YES) is specified. If ALWDSPLAY(*NO) is specified, then the session is set to batch mode. The mode is automatically set to batch when running this command in a batch environment.

***YES** Displays are shown when used in an interactive session. This mode allows you to interact with the query management commands in the procedure.

***NO** No displays are shown.

Top

Examples

Example 1: Running a Query Management Procedure

```
STRQMPRC   SRCMBR(MYPROC)  SRCFILE(RPTLIB/PROCFILE)
```

This command starts the query management procedure stored as the member named MYPROC in the source file named PROCFILE in the RPTLIB library.

Example 2: Taking Information From QRYDFN Objects

```
STRQMPRC   SRCMBR(MYPROC)  SRCFILE(PROCFILE)
           ALWQRYDFN(*YES)  ALWDSPLAY(*NO)
```

This command starts the query management procedure stored as the member named MYPROC in the first file named PROCFILE in the library list for the job. Query and form information is allowed to be taken, as needed, from QRYDFN objects when the procedure statements are processed. No reports are shown but they can be printed if the user specifies a print request. Objects are replaced without confirmation if confirmation is not requested by the user. The procedure ends with some errors if processing locates a global variable that is not set or if confirmation was requested before replacing objects that already exist.

Top

Error messages

***ESCAPE Messages**

QWM2701
&1 command failed.

QWM2703
&1 command ended.

QWM2707
*LIBL not allowed when SQL naming applied.

QWM2709
User or password not valid with relational database value.

QWM2710
Password value *NONE only valid with user value *CURRENT.

QWM2712
Character in user name not valid.

Top

Start Query Management Query (STRQMQRY)

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

Parameters
Examples
Error messages

The Start Query Management Query (STRQMQRY) command is used to run a query.

To use this command, you must first identify the query that is to be processed. The query is any single Structured Query Language (SQL) statement in a QMQRY object. The SQL statement can also be taken from a query definition (QRYDFN) object when a QMQRY object does not exist.

You can show the output on the display, print it, or store it in a database file.

If the SQL statement inside the query does not create an answer-set, then no report or output file is created. This happens if the SQL statement inside a query is not valid or the SQL statement is not a SELECT clause.

If the query contains substitution variables, the SETVAR parameter can be used to set the variables for the query. If prompting is enabled, query management asks the user to provide a value for each variable that was not set.

Top

Parameters

Keyword	Description	Choices	Notes
QMQRY	Query management query	*Qualified object name*	Required, Positional 1
	Qualifier 1: Query management query	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTPUT	Output	*****, *PRINT, *OUTFILE	Optional, Positional 2
QMFORM	Query management report form	Single values: ***SYSDFT**, *QMQRY Other values: *Qualified object name*	Optional
	Qualifier 1: Query management report form	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
OUTFILE	File to receive output	*Qualified object name*	Optional
	Qualifier 1: File to receive output	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
OUTMBR	Output member options	*Element list*	Optional
	Element 1: Member	***FIRST**	
	Element 2: Replace or add records	***REPLACE**, *ADD	
DATETIME	Date and time	***YES**, *NO	Optional
PAGNBR	Page numbers	***YES**, *NO	Optional
RDB	Relational database	*Simple name*, ***NONE**, *CURRENT	Optional
RDBCNNMTH	Connection Method	***DUW**, *RUW	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
USER	User	*Name*, ***CURRENT**	Optional
PASSWORD	Password	*Character value*, *NONE	Optional
NAMING	Naming convention	***SYS**, *SQL, *SAA	Optional
ALWQRYDFN	Allow information from QRYDFN	***NO**, *YES, *ONLY	Optional
SETVAR	Set variables	Values (up to 50 repetitions): *Element list*	Optional
	Element 1: Variable name	*Character value*	
	Element 2: Variable value	*Character value*	

Top

Query management query (QMQRY)

Specifies the query management query (QMQRY) to be run.

This is a required parameter.

Qualifier 1: Query management query

name Specify the name of the query to run.

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 where the query is located.

Top

Output (OUTPUT)

Specifies whether the output from the command is shown at the requesting work station, printed with the job's spooled output, or directed to a database file.

*****
The output produced by the query is formatted with the specified report form and, in interactive mode, sent to the work station that runs the command. If the command is run in batch mode, the output is sent to the default printer used by query management.

***PRINT**
The output produced by the query is formatted with the specified query management form, then sent to the default printer used by query management.

***OUTFILE**
The output produced by the query is written to a database file (table), which is inserted into a collection.

Top

Query management report form (QMFORM)

Specifies which query management report form is to be applied to the answer-set to format the printed or displayed output.

Single values

***SYSDFT**
A default report form is created and used for the report that is printed or displayed.

***QMQRY**
The value specified on the **Query management query (QMQRY)** parameter is used to locate the report form.

Qualifier 1: Query management report form

name Specify the name of the report form.

Qualifier 2: Library

***LIBL** The library list is used to locate the report form.

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

name Specify the name of the library to be searched.

Top

File to receive output (OUTFILE)

Specifies the database file that receives the query output. If the file specified does not exist, the system creates it in the specified library as a table in a collection. If the file is created by this function, the authority for users without specific authority is *EXCLUDE.

Qualifier 1: File to receive output

name Specify the name of the database file that receives the output of the command.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Top

Output member options (OUTMBR)

Specifies the database file member to which the output is directed.

Element 1: Member

***FIRST**
The first member in the file receives the output.

Element 2: Replace or add records

***REPLACE**
The file is cleared before new records are inserted.

***ADD** New records are added after any existing records.

Top

Date and time (DATETIME)

Specifies whether the system date and time are printed on the bottom of each page.

***YES** The system date and time are printed on the bottom of each page.

***NO** The system date and time are not printed on the bottom of each page.

Top

Page numbers (PAGNBR)

Specifies whether page numbers are printed on the bottom of each page.

***YES** The page numbers are printed on the bottom of each page.

***NO** The page numbers are not printed.

Top

Relational database (RDB)

Specifies the relational database that is accessed during the processing of this command.

***NONE**
The local database is accessed. If you are connected to a remote database, the connection is reset to local and remains local until completion of this command. If the connection management method is *DUW the remote connection is left in a dormant state.

***CURRENT**
The relational database to which you are currently connected is accessed.

With *RUW connection management, if the user is connected to a remote database, *OUTFILE cannot be specified on the OUPUT parameter.

With *DUW connection management, if the user is connected to a remote database and OUTPUT(*OUTFILE) is specified, the connection is set to local for the *OUTFILE processing and then the remote connection is restored when the STRQMQRY command is completed.

name Specify the name of the relational database that is accessed. The database you specify must have an entry in the relational database directory.

With *RUW connection management, if the relational database specified is a remote database and OUTPUT(*OUTFILE) is specified, the connection is reset to local for the *OUTFILE processing and remains local when the STRQMQRY command is completed.

With *DUW connection management, if the relational database and OUTPUT(*OUTFILE) is specified, the connection is set to local for the *OUTFILE processing and then the remote connection is restored upon completion of the STRQMQRY command.

Top

Connection Method (RDBCNNMTH)

Specifies the connection method to use.

***DUW**
Connections to several relational databases are allowed. Consecutive CONNECT statements to additional relational databases do not result in the disconnection of previous connections.

***RUW** Only one connection to a relational database is allowed. Consecutive CONNECT statements result in the previous connections being disconnected before a new connection is established.

Top

User (USER)

Specifies the user name sent to the remote system when starting the conversation.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name being used for the application requester job.

Top

Password (PASSWORD)

Specifies the password to be used on the remote system.

***NONE**
No password is sent. The user name specified on the USER parameter is not valid if this value is specified.

character-value
Specify the password of the user name specified on the USER parameter.

Top

Naming convention (NAMING)

Specifies the naming convention used for naming objects.

***SYS** The system naming convention is used (database-name/object-name).

***SQL** The SQL naming convention is used (database-name.object-name). If NAMING(*SQL) is specified, the *LIBL value cannot be specified or allowed to be a default value for locating any of the objects specified on other parameters on this command.

***SAA** The SQL naming convention is used (database-name.object-name). If NAMING(*SAA) is specified, the *LIBL value cannot be specified or allowed to be a default value for locating any of the objects specified on other parameters on this command.

Top

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query or form information is taken from a query definition (QRYDFN) object when no query management query (QMQRY) or query management form (QMFORM) object can be found using the specified object name. Any information that has to be derived in this way is discarded when the command completes. No query management object is created.

***NO** Information is not taken from a QRYDFN object.

***YES** Information is taken from a QRYDFN object when the specified QMQRY or QMFORM object name is not found.

***ONLY**
Information is taken only from a QRYDFN object. Query management objects are ignored.

Top

Set variables (SETVAR)

Specifies the variables that are set by query management before the query is run. Up to 50 variables can be set.

Element 1: Variable name

character-value
Specify a variable name, from 1 to 30 characters. Because lower-case characters in variable names are changed to upper-case characters when passed to the command processing program, you cannot use this parameter to set values for variables with mixed case names.

Element 2: Variable value

character-value
Specify a variable value, from 0 to 55 characters. If you enclose a value in apostrophes, the apostrophes are removed and double apostrophes within the value are condensed to single apostrophes when the value is passed to the command processing program.

Top

Examples

Example 1: Displaying Query Output

```
STRQMQRY   QMQRY(MYLIB/MYQRY)  QMFORM(FORM1)
```

This command runs query management query MYQRY located in library MYLIB. The library list is searched for form FORM1, which is used for the output sent to the display.

Example 2: Taking Information From Either QMQRY or QRYDFN

```
STRQMQRY   QMQRY(MYLIB/MYQRY)  QMFORM(FORM1)  ALWQRYDFN(*YES)
```

This command runs query management query (QMQRY) MYQRY located in library MYLIB. If QMQRY object MYQRY is not found in library MYLIB, the information is taken from query definition (QRYDFN) MYQRY located in library MYLIB. The library list is searched for query management form FORM1 whose information is used to format the output. If QMFORM object FORM1 is not found in the library list, the library list is searched for QRYDFN FORM1, and that information is used to format the output shown on the display.

Example 3: Printing Query Output

```
STRQMQRY   QMQRY(MYLIB/QUERY1)  OUTTYPE(*PRINTER)
```

This command runs query QUERY1 located in library MYLIB. The report is formatted and printed on the printer specified in the printer file associated with the query session.

Example 4: Sending Output to an Existing File

```
STRQMQRY   QMQRY(*CURLIB/MYQRY)
           OUTPUT(*OUTFILE)  OUTFILE(MYTAB)  OUTMBR(*FIRST *ADD)
```

This command runs the query named MYQRY located in the current library for the user's job. The selected data records are added to the previously created table named MYTAB in collection MYCOL.

Example 5: Running a Query Containing Substitution Variables

```
STRQMQRY   QMQRY(MYQUERY)
           SETVAR((VAR1 'select * from mytable')
                  (VAR2 'where salary > 15000'))
```

This command runs query MYQUERY, which contains only substitution variables, &VAR1 and &VAR2. These two variables contain the entire structured query language (SQL) statement.

Example 6: Changing a Variable

```
STRQMQRY   QMQRY(QRYNAME)  SETVAR((LASTNAME '''Smith'''))
```

This command runs query QRYNAME, setting the variable LASTNAME to the value, 'Smith'.

Top

Error messages

***ESCAPE Messages**

QWM2701
&1 command failed.

QWM2703
&1 command ended.

QWM2707
*LIBL not allowed when SQL naming applied.

QWM2709
User or password not valid with relational database value.

QWM2710
Password value *NONE only valid with user value *CURRENT.

QWM2712
Character in user name not valid.

Top

Start Query (STRQRY)

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

Parameters
Examples
Error messages

The Start Query (STRQRY) command displays the Query for iSeries main menu.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
STRQRY
```

This command shows the main Query Utilities menu.

Top

Error messages

None

Top

© Copyright IBM Corp. 1998, 2006

Start QSH (STRQSH)

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

Parameters
Examples
Error messages

The Start QSH (STRQSH) command, as known as QSH, starts the **qsh** shell interpreter.

If run in an interactive job, STRQSH starts an interactive shell session. If a shell session is not already active in the job, then:

1. A new shell session is started and a terminal window is displayed.
2. **qsh** runs the commands from the file /etc/profile if it exists.
3. **qsh** runs the commands from the file .profile in the user's home directory if it exists.
4. **qsh** runs the commands from the file specified by the expansion of the ENV variable if it exists.

If a shell session is already active in an interactive job, you are reconnected to the existing session. From the terminal window, you can enter shell commands and view output from the commands.

Using the Terminal Window

The terminal window has two parts:

- an input line for entering commands, and
- an output area that contains an echo of the commands you entered and any output generated by the commands.

The terminal window supports the following function keys:

F3 (Exit)
Close the terminal window and end the qsh session.

F5 (Refresh)
Refresh the output area.

F6 (Print)
Print the output area to a spool file.

F7 (Up)
Roll output area up one page.

F8 (Down)
Roll output area down one page.

F9 (Retrieve)
Retrieve a previous command. You can press this key multiple times to retrieve any previous command. For example, to retrieve the second to last command you entered, press this key two times. You can also select a specific command to be run again by placing the cursor on that command and pressing this key. When the interactive job is running in a double-byte CCSID, this key is not available.

F11 (Toggle line wrap)
Toggle the line wrap/truncate mode in the output area. In line wrap mode, lines longer than the width of the terminal window are wrapped to the next line. In truncate mode, the portion of a line beyond the width of the terminal window is not shown.

© Copyright IBM Corp. 1998, 2006

F12 (Disconnect)
Disconnect from the **qsh** session. This key only closes the terminal window and does not end the **qsh** session. You can redisplay the disconnected **qsh** session by running STRQSH again.

F13 (Clear)
Clear the output area.

F17 (Top)
Display top of output area.

F18 (Bottom)
Display bottom of output area.

F19 (Left)
Shift output area to the left.

F20 (Right)
Shift output area to the right.

F21 (CL command entry)
Display a command entry window where you can enter CL commands.

Also, you can use SysReq 2 to interrupt the currently running command.

Error messages for STRQSH

***ESCAPE Messages**

QSH0002
Error found with QSH session, reason code &1, errno &2.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	*Character value*, ***NONE**	Optional

Top

Command (CMD)

Specifies the shell command to be run.

The possible values are:

***NONE:**
No command is provided and an interactive session is started.

command
A shell command to run. The command can be a maximum of 5000 bytes in length. If a blank or other special characters are used, the command must be enclosed in apostrophes (’). If an apostrophe is intended, two apostrophes must be used (’’).

Note: The case is preserved when lowercase characters are specified.

Top

Examples

None

Top

Error messages

***ESCAPE Messages**

QSH0002
Error found with QSH session, reason code &1, errno &2.

Top

Start Question and Answer (STRQST)

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

Parameters
Examples
Error messages

The Start Question and Answer (STRQST) command shows the main Question & Answer (Q & A) menu. More information is available in the Basic System Operation information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Top

Parameters

Keyword	Description	Choices	Notes
QSTDB	Q/A database	*Name*, ***SELECT**	Optional, Positional 1
LIB	Lib containing Q/A database	*Name*, ***QSTLIB**	Optional, Positional 2

Top

Q/A database (QSTDB)

Specifies the Question and Answer (Q & A) database with which to work.

The possible values are:

***SELECT**
You are asked to specify a Q & A database. If only one Q & A database exists on the system, it is the default.

question-database
Specify the name of the Q & A database with which to work.

Top

Lib containing Q/A database (LIB)

Specifies the name of the library that contains the Q & A database.

The name of the Q & A database can be qualified by one of the following library values:

***QSTLIB**
The library containing the specified Q & A database is searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in any library to which you are authorized can be selected.

library-name
Specify the name of the library to be searched. If *SELECT is specified on the QSTDB parameter, any Q & A database in the library to which you are authorized can be selected.

© Copyright IBM Corp. 1998, 2006

Top

Examples

```
STRQST
```

This command shows the Question and Answer (Q & A) main menu.

Top

Error messages

None

Top

Start REXX Procedure (STRREXPRC)

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

Parameters
Examples
Error messages

The Start REXX Procedure (STRREXPRC) command calls the interpreter, explicitly specifying the library, file, and source member to interpret.

Top

Parameters

Keyword	Description	Choices	Notes
SRCMBR	Source member	*Name*	Required, Positional 1
SRCFILE	Source file	*Qualified object name*	Optional
	Qualifier 1: Source file	*Name*, **QREXSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PARM	Parameters	*Character value*, ***NONE**	Optional
CMDENV	Command environment	Single values: ***COMMAND**, *CPICOMM, *EXECSQL Other values: *Qualified object name*	Optional
	Qualifier 1: Command environment	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
EXITPGM	Exit program	Single values: ***NONE** Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Exit code	2, 3, 4, 5, 7, 8, 9, 10	

Top

Source member (SRCMBR)

Specifies the name of the source file member that contains the REXX procedure to be started.

This is a required parameter.

Top

Source file (SRCFILE)

Specifies the source file that contains the REXX procedure to be started.

The possible **source file** values are:

© Copyright IBM Corp. 1998, 2006

QREXSRC
The IBM-supplied source file, QREXSRC, contains the REXX procedure.

source-file-name
Specify the name of the source file that contains the REXX procedure to be run.

The possible library values are:

***LIBL** The library list is used to locate the program.

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

library-name
Specify the library name.

Top

Parameters (PARM)

Specifies procedure parameter values passed to the REXX procedure when it is started. These values are accessed through the argument (ARG) instruction within the REXX procedure.

The possible values are:

***NONE**
There are no procedure parameters for the REXX procedure. The ARG instruction returns a null string.

procedure-parameters
Specify the procedure parameter value to pass to the REXX procedure. A maximum of 3000 characters is allowed.

Top

Command environment (CMDENV)

Specifies the initial command environment program to process commands embedded in the REXX procedure. The REXX interpreter will call this environment whenever a command is encountered within the REXX procedure.

The possible values are:

***COMMAND**
The AS/400 control language (CL) command environment is used.

***CPICOMM**
The Common Programming Interface (CPI) for Communications command environment is used.

***EXECSQL**
The Structured Query Language (SQL) Command environment is used. EXECSQL is the command environment used for CL commands that are imbedded within a REXX procedure.

program-name
Specify the name of the program to process commands embedded in the REXX procedure.

The possible library values are:

***LIBL** The library list is used to locate the program.

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

library-name
Specify the name of the library where the program is located.

Top

Exit program (EXITPGM)

Specifies exit programs to be used when the interpreter is called. A maximum of 8 program and exit code pairs can be specified.

Note: If multiple system exit codes are specified, the last one specified is the one taken.

The possible values are:

***NONE**
There are no exit programs for this call.

program-name
Specify a program name. A maximum of 8 program names can be specified.

The possible library values are:

***LIBL** The library list is used to locate the programs.

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

library-name
Specify the name of the library where the programs are located.

exit-code
Specify an exit code. A maximum of 8 exit codes can be specified.

Exit-code	Description
2	The associated program is called whenever an external function or subroutine has been called by the REXX program. The exit program is then responsible for locating and calling the requested routine.
3	The associated program is called whenever the interpreter is going to call a command. The exit program is responsible for locating and calling the command given the command string and the current environment name.
4	The associated program is called whenever a REXX instruction or function attempts an operation on the REXX external data queue.
5	The associated program is called when session input or output operations are attempted.
7	The associated program is called after running each clause of the REXX procedure to determine whether it should be stopped.
8	The associated program is called after running each clause of the REXX program to check if tracing should be turned on or off.

9 The associated program is called before interpretation of the first instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

10 The associated program is called after interpretation of the last instruction of a REXX procedure (including REXX procedures called as external functions and subroutines).

Top

Examples

```
STRREXPRC   SRCMBR(ABC)
```

This command calls the REXX interpreter instructing it to run the source member named ABC in the first QREXSRC source file in the library list (*LIBL).

Top

Error messages

***ESCAPE Messages**

CPF7CFB
Error occurred while processing REXX exit programs.

CPF7CFD
Error occurred running REXX procedure &1.

CPF7CFF
REXX procedure &1 ended; return code &4.

CPF7CF2
REXX procedure &1 not found.

CPF7CF3
Not authorized to source file &2.

CPF7CF4
Cannot allocate REXX procedure &1.

CPF7CF6
Cannot allocate REXX source file &2.

CPF7CF7
REXX external data queue is damaged.

Top

Start Remote Support (STRRMTSPT)

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

Parameters
Examples
Error messages

The Start Remote Support (STRRMTSPT) command creates and varies on all configuration objects needed for remote support. Remote support allows the IBM service organization to access your system. Remote support options that are available include:

- remote work station
- virtual device
- virtual device over a systems network architecture (SNA) connection
- virtual device over a point-to-point protocol (PPP) connection using internet protocol (IP)
- virtual device over a virtual private network (VPN) connection using internet protocol (IP)

Each remote support option has different command parameters that are required for connectivity. If any existing remote support configuration objects are found, they are deleted and then re-created. After the configuration objects have been created, they are varied on. You must provide a user identifier and password before the support person can sign-on your system.

Restrictions:

1. When using DEVCLS(*RMT), the remote work station used by a support organization must be one of those listed in the DSPTYPE and DSPMODEL parameters. If your support organization has a configuration that does not match, you must work with the support person to create the correct configuration objects on your system.
2. When using DEVCLS(*VRT) or DEVCLS(*IPS), the QUSER user profile must not be disabled.

Top

Parameters

Keyword	Description	Choices	Notes
DEVCLS	Device class	*RMT, *VRT, *IPS, *PPP, *VPN	Required, Positional 1
DSPTYPE	Display type	3179, 3180, 3196, 3197, 5251, 5291, 5292	Optional, Positional 2
DSPMODEL	Display model	1, 2, 11, A1, A2, B1, B2, C1, C2, D1, D2, W1, W2, 0001, 0002, 0011	Optional, Positional 3
STNADR	Station address	01-FE, **FE**	Optional, Positional 4
USRPRF	User profile	*Character value*, **QPGMR**	Optional, Positional 5
RSRCNAME	Resource name	*Character value*, ***DFT**	Optional, Positional 6
RMTLOCNAME	Remote location	*Communications name*, **QREMOTE**	Optional, Positional 7
LCLLOCNAME	Local location	*Communications name*, **QLOCAL**, *NETATR	Optional, Positional 8
RMTNETID	Remote network identifier	*Communications name*, *NETATR, ***NONE**	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
MAXLENRU	Maximum length of request unit	241-32767, **32767**, *SAME, *CALC	Optional
DTACPR	Data compression	*SAME, *NETATR, *NONE, *ALLOW, ***REQUEST**, *REQUIRE	Optional
INDTACPR	Inbound data compression	*SAME, *RLE, *LZ9, *LZ10, ***LZ12**, *NONE	Optional
OUTDTACPR	Outbound data compression	*SAME, *RLE, *LZ9, *LZ10, ***LZ12**, *NONE	Optional
MODEM	Modem	*Character value*, ***RSRCNAME**, *RMTPPP, *SELECT	Optional
MDMRMTSYS	Modem remote system	*Character value*	Optional

Top

Device class (DEVCLS)

Specifies the device class for this display station. Different configuration objects are created, depending on the value specified.

This is a required parameter.

***RMT** This device class is for a device connected to a remote work station. The configuration objects created for this option include a line description, a controller description, a display device description, and a printer device description.

***VRT** This device class is for a virtual device. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.

***IPS** This device class is for a virtual device and also supports internet protocol (IP) over SNA sessions on this connection. The configuration objects created for this option include a line description, a controller description, a device description, a virtual control description, and a virtual display device description.

***PPP** This device class supports IP over a point-to-point protocol (PPP) connection. The configuration objects created for this option include a line description, a controller description, a device description, and a PPP profile. DEVCLS(*PPP) requires that the QRETSVRSEC system value be set to '1' to retain server security data. This allows for additional authentication to be performed when the service organization attempts to connect.

***VPN** This device class supports IP over a virtual private network (VPN) connection. The Universal Connection wizard must be run before running the STRRMTSPT command in order to set up a VPN configuration on the system. Once the STRRMTSPT command is entered, eleven alphanumeric characters are displayed on your screen. These characters need to be told to the Customer Support Representative for a complete connection to be established. No configuration objects are created for this option.

Top

Display type (DSPTYPE)

Specifies the workstation display device type which will be used for remote support. A value must be specified for this parameter when DEVCLS has a value of *RMT, *VRT, or *IPS.

Valid display station type values are:

- 3179
- 3180

- 3196
- 3197
- 5251
- 5291
- 5292

Top

Display model (DSPMODEL)

Specifies the model number of the device for this description. A value must be specified for this parameter when DEVCLS has a value of *RMT, *VRT, or *IPS.

The possible values for the device model for each device type are:

TYPE	MODEL
3179	2
3180	2
3196	A1, A2, B1, or B2
3197	C1, C2, D1, D2, W1, or W2
5251	11
5291	1, or 2
5292	1, or 2

Top

Address (STNADR)

Depending on the value specified for the DEVCLS parameter, this parameter specifies the address that will be used for either the station address or the local IP address qualifier. If the DEVCLS is not *PPP or *VPN, then this parameter specifies the station address. The station address is the hexadecimal address by which the local system is known to the remote system. The hexadecimal address is the polling address assigned to this system.

If the DEVCLS is *IPS, *PPP, or *VPN, this parameter is used as the local IP address qualifier. This specifies the value of the last byte of the local internet address that will be used.

FE The hexadecimal value FE is the local system address.

address
Specify a hexadecimal value from 01 to FE.

Top

User profile (USRPRF)

Specifies the name of the user profile that the IBM service personnel will use to sign on to your system. This profile is made the owner of the objects created by the system when remote support is started. The user profile must already exist on your system.

QPGMR
The default system-supplied user profile, QPGMR, is used to sign on to your system.

user-profile-name
Specify the name of the existing user profile that will be used to sign on to the remote system.

Top

Resource name (RSRCNAME)

Specifies the name of the resource used to access iSeries electronic customer support. This parameter is not used for DEVCLS(*VPN).

***DFT**

- For DEVCLS(*PPP):
  - Look for resources being used by the 2771 or 2793 integrated modem. If only one 2771 or 2793 is defined, that resource is used for this PPP line. Note that more than one 2771 or 2793 could be defined, but a 2771 or 2793 resource can only be calculated if only one is defined.
  - If a 2771 or 2793 modem resource cannot be used, determine if any resources are defined for use by electronic customer support (ECS). If an ECS resource is available, that resource is used for this PPP line.
  - If neither a single 2771 or 2793 integrated modem or ECS resource is available, the resource cannot be calculated and it will have to be explicitly specified.
- For all other DEVCLS values (except *VPN): The resource name associated with the shipped default port for accessing electronic customer support is used. This is the first port on the I/O (input/output) adapter in card position B of the first multifunction IOP (input/output processor) on the bus. If this port does not exist on the system, resource name CMN01 is used.

resource-name
Specify the name of the resource used to access electronic customer support.

Top

Remote location (RMTLOCNAME)

Specifies the remote location name of the system with which this object communicates. This parameter is used when DEVCLS is *VRT or *IPS.

QREMOTE
The default system-supplied remote support location name.

remote-location-name
Specify the name of the remote support location.

Top

Local location (LCLLOCNAME)

Specifies the unique location name that identifies the local system to remote devices. The name cannot be the same as that specified for the **Remote location** prompt (RMTLOCNAME parameter). The combination of the names specified for the **Local location** prompt (LCLLOCNAME parameter) and the **Remote location** prompt (RMTLOCNAME parameter) must be unique for each device attached to the same controller. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS.

QLOCAL
The default system-supplied local location name is used.

***NETATR**
The LCLLOCNAME value specified in the system network attributes is used.

local-location-name
Specify the location name to be used to identify the local system to remote devices.

Top

Remote network identifier (RMTNETID)

Specifies the name of the remote network in which the remote system resides. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS.

***NONE**
No remote network name is used.

remote-network-name
Specify the remote network name.

Top

Maximum length of request unit (MAXLENRU)

Specifies the maximum request unit (RU) length allowed. This parameter is used when the DEVCLS parameter has a value of *VRT or *IPS.

32767 The maximum request unit length is 32767 bytes.

***CALC**
The system calculates the maximum value to use.

maximum-request-unit-length
Specify a value, ranging from 241 through 32767 bytes, for the maximum length of incoming request units.

Some common values, based on line type, are:

- SDLC lines: 256, 512, 1024, 2048
- Token-Ring Network lines: 256, 512, 1024, 1985
- X.25 (QLLC) lines: 247, 503, 1015
- X.25 (ELLC) lines: 241, 497, 1009

More information on setting the maximum RU length is in the Communications Configuration book, SC41-5401 book.

Top

Data compression (DTACPR)

Specifies whether data compression is used.

***REQUEST**
Data compression is requested on the session by the local system. However, the request can be refused or changed to lower compression levels by the remote system. Data compression is allowed on the session if requested by the remote system. The requested compression levels for inbound and outbound data are the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

***NONE**
Compression is not allowed on the session.

***ALLOW**
Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.

If data compression is requested by the remote system, the data compression levels used by the session are the lower of the requested levels and the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

***REQUIRE**
Data compression is required on the session. If the remote system does not accept the local system's exact required levels of compression, the session is not established.

The data compression levels that the local system require are the levels specified on the **Inbound data compression** and **Outbound data compression** prompts (INDTACPR and OUTDTACPR parameters).

Top

Inbound data compression (INDTACPR)

Specifies the desired level of compression for inbound data. No data compression occurs if *NONE is specified on the **Data compression** prompt (DTACPR parameter). Adaptive dictionary-based compression is a dynamic compression algorithm, similar to Lempel-Ziv, that compresses previously seen strings to 9-, 10-, and 12-bit codes. This algorithm is referred to as LZ in the following parameters.

***LZ12** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***RLE** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

Top

Outbound data compression (OUTDTACPR)

Specifies the desired level of compression for outbound data. No data compression occurs if *NONE is specified on the **Data compression** prompt (DTACPR parameter).

***LZ12** The LZ algorithm with the 12-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. LZ12 requires the most storage and processing time of the LZ algorithms; however, it compresses the data stream the most.

***RLE** The Run Length Encoding (RLE) algorithm is used. RLE substitutes a 1- or 2-byte sequence in the data stream for each repeated occurrence of the same character. This algorithm requires no storage and less processing time than the other options.

***LZ9** The LZ algorithm with the 9-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ9 requires the least storage and processing time of the LZ algorithms; however, it compresses the data stream the least.

***LZ10** The LZ algorithm with the 10-bit code for repeated substrings in the data stream is used. These codes refer to entries in a common dictionary, created as the data flows between the sender and receiver. The LZ algorithms require storage and extra processing time. The LZ10 table algorithm requires more storage and processing time than the LZ9, but less than the LZ12. The LZ10 compresses the data stream more than the LZ9, but less than the LZ12.

Top

Modem (MODEM)

Specifies the modem description to use for the point-to-point protocol (PPP) profile.

***RSRCNAME**

The modem description will be determined based on the value defined for the RSRCNAME parameter.

- If the resource is defined to use the 2771 integrated modem, the '2771 Internal Modem' description is used.
- If the resource is defined to use the 2772 integrated modem, the '2772 Internal Modem' description is used.
- If the resource is defined to use the 2793 integrated modem, the '2793 Internal Modem' description is used.
- If the resource is defined to use the 2805 integrated modem, the '2805 Internal Modem' description is used.
- If the ECS resource was chosen, the 'IBM 7852-400' modem description is used.
- Otherwise if the resource does not have a pre-defined modem description, MODEM(*RSRCNAME) cannot be used and the modem description will have to be explicitly defined.

***RMTPPP**

Indicates that a modem on a different partition or system is to be used. The internet address or host name where the modem is located must be specified for the **Modem remote system (MDMRMTSYS)** parameter.

***SELECT**

A list of modem descriptions is shown from which you can select the modem to use. This option is only valid when running the STRRMTSPT CL command in interactive mode, otherwise an

error will occur. If you are running interactively, it is recommended that you use the *SELECT value to help ensure that you properly select the modem to use.

'modem-identification'
Specify the name of the modem to use. Note that modem names are case sensitive and must match exactly to the modems defined for the system.

'generic-modem-identification'*
Specify the generic name of the modem you wish to use. A generic modem name is a character string of one or more characters followed by an asterisk (*); for example, 'abc*'. If a generic name is specified, then the FIRST modem name that matches with the generic name will be used. It is recommended that you include as many characters in the modem name string as possible to avoid any ambiguity. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete modem name. The actual modem name chosen will be posted in a message in the joblog.

Top

Modem remote system (MDMRMTSYS)

Specifies the internet protocol (IPv4) address or the host name for the system or partition where the modem to be used for remote support is located.

Note: This parameter is required if *RMTPPP is specified for the **Modem (MODEM)** parameter. If the MODEM parameter has any value other than *RMTPPP, this parameter cannot be specified.

character-value
Specify the IP address or host name of the remote system or partition where the modem is located.

Top

Examples

Example 1: Start Remote Support using PPP

```
STRRMTSPT   DEVCLS(*PPP)  STNADR(FA)
```

This command creates and starts a PPP answer profile and associated configuration objects. The local internet address assigned is 169.254.2.250.

Example 2: Start Remote Support for Virtual Device

```
STRRMTSPT   DEVCLS(*VRT)  DSPTYPE(5251)
            DSPMODEL(11)  STNADR(FE)
```

This command creates and varies on the 5251 Model 11 Display Station located at station address FE.

Example 3: Start Remote Support using VPN

```
STRRMTSPT   DEVCLS(*VPN)
```

This command starts remote support over a virtual private network (VPN) connection. Before running this command, the Universal Connection wizard needs to be run, specifying a VPN or multihop type of connection (local or remote) for Electronic Customer Support (ECS).

Example 4: Start Remote Support using a Remote Modem and PPP

```
STRRMTSPT   DEVCLS(*PPP)  STNADR(FE)  MODEM(*RMTPPP)
            MDMRMTSYS('10.1.1.2')
```

This command creates and starts an L2TP remote answer profile and associated configuration objects. Before running this command, the Universal Connection wizard needs to be run on the system or partition with internet address 10.1.1.2 where the modem to be used is located. When running the Universal Connection wizard, you need to specify that the modem should provide connectivity to others and that the connection type is AT&T. An alternative would be to create an L2TP terminator profile using the Remote Access Service New Profile GUI on the system or partition where the modem is located, specifying that outgoing calls are supported.

Top

Error messages

***ESCAPE Messages**

CPF1394
CPF1394 User profile &1 cannot sign on.

CPF9801
Object &2 in library &3 not found.

CPF9899
Error occurred during processing of command.

TCP83A7
QRETSVRSEC=0. Unable to save remote service password for &2.

Top

Start Remote Writer (STRRMTWTR)

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

Parameters
Examples
Error messages

The Start Remote Writer (STRRMTWTR) command starts a spooling writer that sends spooled files from an output queue to a remote system. The writer, which is a system job, takes spooled files from an output queue and sends them to a remote system using SNADS or TCP/IP.

After the spooled file is successfully sent to a remote system, the spooled file will be deleted or saved, as specified by the SAVE spooled file attribute.

More than one writer can be active at the same time (as determined by the spooling subsystem description), and up to 10 writers can be active to the same output queue. Each writer must have a unique writer name, and only one type (print, remote, or diskette) of writer can be active to a single output queue. A writer that has been started can be actively sending output or waiting for a file to be put on the output queue. Optionally, the writer can end automatically when it has processed all the files on the output queue. You can also change, hold, or cancel the writer.

You can continue with other work after starting a writer because each job runs independently. The writer is owned by the user who issues the STRRMTWTR command.

Top

Parameters

Keyword	Description	Choices	Notes
OUTQ	Output queue	Single values: *ALL Other values: *Qualified object name*	Required, Positional 1
	Qualifier 1: Output queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MSGQ	Queue for writer messages	Single values: ***OUTQ**, *REQUESTER Other values: *Qualified object name*	Optional, Positional 3
	Qualifier 1: Queue for writer messages	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
FORMTYPE	Form type options	*Element list*	Optional
	Element 1: Form type	*Character value*, ***ALL**, *STD, *FORMS	
	Element 2: Message option	***NOMSG**, *INQMSG, *MSG, *INFOMSG	
WTR	Writer	*Name*, ***OUTQ**, *SYSGEN	Optional, Positional 2
AUTOEND	Auto-end options	*Element list*	Optional, Positional 4
	Element 1: Automatically end writer	***NO**, *YES	
	Element 2: If yes, when to end	***NORDYF**, *FILEEND	

Top

© Copyright IBM Corp. 1998, 2006

Output queue (OUTQ)

Specifies the output queue that holds the spooled files to be written.

Single values

***ALL** Starts remote writers for every output queue on the system which have a remote system specified. No writer starts if an output queue has a different type of writer already started. The number of ″writers to autostart″ attribute in the output queue determines the number of writers started to a particular queue. This will determine how many remote writer jobs will be sending spooled output to a remote system, from a single output queue.

Qualifier 1: Output queue

name Specify the name of the output queue from which the writer processes output files.

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 be searched.

Top

Queue for writer messages (MSGQ)

Specifies the message queue to which messages created by the writer are sent.

Single values

***OUTQ**
Messages are sent to the message queue of the output queue specified on the OUTQ parameter.

***REQUESTER**
The messages are sent to the workstation message queue of the workstation of the user who started the writer. If this value is specified for a batch job, *OUTQ is used.

Qualifier 1: Queue for writer messages

name Specify the name of the message queue to which messages created by the writer are sent.

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 be searched.

Top

Form type options (FORMTYPE)

Specifies the name of the form type. The writer uses this value to select spooled files from the specified output queue for sending to a remote system.

Note: A spooled file's form type is specified in the device file that produced the spooled file.

Element 1: Form type

***ALL** All form types are processed by the writer.

***FORMS**
The writer first chooses the first available spooled file on the output queue. After the first spooled file is complete, all spooled files with the same form type are processed. The writer then chooses the first available spooled file on the output queue and repeats the process for that form type.

***STD** Only spooled files that specify the standard form type are selected.

form-type
Specify the form type of the spooled files being produced.

Element 2: Message option

***NOMSG**
Neither an inquiry message nor an informational message is sent to the message queue.

***INQMSG**
An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type last sent.

***INFOMSG**
An informational message is sent to the message queue when no spooled files requiring this form type remain in the output queue.

***MSG** An inquiry message is sent to the message queue when a spooled file has a form type that is different than the form type in the remote and an informational message is sent when no spooled files requiring this form type remain in the output queue.

Top

Writer (WTR)

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

***OUTQ**
The name of the writer is the same as that of the output queue specified on the OUTQ parameter. If OUTQ(*ALL) is specified, and you have more than one writer to start, the name for each additional writer is taken from the first nine characters of the output queue followed by a digit. This last digit is sequentially assigned, starting with 2 for the second writer, 3 for the third writer, and continuing through 0 for the tenth writer.

***SYSGEN**
The writer name is generated by the system, starting with 'RMTW000001' and incrementing the numeric part for each successive writer.

name Specify the name by which the writer being started is identified.

Top

Auto-end options (AUTOEND)

Specifies whether the writer ends automatically.

Element 1: Stop Writer Option

***NO** The writer does not end when the last available file has been removed from the output queue. It waits for another spooled file entry to be put on the queue. This is a single value (Element 2 is not specified).

***YES** The writer automatically ends after it has reached the condition specified on the second element of this parameter.

Element 2: Conditions for Stopping Writer

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

***FILEEND**
The writer ends after it finishes processing one spooled file.

Top

Examples

```
STRRMTWTR   OUTQ(RMTOUTQ)  WTR(TOM)
```

This command starts a spooling writer named TOM. This writer takes the output from the output queue named RMTOUTQ and sends the output to the remote system and printer queue specified in the output queue RMTOUTQ. Writer messages are sent to the system operator's message queue, and the writer waits for more output when the queue is emptied.

Top

Error messages

***ESCAPE Messages**

CPF330A
Output queue &1 has RMTSYS specified as *NONE.

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

CPF3310
Writer &1 already started.

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

CPF3362
Objects in QTEMP not valid for parameter values.

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

Top

Start S/36 Session (STRS36)

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

Parameters
Examples
Error messages

The Start System/36 (STRS36) command starts a System/36 environment session (if one is not active already). Even if the System/36 environment is active, this command allows the user to show a menu or run a program or procedure before showing a menu.

When the command ends, the System/36 environment returns to the active or inactive state from which this command is issued.

Restrictions: This command cannot be used if a System/36 procedure is already in process. This command cannot be placed in a procedure or in a program that is started by a procedure.

Top

Parameters

Keyword	Description	Choices	Notes
MENU	Menu	*Name*, ***SAME**	Optional, Positional 1
CURLIB	Current library	*Name*, ***SAME**	Optional, Positional 2
PRC	Procedure or program	*Name*, ***NONE**	Optional, Positional 3
FRCMNU	Force menu	***NO**, *YES	Optional

Top

Menu (MENU)

Specifies the first menu that is shown when the System/36 environment is started.

The possible values are:

***SAME**
The menu specified in the job does not change. If no menu is specified in the job, the initial menu specified in the user profile is shown.

menu-name
Specify the name of the first menu shown when the System/36 environment is started.

Top

Current library (CURLIB)

Specifies the current library to use in the System/36 environment.

The possible values are:

© Copyright IBM Corp. 1998, 2006

***SAME**
The current library does not change. If the current library is *CRTDFT and *SAME is specified, the current library is set to #LIBRARY.

library-name
Specify the name of the library you want to use as the current library in the System/36 environment.

Top

Procedure or program (PRC)

Specifies the name of the procedure or program to run before the menu is shown.

The possible values are:

***NONE**
No procedure or program runs.

procedure-name
Specify the name of the procedure or program to run.

Top

Force menu (FRCMNU)

Specifies whether a menu is shown and what this command does in an active System/36 environment.

The possible values are:

***NO** The specified menu is not shown if the System/36 environment is active when this command is issued. This command does nothing, and the user is returned to the point at which the command was issued.

***YES** The specified menu is shown even if the System/36 environment is active when this command is issued. The current library is set and the program or procedure is run as specified in this command.

Top

Examples

```
STRS36   MENU(USER)  CURLIB(MYLIB)  PRC(INITPROC)
```

This command starts a System/36 environment session. This command:

- Displays the USER menu
- Changes the current library to MYLIB
- Runs procedure INITPROC before showing the USER menu

Top

Error messages

***ESCAPE Messages**

CPF3709
Tape devices do not support same densities.

CPF3738
Device &1 used for save or restore is damaged.

CPF3767
Device &1 not found.

Top

Start S/36 Procedure (STRS36PRC)

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

Parameters
Examples
Error messages

The Start System/36 Procedure (STRS36PRC) command starts a System/36 procedure. It is valid whether or not the System/36 Environment is active, but it is not valid if a System/36 procedure is already running. It cannot be placed in a procedure or in a program that is called by a procedure.

Top

Parameters

Keyword	Description	Choices	Notes
PRC	Procedure	*Name*	Required, Positional 1
CURLIB	Current library	*Name*, ***SAME**	Optional, Positional 2
PARM	Procedure parameters	*Character value*	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

© Copyright IBM Corp. 1998, 2006

Procedure parameters (PARM)

Specifies procedure parameters for the procedure. Procedure parameters allow information to be passed to the procedure. If no parameters are specified, no parameters are passed to the procedure.

Top

Examples

Example 1: Changing the Current Library

```
STRS36PRC   PRC(PROC1)  CURLIB(MYLIB)
```

This command changes the current library to MYLIB and runs procedure PROC1.

Example 2: Listing Files Used by the System

```
STRS36PRC   PRC(CATALOG)  PARM('ALL,F1')
```

This command lists all files used by the System/36 environment.

Top

Error messages

***ESCAPE Messages**

SSP0010

System/36 job ended abnormally.

Top

Start Subsystem (STRSBS)

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

Parameters
Examples
Error messages

The Start Subsystem (STRSBS) command starts a subsystem using the subsystem description specified in the command. When the subsystem is started, the system allocates the necessary and available resources (storage, work stations, and job queues) that are specified in the subsystem description.

Allocating Storage: Storage is allocated to the subsystem according to the storage pool definitions specified in the subsystem description, starting with the lower numbered storage pool definitions. If all the pool definitions cannot be allocated, because the maximum number of storage pools on the system is reached or because insufficient storage is available, messages indicating which pools could not be allocated are sent to the system operator. If storage becomes available later, or if the number of active storage pools is reduced, the available resources are automatically allocated to the subsystem to satisfy its unfulfilled requirements. Any jobs that would normally run in a storage pool that is not allocated are run in the shared storage pool *BASE.

Allocating Work Stations: Work stations are allocated to the subsystem according to the work station entries in the subsystem description. Each work station whose name (or type, if not specified by name) is contained in one of the subsystem description's work station entries, and whose entry specifies AT(*SIGNON), is allocated to this subsystem unless it is currently signed on to another subsystem. The sign-on prompt is displayed on each work station that is allocated. Work stations that are already signed on in another subsystem remain allocated to that subsystem until the subsystem that allowed the sign-on is ended, or until the user transfers the job to this subsystem.

If multiple subsystems specify the same work station in their work station entries, each subsystem, as it is started, attempts to allocate that work station. Each successive subsystem allocates that work station unless a user signs on while the work station is allocated to one of the previously started subsystems. When a signed-on work station is signed off, it still remains allocated to the same subsystem until another subsystem is started that specifies that work station. However, if a work station is varied offline and several active subsystems specify that work station, the subsystem to which the work station is allocated when it is varied online is unpredictable.

Allocating Job Queues: If a job queue is specified in the work entries of the subsystem description, the job queue is allocated to the subsystem. If the job queue does not exist or if it is already allocated to an active subsystem, no job queue is allocated to the subsystem and a message is sent to the system operator. If the job queue later becomes available, it is automatically allocated to the subsystem.

Restrictions:

1. To use this command, you must have:
   - use (*USE) authority to the subsystem description and execute (*EXECUTE) authority to the library that contains that subsystem description.
   - job control (*JOBCTL) special authority.

Top

© Copyright IBM Corp. 1998, 2006

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	*Qualified object name*	Required, Positional 1
	Qualifier 1: Subsystem description	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

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   SBSD(QBATCH)
```

This command starts the batch subsystem named QBATCH.

Example 2: Starting a User Subsystem

```
STRSBS   SBSD(QGPL/TELLER)
```

This command starts the subsystem that is associated with the TELLER subsystem description in the QGPL library. The subsystem name is TELLER.

Top

Error messages

***ESCAPE Messages**

CPF1001
Wait time expired for system response.

CPF1004
Function check occurred during start subsystem.

CPF1010
Subsystem name &1 active.

CPF1011
Start subsystem failed for SBSD &1 in library &2.

CPF1012
No authority to start subsystem.

CPF1013
Subsystem &1 in library &2 not found.

CPF1014
Subsystem &1 not started.

CPF1031
Not authorized to library &1.

CPF1038
No authority to use command.

CPF1049
Cannot allocate subsystem &1 in library &2.

CPF1050
Not enough storage to start subsystem.

CPF1057
Subsystem &1 in library &2 damaged.

CPF1067
Cannot allocate library &1.

CPF1080
Library &1 not found.

CPF1086
Subsystem &1 in &2 allocated to your job.

CPF1099
Subsystem not started because system ending.

CPF3D87
Attempted to use system program QLPCTLIN in QSYS.

Top

Start Search Index (STRSCHIDX)

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

Parameters
Examples
Error messages

The Start Search Index (STRSCHIDX) command allows you to access a search index without using the Help key and the F11 key.

Restrictions:

- You must have use (*USE) authority for the search index.

Top

Parameters

Keyword	Description	Choices	Notes
SCHIDX	Search index	Single values: ***USER** Other values: *Qualified object name*	Optional, Positional 1
	Qualifier 1: Search index	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	

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

© Copyright IBM Corp. 1998, 2006

Examples

```
STRSCHIDX
```

This command accesses the search index names that the user has saved.

Top

Error messages

***ESCAPE Messages**

CPF6E66
Requested help information not available.

Top

Start Support Network (STRSPTN)

Where allowed to run:

- Batch job (*BATCH)
- Batch program (*BPGM)
- Interactive program (*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The Start Support Network (STRSPTN) command allows you to establish an application session through a remote support network to the specified destination application.

This command is provided for customers who want to write their own programs to interface with one of the remote support systems.

Top

Parameters

Keyword	Description	Choices	Notes
ACCOUNT	Account	*Character value*	Required, Positional 1
SPTUSRID	User ID	*Character value*	Required, Positional 2
SPTPWD	Password	*Character value*	Required, Positional 3
FEADEV	Device description	*Name*	Required, Positional 4
DESTAPP	Destination application	*Character value*	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.

© Copyright IBM Corp. 1998, 2006

Top

Password (SPTPWD)

Specifies the network password for the specified user identifier. The password must be registered with the support network.

Note: A support network password can expire. The user must then change the password interactively by using the Work with Product Information (WRKPRDINF) command.

This is a required parameter.

Top

Device description (FEADEV)

Specifies the Front End Application (FEA) device description.

This is a required parameter.

Top

Destination application (DESTAPP)

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

This is a required parameter.

Top

Examples

```
STRSPTN   ACCOUNT(11420880)  SPTNUSRID(ACME)  SPTNPWD(11111)
          FEADEV(QTIFEA)  DESTAPP(AAAAAA)
```

This command establishes a communication path through the remote support network, for user ID ACME operating under password 11111 at account 11420880. The path allows access to application AAAAAA.

Top

Error messages

None

Top

Start Service Agent (STRSRVAGT)

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

Parameters
Examples
Error messages

The Start Service Agent (STRSRVAGT) command allows a user to start an aspect of Service Agent. The aspect to be started is specified by the **Type (TYPE)** parameter.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority and also have use (*USE) authority to the Create Line Description PPP (CRTLINPPP) command to create service configurations.
- You must have job control (*JOBCTL) special authority to run the TYPE(*SBSJOB) of this command.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type	*ACTIVATE, *ACTPWD, *MASTERPWD, *SBSJOB	Optional, Positional 1
ACTPWD	Activation password	*Character value*	Optional
MSTPWD	Master password	*Character value*	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.

© Copyright IBM Corp. 1998, 2006

Top

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.

Top

Master password (MSTPWD)

Specifies the current value of the master password. This may be either the original master password, or the additional value already created.

Note: This is a required parameter when TYPE(*MASTERPWD) is specified.

character-value
Specify the master password.

Top

Examples

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

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

Top

Error messages

***ESCAPE Messages**

CPF9899
Error occurred during processing of command.

Top

Start Service Job (STRSRVJOB)

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

Parameters
Examples
Error messages

The Start Service Job (STRSRVJOB) command starts the remote service operation for a specified job (other than the job issuing the command) so that other service commands can be entered to service the specified job. Any dump, debug, and trace commands can be run in that job until service operation ends. Service operation continues until the End Service Job (ENDSRVJOB) command is run.

Restrictions:

- To use this command, you must be signed on as QPGMR, QSYSOPR, QSRV, or QSRVBAS, or have use (*USE) authority to the user profile of the job being serviced.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	*Qualified job name*	Required, Positional 1
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
DUPJOBOPT	Duplicate job option	***SELECT**, *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

© Copyright IBM Corp. 1998, 2006

Duplicate job option (DUPJOBOPT)

Specifies the action taken when duplicate jobs are found by this command.

***SELECT**
The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, an escape message is issued.

***MSG** An escape message is issued when duplicate jobs are found.

Top

Examples

```
STRSRVJOB   JOB(ABCD)
```

This command starts the remote service operation so that any trace, debug, or dump commands entered in this job are applied to the job named ABCD.

Top

Error messages

***ESCAPE Messages**

CPF3501
Job is already being serviced, traced, or debugged.

CPF3520
Job not found.

CPF3524
More than one job with specified name found.

CPF3531
Job cannot be serviced.

CPF3536
Job completed and cannot be serviced.

CPF3549
Job &1/&2/&3 cannot be serviced.

CPF3676
Not authorized to service specified job.

CPF3909
Service command will not be processed.

CPF3918
Service request canceled.

CPF3938
Already servicing another job.

Top

Start System Service Tools (STRSST)

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

Parameters
Examples
Error messages

The Start System Service Tools (STRSST) command shows the System Service Tools (SST) menu.

Note: Improper use of service tools can damage your server.

You can:

- Start a service function
- Work with active service functions
- Work with disk unit configuration and data
- Work with diskette data recovery
- Work with system partitions
- Work with system capacity
- Work with system security
- Work with service tools user IDs 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.

© Copyright IBM Corp. 1998, 2006

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), the Change TCP/IP Interface (CHGTCPIFC) commands, or the iSeries Navigator.

The Change IPL Attributes (CHGIPLA) command with the STRTCP parameter set to a value of *YES can be used to automatically submit the STRTCP command at the completion of IPL.

The TCP/IP application server jobs that can be started are:

- Apache Software Foundation (ASF) Tomcat server
- Bootstrap Protocol (BOOTP)
- Common Information Model Object Manager (CIMOM)
- 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)
- Post Office Protocol (POP)
- Quality of Service (QoS)
- Remote Execution (REXEC)
- Router Daemon (ROUTED)
- Service and Support Proxy (SRVSPTPRX)

© Copyright IBM Corp. 1998, 2006

- 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 (CHGSNMPA) command specifying AUTOSTART(*YES)
- For the RouteD server use the Change RouteD Attributes (CHGRTDA) command specifying AUTOSTART(*YES)
- For the BOOTP server use the Change BOOTP Attributes (CHGBPA) command specifying AUTOSTART(*YES)
- For the TFTP server use the Change TFTP Attributes (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 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 iSeries Navigator.
- Change DIRSVR server attributes, including the AUTOSTART parameter, using iSeries Navigator.

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

When the STRTCP command is issued, the QTCPIP job in the QSYSWRK subsystem is started. The QTCPIP job is used for activating and deactivating TCP/IP interfaces.

Note: The Start TCP/IP (STRTCP) CL command does not need to be issued to use socket applications that run over an SNA network.

Attention: Before attempting to start an X.25 interface, ensure that the remote system information (RSI) for non-DDN X.25 interfaces that use a permanent virtual circuit (PVC) is configured. Use the Add TCP/IP Remote System Information (ADDTCPRSI) command to do this.

Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
STRSVR	Start application servers	***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 Start TCP/IP (STRTCP) command.

***YES** The STRTCP command starts all TCP/IP application servers with a configuration attribute of AUTOSTART(*YES) when TCP/IP is activated.

***NO** The STRTCP command does not start any TCP/IP application servers when TCP/IP is activated.

Note: This parameter can be used when TCP/IP application servers using AnyNet are already active on your system and you intend to also activate TCP/IP. STRTCP STRSVR(*NO) activates TCP/IP processing without starting additional TCP/IP servers.

When TCP/IP or AnyNet is already active, use the Start TCP/IP Server (STRTCPSVR) command to start additional TCP/IP application servers.

Top

Start TCP/IP interfaces (STRIFC)

Specifies whether or not to activate all TCP/IP interfaces that specify AUTOSTART(*YES) when TCP/IP is activated.

***YES** When the QTCPIP job is started by the STRTCP command, the job will attempt to activate all TCP/IP interfaces for which AUTOSTART(*YES) is specified.

***NO** The AUTOSTART(*YES) interface parameter will be ignored. No TCP/IP interfaces will be automatically started.

Note: Specifying STRIFC(*NO) only inhibits the activation of TCP/IP interfaces. It has no effect on interfaces for other AnyNet protocols such as IP over SNA or IP over IPX.

Top

Start point-to-point profiles (STRPTPPRF)

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) STRPTPPRF(*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) STRPTPPRF(*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 Information (ADDTCPRSI) command to do this. Incoming data from a remote system on the X.25 network is not processed unless an RSI entry for the PVC is configured on the X.25 interface before the interface is started.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	*Character value*, *AUTOSTART	Optional, Positional 1
ALIASNAME	Alias name	*Simple name*	Optional

Top

Internet address (INTNETADR)

Specifies the interface to be started.

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

character-value

Specifies the internet address of an interface that had previously been added to the TCP/IP configuration with the ADDTCPIFC command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

© Copyright IBM Corp. 1998, 2006

***AUTOSTART**
The TCP/IP interfaces that are started are those that have set the AUTOSTART parameter to a value of *YES using the Add TCP/IP Interface (ADDTCPIFC) or the Change TCP/IP Interface (CHGTCPIFC) commands.

Top

Alias name (ALIASNAME)

Specifies the name of the interface to be started. The alias name specified must be for an IPv4 interface. If the alias name is for an IPv6 interface, an error message will be returned.

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

simple-name
Specify the name associated with the interface to be started.

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.

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 iSeries Navigator to configure *PPP profiles.

Top

Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	*Character value*, *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

© Copyright IBM Corp. 1998, 2006

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. The interface server job, QTCPIP, must be available. If TCP/IP processing was started by running the STRTCP (Start TCP/IP) command when the system was in restricted state, the STRTCPSVR command will fail.

All servers have an autostart (AUTOSTART) parameter on the associated configuration command (for example, Change FTP Attributes (CHGFTPA)). This parameter indicates if the server should be started when the Start TCP/IP (STRTCP) command is entered. The STRTCPSVR command ignores the value of a server's autostart attribute, unless *AUTOSTART is specified for the **Server application (SERVER)** parameter. If a server has *YES for the AUTOSTART attribute, then the STRTCPSVR command will start the server when SERVER(*AUTOSTART) is specified. Additional servers can automatically be added to list of servers that STRTCPSVR will support by using the ADDTCPSVR (Add TCP/IP Server) CL command.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server application	Single values: ***ALL**, *AUTOSTART Other values (up to 300 repetitions): *Character value*	Optional, Positional 1
RESTART	Restart server	***NONE**, *HTTP, *DNS, *DHCP, *QOS	Optional
HTTPSVR	HTTP server	Single values: ***ALL** Other values: *Element list*	Optional
	Element 1: Server instance	*Name*, *ADMIN	
	Element 2: Instance startup values	*Character value*, ***NONE**	
DNSSVR	DNS server	Single values: ***ALL** Other values: *Element list*	Optional
	Element 1: Server instance	*Character value*	
	Element 2: Instance startup values	*Character value*, ***NONE**	
TCMSVR	TCM server	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Instance name	*Character value*, *ALL	
	Element 2: Instance startup values	*Character value*, ***NONE**	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
TOMCATSVR	ASFTOMCAT server	Single values: ***NONE** Other values: *Element list*	Optional
	Element 1: Server instance name	*Character value*, *ALL	
NTPSRV	SNTP service	*ALL, *CFGFILE, ***CLIENT**, *SERVER	Optional

Top

Server application (SERVER)

Specifies the TCP/IP application servers to be started by this command.

Single values

***ALL** All of the TCP/IP application servers and all HTTP, DNS, and TCM server instances are started.

***AUTOSTART**
All of the TCP/IP application servers that are defined with *YES for the AUTOSTART attribute are to be started.

Other values (up to 300 repetitions)

***ASFTOMCAT**
The Apache Software Foundation (ASF) Tomcat server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*ASFTOMCAT) will result in a diagnostic message if the Apache Software Foundation (ASF) Tomcat server has already been started.

***BOOTP**
The Bootstrap Protocol (BOOTP) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*BOOTP) will result in a diagnostic message if the BOOTP server job has already been started.

***CIMOM**
The Common Information Model Object Manager (CIMOM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*CIMOM) will result in a diagnostic message if the CIMOM server has already been started.

***DBG** The Debug (DBG) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*DBG) will result in a diagnostic message if the DBG server has already been started.

***DDM**
The Distributed Data Manager (DDM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*DDM) will result in a diagnostic message if the DDM server job has already been started.

***DHCP**
The Dynamic Host Configuration Protocol (DHCP) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*DHCP) will result in a diagnostic message if the DHCP server has already been started.

***DIRSRV**
The Lightweight Directory Access Protocol (LDAP) server is started. The LDAP server is also known as the directory services (DIRSRV) server. Subsequent use of the STRTCPSVR command specifying SERVER(*DIRSRV) will result in a diagnostic message if the LDAP server has already been started.

***DLFM**
The DataLink File Manager (DLFM) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*DLFM) will result in a diagnostic message if the DLFM server has already been started.

***DNS** One or more instances of the Domain Name System (DNS) server are started. See the parameter documentation for the **DNS server (DNSSVR)** parameter for more information on DNS server instances. Subsequent use of the STRTCPSVR command specifying SERVER(*DNS) will result in a diagnostic message if the DNS server has already been started.

***DOMINO**
The Lotus Domino (DOMINO) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*DOMINO) will result in a diagnostic message if the DOMINO server has already been started.

***EDRSQL**
The Extended Dynamic Remote SQL (EDRSQL) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*EDRSQL) will result in a diagnostic message if the EDRSQL server has already been started.

***FTP** The File Transfer Protocol (FTP) servers are started, based on the number of servers configured with the Change FTP Attributes (CHGFTPA) command. Subsequent use of the STRTCPSVR command specifying SERVER(*FTP) will start one additional FTP server.

Note: Having more than one FTP server job running can improve the performance of initiating a session when multiple users attempt to connect to the server in a short period of time.

***HOD** The IBM Host On-Demand (HOD) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*HOD) will result in a diagnostic message if the HOD server has already been started.

***HTTP**
One or more instances of the World Wide Web HyperText Transfer Protocol (HTTP) server is started. See the parameter documentation for the **HTTP server (HTTPSVR)** parameter for more information on HTTP server instances. Subsequent use of the STRTCPSVR command specifying SERVER(*HTTP) will result in a diagnostic message if the HTTP server has already been started.

***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 End TCP/IP Server (ENDTCPSVR) command specifying SERVER(*NTP), which will end all active SNTP services. Then run the STRTCPSVR command again, specifying SERVER(*NTP), and either NTPSRV(*CLIENT) or NTPSRV(*SERVER).

***ODPA**
The On-Demand Platform Authentication (ODPA) server is started. Subsequent use of the STRTCPSVR SERVER(*ODPA) command results in a diagnostic message if the ODPA server has already been started.

***ONDMD**
The On Demand Server (ONDMD) is started. Subsequent use of the STRTCPSVR command specifying SERVER(*ONDMD) will result in a diagnostic message if the ONDMD server has already been started.

***POP** The Post Office Protocol (POP) version 3 mail servers are started based on the number of servers configured with the Change POP Server Attributes (CHGPOPA) command. Subsequent use of the STRTCPSVR command specifying SERVER(*POP) will start one additional POP server.

***QOS** The Quality of Service (QoS) server is started. Subsequent use of the STRTCPSVR SERVER(*QOS) command results in a diagnostic message if the QoS server has already been started.

***REXEC**
The Remote Execution (REXEC) servers are started based on the number of servers configured with the Change REXEC Attributes (CHGRXCA) command. Subsequent use of the STRTCPSVR command specifying SERVER(*REXEC) will start one additional REXEC server.

***ROUTED**
The Router Daemon (ROUTED) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*ROUTED) will result in a diagnostic message if the RouteD server has already been started.

***SMTP**
The Simple Mail Transfer Protocol (SMTP) client and server jobs are started. Additional SMTP client and server jobs cannot be started. Subsequent use of the STRTCPSVR SERVER(*SMTP) command results in a diagnostic message if the SMTP server jobs have already been started.

***SNMP**
The Simple Network Management Protocol (SNMP) agent server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*SNMP) will result in a diagnostic message if the SNMP server has already been started.

***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.

***TCM** One or more instances of the Triggered Cache Manager (TCM) server is started. See the parameter documentation for the **TCM server (TCMSVR)** parameter for more information on specifying which TCM server instances to start. Subsequent use of the STRTCPSVR command specifying SERVER(*TCM) will result in a diagnostic message if the indicated TCM server instance has already been started.

***TELNET**
The TELNET server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*TELNET) will start one additional TELNET server.

Note: Having more than one TELNET server job running reduces the chances of having connection attempts refused.

***TFTP** The Trivial File Transfer Protocol (TFTP) servers are started based on the number of servers configured with the Change TFTP Attributes (CHGTFTPA) command. Subsequent use of the STRTCPSVR SERVER(*TFTP) command results in a diagnostic message if the TFTP server jobs have already been started.

***VPN** The Virtual Private Network (VPN) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*VPN) will result in a diagnostic message if the VPN server has already been started.

***WEBFACING**
The WebFacing (WEBFACING) server is started. Subsequent use of the STRTCPSVR command specifying SERVER(*WEBFACING) will result in a diagnostic message if the WebFacing server has already been started.

Top

Restart server (RESTART)

Specifies whether to restart the selected server when the STRTCPSVR command is run.

The SERVER parameter value specified must be one of the following:
- *ALL
- *DNS
- *DHCP
- *HTTP
- *QOS

or this parameter is ignored.

***NONE**
Do not restart any server.

***HTTP**
Restart the HTTP server using the values specified on the **HTTP server (HTTPSVR)** parameter. If the server is not currently running when the STRTCPSVR command is run, this parameter is ignored and the server is started.

***DHCP**
The Dynamic Host Configuration Protocol (DHCP) server job that is already running will

reprocess its configuration file and initialize with any changes that it found. If the DHCP processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

***DNS** A Domain Name System (DNS) server job that is already running will reprocess its configuration file and initialize with any changes that it found. If the DNS processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

***QOS** The Quality of Service (QoS) server job that is already running will reprocess its configuration file and initialize with any changes that it found. If the QoS processing server is not currently running or cannot be successfully interrupted for the restart, a diagnostic message will be issued.

Top

HTTP server (HTTPSVR)

Specifies the HTTP server instance to be started as well as any additional startup values to be used by the HTTP server to control the server instance. (This server is also known as the IBM HTTP Server).

If multiple HTTP server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

For more information on the HTTP Server, visit the HTTP Server homepage:
http://www.ibm.com/eserver/iseries/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

-fsccsid [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.

Top

DNS server (DNSSVR)

Specifies the Domain Name System (DNS) server instance to be started as well as any additional startup values to be used by the DNS server to control the server instance.

If multiple DNS server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single values

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

Element 1: Server instance

character-value
Specify the name of the DNS server instance to be started.

Element 2: Instance startup values

***NONE**
No additional startup values are defined.

character-value
Specifies additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified server instance.

Note: The user must have input/output system configuration (*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority, the start request will be rejected.

The list of instance startup values for the DNS Server follows:

-d [n] where n is a number from 1 to 11. This sets the debug level of the server to start.

Top

TCM server (TCMSVR)

Specifies the Triggered Cache Manager (TCM) server instance to be started as well as any additional startup values to be used by the TCM server to control the server instance.

If multiple TCM server instances have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single values

***NONE**
No defined instances of the TCM server will be started.

Element 1: Instance name

***ALL** All defined instances for the TCM server will be started.

character-value
Specify the name of the TCM server instance to be started.

Element 2: Instance startup values

***NONE**
No additional startup values are defined.

character-value
Specify the additional startup values to be used for this server instance. These values will be used to override previously-defined server startup values for the specified instance name.

Note: The user must have input/output system configuration (*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority, the start request will be rejected.

Top

ASFTOMCAT server (TOMCATSVR)

Specifies the Tomcat server instance to be started.

If multiple Tomcat server instance names have been defined, you can choose to start all instances, or start one specific instance by specifying the instance name to be started.

Single values

***NONE**
No defined instances of the Tomcat server will be started.

Element 1: Server instance name

***ALL** All defined instances for the Tomcat server will be started.

character-value
Specify the name of the Tomcat server instance to be started.

Note: The user must have input/output system configuration (*IOSYSCFG) special authority to specify overrides. If instance startup values are specified and the user does not have *IOSYSCFG authority, the start request will be rejected.

Top

SNTP service (NTPSRV)

Specifies the Simple Network Time Protocol (SNTP) services to be started.

Single values

***CLIENT**
The SNTP client is started.

***SERVER**
The SNTP server is started.

***ALL** Both the client and server are started.

***CFGFILE**
The values of the keywords Client Autostart (AUTOSTART) and Server Autostart (SVRAUTOSTR) in the Change SNTP Attributes (CHGNTPA) command configuration file are used to determine what services to start. If the value is set to *YES, that service is started.

Top

Examples

Example 1: Starting All TCP/IP Servers with AUTOSTART(*YES)

```
STRTCPSVR   SERVER(*AUTOSTART)
```

This command starts all of the TCP/IP application servers that have the AUTOSTART attribute in the application configuration set to *YES.

Example 2: Starting All TCP/IP Servers

```
STRTCPSVR   SERVER(*ALL)
```

This command starts all of the TCP/IP application servers that have been configured. For example: If the Change FTP Attributes (CHGFTPA) command was previously used to configure two FTP servers, both servers are started when STRTCPSVR is issued. This example is also true for other TCP/IP application servers.

Where appropriate, the number of servers to start is based on the number of servers configured for the server being started.

Example 3: Starting the TELNET Server

```
STRTCPSVR   SERVER(*TELNET)
```

This command starts the TCP/IP TELNET application server. If the TELNET server was previously started, one additional TELNET server job is started.

Example 4: Restarting the HTTP Server

```
STRTCPSVR   SERVER(*HTTP)  RESTART(*HTTP)
```

This command restarts the TCP/IP HTTP application server for all instances of the HTTP server. If the HTTP server was not currently running, then all defined instances of the HTTP server would be started.

Example 5: Starting an HTTP Server Instance

```
STRTCPSVR   SERVER(*HTTP)  HTTPSVR('http1')
```

This command starts the TCP/IP HTTP application server instance named 'http1' using the startup values previously defined for this server instance.

Example 6: Specifying Startup Values for an HTTP Instance

```
STRTCPSVR   SERVER(*HTTP)  HTTPSVR(HTTP1 '-p 81 -sslport 443')
```

This command starts the TCP/IP HTTP application server instance named HTTP1, and specifies that the server instance should listen on port 81 for unsecure requests and on port 443 for secure requests. The ports defined here will override any previously defined ports to be used by this server instance.

Example 7: Starting a DNS Server Instance

```
STRTCPSVR   SERVER(*DNS)  DNSSVR('dns1')
```

This command starts the TCP/IP DNS application server instance named 'dns1' using the startup values previously defined for this server instance.

Example 8: Specifying Startup Values for a DNS Instance

```
STRTCPSVR   SERVER(*DNS)  DNSSVR(DNS1 '-d 5')
```

This command starts the TCP/IP DNS application server instance named DNS1, and specifies that the server instance should turn on debug level 5.

Top

Error messages

***ESCAPE Messages**

CPF3894
Cancel reply received for message &1.

TCP1A11
&1 failed.

TCP1A77
&1 completed successfully; however errors occurred.

Top

Start TIE Session (STRTIESSN)

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 (STRTIESSN) 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	*Name*	Required, Positional 1
SPTPWD	Password	*Character value*	Required, Positional 2
ACCOUNT	Account	*Character value*, ***RTV**	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.

© Copyright IBM Corp. 1998, 2006

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 Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- If you specify a generic user name for the **Job name (JOB)** parameter, you must have all object (*ALLOBJ) special authority, or be authorized to the Trace Any User function of Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_ALLOBJ_TRACE_ANY_USER, can also be used to change the list of users that are allowed to perform trace operations.
- A trace cannot be defined to trace all job names and all users.
- At least one generic job name must be specified if *YES has been specified for the **Restart after next IPL (RESTRIPL)** parameter.
- Watch-related parameters cannot be specified if *YES is specified for the RESTRIPL 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 Operating System through iSeries 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

© Copyright IBM Corp. 1998, 2006

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	*Name*, *GEN	Required, Positional 1
JOB	Jobs	Single values: ***** Other values (up to 8 repetitions): *Element list*	Optional
	Element 1: Job name	*Qualified job name*	
	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	Values (up to 20 repetitions): *Character value*, ***ALL**, *INITIAL, *SELECT	
JOBTYPE	Job types	***ALL**, *ACTIVE, *NEW	Optional
MAXSTG	Maximum storage to use	1024-4000000, **10000**	Optional
TRCFULL	Trace full	*STOPTRC, ***WRAP**	Optional
JOBTRCTYPE	Trace type	Values (up to 2 repetitions): ***ALL**, *FLOW, *DATA, *TRCTYPE	Optional
TRCTYPE	Trace type	Single values: ***NONE** Other values (up to 50 repetitions): *Element list*	Optional
	Element 1: Component	*Character value*, *ASPMGT, *CLUSTER, *CMTCTL, *DBHSVR, *DBSQL, *DEVCFG, *DIRSRV, *EDRSQL, *ENVVAR, *FILESVR, *FLIGHTRCD, *HPT, *HTTP, *IFS, *IPC, *IPPDRV, *LOCKSPACE, *NETSVR, *NPS, *OPASSIST, *OPENSSL, *POSIXMISC, *POSIXPROC, *PTHREAD, *QNTC, *QSHELL, *RAS, *REGFAC, *SECURITY, *SERVICE, *SOCKETS, *TCPIPCFG, *THREADMGT, *WRKMGT	
	Element 2: Trace level	*ERROR, *INFO, *VERBOSE	
TRCFTR	Trace filter	*Name*, ***NONE**	Optional
RESTRIPL	Restart after next IPL	***NO**, *YES	Optional
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**	

Keyword	Description	Choices	Notes
WCHLICLOG	Watch for LIC log entry	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Major code	*Character value*, *ALL	
	Element 2: Minor code	*Character value*, *ALL	
	Element 3: Comparison data	*Character value*, ***NONE**	
WCHTIMO	Length of time to watch	1-43200, ***NOMAX**	Optional
TRCPGM	Trace program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Trace program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
TRCPGMITV	Time interval	1-9999, ***NONE**	Optional

Top

Session ID (SSNID)

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

This is a required parameter.

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

name Specify the session identifier for this trace.

Top

Job name (JOB)

Specifies which jobs are to be traced.

Single values

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

Element 1: Job name

Qualifier 1: Job name

generic-name
Specify the generic name of the jobs to be traced. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job names that begin with the generic prefix.

***ALL** All jobs names with the specified job user name are traced. *ALL for the job name is considered to be a generic job specification because it will trace all jobs that meet the job user name qualifiers that you specified.

name Specify the name of the job to be traced. Up to eight job names may be specified, but only one generic job name is allowed.

Qualifier 2: User

generic-name
Specify the generic user name of the jobs from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic user name specifies all jobs with user names that begin with the generic prefix.

***ALL** All job user name with the specified job name are traced. *ALL for the job user name is considered to be a generic job specification because it will trace all jobs that meet the job name qualifiers that you specified.

name Specify the user name of the job to be traced.

Qualifier 3: Number

***ALL** All jobs with the specified job name and user name are traced. *ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified.

000001-999999
Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name qualifier or generic user name qualifier is specified.

Element 2: Thread ID to include

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

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

***SELECT**
A list of thread identifiers is shown from which the user can select up to twenty to be traced. This value is only valid if the command is run in an interactive job.

thread-identifier
Specify the identifiers of up to twenty threads whose information is to be included.

Top

Job types (JOBTYPE)

Specifies the types of jobs for which trace data is to be collected.

***ALL** All jobs that meet the job name selection criteria will be included in this trace collection. This includes jobs that are currently active on the system. If a generic job name was specified, new jobs that start after this trace session begins and meet the generic job name selection criteria, will be included in the trace collection as well.

***ACTIVE**
Only jobs that are currently active at the time this trace session is being activated and meet the job name selection criteria will be included in this trace collection.

***NEW** Only new jobs that begin after this trace session is activated and meet the generic job name selection criteria will be included in this trace collection.

Top

Maximum storage to use (MAXSTG)

Specifies the requested maximum amount of storage, in kilobytes (K), to use for the collected trace records. The system will calculate the minimum amount of storage that is necessary for the trace; this minimum storage size calculation is dependent upon the system's processor configuration. The minimum amount of storage may be significantly larger than the size specified on the MAXSTG parameter; the system will use the larger of the two values.

10000 Up to 10,000 kilobytes of storage is used.

1024-4000000
Specify the maximum amount of storage, in kilobytes, to be used to store trace records (one kilobyte equals 1024 bytes).

Top

Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***WRAP**
When the trace storage is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC**
Tracing stops when the trace storage is full of trace records.

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

Each trace type is identified by a special value. Specify trace component special values from the following table:

TRACE DESCRIPTION	SPECIAL VALUE
ASP Management	*ASPMGT
Cluster Resource Services	*CLUSTER
Commitment Control	*CMTCTL
Database Host Server	*DBHSVR
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
iSeries NetServer	*NETSVR
Network Print Server	*NPS
Operational Assistant	*OPASSIST
OpenSSL APIs	*OPENSSL
POSIX Miscellaneous APIs	*POSIXMISC
POSIX Process-Related APIs	*POSIXPROC
POSIX Thread APIs	*PTHREAD
QNTC File System	*QNTC
Qshell	*QSHELL
Remote Access Services	*RAS
Registration Facility	*REGFAC
Software Security	*SECURITY
Service Facility	*SERVICE
Sockets APIs	*SOCKETS
TCP/IP Configuration	*TCPIPCFG
Thread Management	*THREADMGT
Work Management	*WRKMGT

Element 2: Trace level

Specifies the trace level to be used for the specified component.

***ERROR**
The trace level ERROR will be used. Components typically trace error return codes, exception conditions, and invalid input. The amount of data is usually small.

***INFO**
The trace level INFO will be used, which also includes ERROR trace level data. Components typically trace entry and exit from interfaces, parameters and return codes, and major changes of flow or semantics caused by input or other decisions. The amount of data is moderate.

***VERBOSE**
The trace level VERBOSE will be used, which also includes INFO and ERROR trace level data. Components typically trace detailed data that could assist in debugging control flow, data corruption, data structures, environment, call stacks, and resource allocations. The amount of data can be large.

Top

Trace filter (TRCFTR)

Specifies the trace filter to be used. The Add Trace Filter (ADDTRCFTR) command must be issued to define the trace filter. The filter determines what information is collected in the trace based on the filter values. The filter values are compared to the actual trace data. If a filter is not specified, then all trace information is collected.

The trace filter parameter can only be specified if *ALL or *FLOW has been specified for the **Trace type (JOBTRCTYPE)** parameter. The trace filter applies to the *FLOW (call/return) trace only.

***NONE**
No trace filter will be used.

name Specify the name of the trace filter to be used.

Top

Restart after next IPL (RESTRIPL)

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 RESTRIPL(*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 RESTRIPL(*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.

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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be

compared are TDE number, task name, server 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.

Top

Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

***NOMAX**
There is no time limit for watching for a particular message or LIC log entry.

1-43200
Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

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

```
1   Trace option setting   Input    Char(10)
2   Reserved               Input    Char(10)
3   Error detected         Output   Char(10)
4   Comparison data        Input    Char(*)
```

Allowed values for the ″Trace option setting″ parameter are:

***ON** The watch for trace facility is starting when the collection of trace information is started.

***MSGID**
A match on a message id specified on WCHMSG parameter occurred.

***LICLOG**
A match on a LIC log specified on the WCHLICLOG parameter occurred.

***CMPDATA**
The major and minor code of a LIC log matched, but the comparison data did not.

***INTVAL**
The time interval specified on TRCPGMITV parameter is elapsed.

***WCHTIMO**
The length of time to watch specified on WCHTIMO parameter is elapsed.

The ″Reserved″ parameter must be set to blanks.

Allowed values for the ″Error detected″ parameter are:

***CONTINUE**
The trace and the watch for trace event facility will continue running.

***STOP**
The trace and the watch for trace event facility will be ended.

***ERROR**
Error detected by customer trace program.

Allowed values for the ″Comparison data″ parameter when *MSGID is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(7)      Message ID
11   B     CHAR(9)      Reserved
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      Message comparison data
```

Allowed values for the ″Comparison data″ parameter when *LICLOG or *CMPDATA is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(4)      LIC Log major code
 8   8     CHAR(4)      LIC Log minor code
12   C     CHAR(8)      LIC Log identifier
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      LIC log comparison data
```

Allowed values for the ″Comparison data″ parameter when *ON, *INTVAL or *WCHTIMO is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information (always 4).
```

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.iseries.ibm.com/infocenter .

Single values

***NONE**
No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

Qualifier 1: Trace program

name Specify the name of the trace exit program.

Qualifier 2: Library

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

name Specify the name of the library where the user exit program is located.

Top

Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

***NONE**
No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

1-9999 Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

Examples

Example 1: Start a Trace on Your Job

```
STRTRC   SSNID(*GEN)
```

This command starts a trace on the job that called the STRTRC command. A unique trace session identifier will be generated. The session identifier will be returned in the message data of a completion message sent after the trace session starts successfully.

Example 2: Start a Trace of Another Job

```
STRTRC   JOB(123456/QSYS/QSYSARB)  SSNID(ARBTRACE)
```

This command starts a new trace on the job with job name QSYSARB, job user name of QSYS, and job number of 123456. Only this one job is traced. The trace session identifier is ARBTRACE.

Example 3: Start a Generic Job Trace

```
STRTRC   SSNID(*GEN)  JOB(*ALL/QSYS/QTVTELNET*)  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 (11111111 22222222)))
         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 11111111 and 22222222 are traced. The trace session identifier is MYTRACE.

Example 7: Start a Trace and Watch for a Message to End the Trace

```
STRTRC   SSNID(*GEN)  WCHMSG((MCH2804))
         WCHMSGQ((*SYSOPR) (*JOBLOG))
         WCHJOB((*ALL/MYUSER/MYJOBNAME))
         TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts a trace on the job that called the STRTRC command. The trace will be ended when MCH2804 message is found on the System Operator message queue or within the *ALL/MYUSER/MYJOBNAME job log. Also, MYLIB/TRCEXTPGM is specified as a trace exit program.

Example 8: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
STRTRC   SSNID(*GEN) WCHLICLOG(('99??' 9932 MYJOBNAME))
         WCHTIMO(*NOMAX)
```

This command starts a trace on the job that called the STRTRC command. The trace will be ended when a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 is generated on the system. Also, the LIC log information should contain the text ″MYJOBNAME″. *NOMAX on WCHTIMO parameter indicates that the trace will be active until the event occurs or ENDTRC command is issued manually.

Top

Error messages

***ESCAPE Messages**

CPC3921
STRTRC session ID &1 successfully started.

CPC3922
STRTRC session ID &1 started but some jobs not found.

CPF0006
Errors occurred in command.

CPF39C7
STRTRC session ID &1 already exists.

CPF39C9
Unexpected STRTRC failure, see low-level messages.

CPF39C5
Job name *ALL and user *ALL not allowed

CPF39C6
Not autorized to trace with generic job users.

CPF39CC
No active jobs found, STRTRC session not started.

CPI36CE
Job &3/&2/&1 not found.

CPF39F2
Cannot allocate library &1

CPF98A2
Not authorized to &1 command.

CPF39D3
Unable to start/end the trace.

CPF39D4
Another trace session is already active that specified RESTRIPL(*YES).

Top

Start Trap Manager (STRTRPMGR)

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

Parameters
Examples
Error messages

Use the Start Trap Manager (STRTRPMGR) command to start the OS/400 SNMP trap manager. An optional Forward Trap parameter may be specified, which enables traps that are received on the system to be forwarded to other Network Management stations, as configured in the Simple Network Management Protocol (SNMP) agent and Distributed Protocol Interface (DPI) interface.

Top

Parameters

Keyword	Description	Choices	Notes
FWDTRP	Forward traps	*YES, ***NO**	Optional, Positional 1

Top

Forward Trap (FWDTRP)

Specifies whether traps received on the system are to be forwarded to other network management stations.

The possible values are:

***NO** Received traps will not be forwarded. Traps will only be enqueued.

***YES** Received traps will be forwarded using the facilities provided in the SNMP agent and DPI interface.

Top

Examples

Example 1: Start Trap Manager Job (Enqueue Traps Only)

```
STRTRPMGR
```

This command starts the trap manager job. Traps received by the trap manager are enqueued only.

Example 2: Start Trap Manager Job (Enqueue & Forward Traps)

```
STRTRPMGR   FWDTRP(*YES)
```

This command starts the trap manager job. Traps received by the trap manager are enqueued and forwarded.

Top

© Copyright IBM Corp. 1998, 2006

Error messages

***ESCAPE Messages**

CPFA80D
Trap manager job already active.

CPFA802
Trap manager not started.

Top

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 or LIC log) occurs. 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 Operating System through iSeries 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 Operating System through iSeries 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 operatinal (*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	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
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*, *ALL	
	Qualifier 2: User	*Generic name, name*, *ALL	
	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**	

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 and LIC log 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 occurs.

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.

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, all of them will no longer be watched. CPI3999 message will be sent to the job log and message queue QSYSOPR to indicate an error in the exit program caused the watch session to be ended.

<blanks>
No error detected by watch exit program.

Note: 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, all of them will no longer be watched. CPI3999 message will be sent to the job log and message queue QSYSOPR 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
 *   *     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(2)      Reserved
324  144   BINARY(4)    Offset to comparison data
328  148   BINARY(4)    Length of comparison data
 *    *    CHAR(*)      LIC log comparison data
```

For more information on the watch exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.iseries.ibm.com/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

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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server 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.

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

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

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.

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.

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.

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 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 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	*Simple name*	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
```

© Copyright IBM Corp. 1998, 2006

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)

Threadsafe: No

Parameters
Examples
Error messages

The Transfer Batch Job (TFRBCHJOB) command transfers a batch job to the specified job queue. The job queue does not have to be allocated to an active subsystem at the time of the batch job transfer. The batch job that is transferred is the one in which this command is issued. Routing data and request data can be specified for the batch job when it is transferred. The routing data specified is processed in the subsystem in which the job queue is active. The request data follows other request data for the job. The transferred batch job resumes running the request data following the transfer.

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. 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

© Copyright IBM Corp. 1998, 2006

Parameters

Keyword	Description	Choices	Notes
JOBQ	Job queue	*Qualified object name*	Required, Positional 1
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RTGDTA	Routing data	*Character value*, **QCMDB**, *RQSDTA	Optional
RQSDTA	Request data or command	*Character value*, ***NONE**, *RTGDTA	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	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
PARM	Parameter CL variable names	Values (up to 255 repetitions): *CL variable name*	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.

© Copyright IBM Corp. 1998, 2006

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.

Notes:

1. The job associated with an active subsystem can only find objects that exist in the system auxiliary storage pool (ASP) or in a basic user ASP. Therefore, objects that are referenced by this command should exist in the system ASP or in a basic user ASP.

Top

Parameters

Keyword	Description	Choices	Notes
GRPJOB	Group job	*Name*, ***PRV**, *SELECT	Optional, Positional 1
INLGRPPGM	Initial group program	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Initial group program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SPCENV	Special environment	***DFT**, *INLGRPPGM, *S36, *NONE	Optional
RSTDSP	Restore display	***NO**, *YES	Optional
TEXT	Text 'description'	*Character value*, ***BLANK**	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.

© Copyright IBM Corp. 1998, 2006

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.

© Copyright IBM Corp. 1998, 2006

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	*Qualified object name*	Required, Positional 1
	Qualifier 1: Job queue	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
RTGDTA	Routing data	*Character value*, **QCMDI**, *RQSDTA	Optional
RQSDTA	Request data or command	*Character value*, ***NONE**, *RTGDTA	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.

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 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.

Top

Examples

```
TFRJOB   JOBQ(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.

Top

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.

The possible values are:

***SRC** The job on the current system is suspended, and control is transferred back to the program specified on the System request program prompt (SRQ10PGM parameter) of the Start Pass-Through (STRPASTHR) command on the previous system. When the specified program ends, the target system gets control.

***ALT** The job on the target system is suspended, and control is transferred back to the alternate job on the previous system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.

***HOME**
The job on the target system is suspended, and control is transferred back to the program specified on the System request program prompt (SRQ10PGM parameter) of the Start Pass-Through (STRPASTHR) command on the home system. When the specified program ends, the target system gets control.

***HOMEALT**
The job on the target system is suspended, and control is transferred back to the alternate job on the home system. When control is transferred, the Transfer Job (TFRJOB) command can be used to transfer from the alternate job to the original job, giving control to the target system. Otherwise, when the alternate job ends, the target system gets control.

Top

© Copyright IBM Corp. 1998, 2006

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

© Copyright IBM Corp. 1998, 2006

Trace TCP/IP Route (TRACEROUTE)

Where allowed to run: All environments (*ALL)
Threadsafe: 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	*Character value*	Required, Positional 1
RANGE	Range of hops to probe	*Element list*	Optional
	Element 1: Starting probe TTL (hop limit)	1-255, **1**	
	Element 2: Maximum probe TTL (hop limit)	1-255, **30**	
PROBES	Probes sent per hop	1-64, **3**	Optional
WAITTIME	Response wait time	1-120, **3**	Optional
PKTLEN	Packet length (in bytes)	40-65535, **40**	Optional
OUTPUT	Output	***MSG**, *VERBOSE, *DTAQ	Optional
DTAQ	Data queue	*Qualified object name*	Optional
	Qualifier 1: Data queue	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
ADRVERFMT	Address version format	***CALC**, *IP4, *IP6	Optional
LCLINTNETA	Source IP address	*Character value*, ***ANY**	Optional
RMTPORT	Base remote port	1-65535, **33434**	Optional
NAMELOOKUP	Lookup host names	***YES**, *NO	Optional
PROBEPCL	Probing protocol	***ICMP**, *UDP	Optional
FRAGMENT	Allow fragmentation	***TCPA**, *NO, *YES	Optional

© Copyright IBM Corp. 1998, 2006

Top

Remote system (RMTSYS)

Specifies the remote system name (255 characters) or IP address of the destination system.

character-value
Specify the remote system name or IP address. Either a valid IP Version 4 or IP Version 6 address will be accepted.

Top

Range of hops to probe (RANGE)

Specifies the range of hop systems from which probe responses are expected. Each probe specifies a TTL (Time To Live) integer value. This TTL value is the maximum number of hops the probe can traverse. For example, a probe packet with a TTL of 3 can pass through at most 3 hop systems before the hop system discards the probe and sends information back to the system from which the probe originated.

Element 1 specifies the first TTL value sent in probe packets. Element 2 specifies the last TTL value sent in probe packets. Trace information is generated from each hop system which discards a probe packet because the TTL value in the probe is reached or when the destination system is reached.

Element 1: Starting probe TTL (hop limit)

1 The default starting hop is 1.

1-255 Specify the first hop limit TTL number used for probe packets.

Element 2: Maximum probe TTL (hop limit)

30 The default ending hop is 30.

1-255 Specify the maximum number of hops a probe can traverse to reach the destination system.

Top

Probes sent per hop (PROBES)

Specifies the number of probe packets sent to each hop system for each probe TTL (hop limit) value in the range specified by the RANGE parameter.

3 The default number of probes is three.

1-64 Specify the number of probes to send.

Top

Response wait time (WAITTIME)

Specifies the maximum time, in seconds, to wait for a response from a hop system to each probe.

3 Wait up to 3 seconds for a response.

1-120 Specify the maximum number of seconds to wait for a response.

Top

Packet length (in bytes) (PKTLEN)

Specifies the total length, in bytes, of the IP packet sent for each probe.

40 The probe packet length is 40 bytes.

40-65535
Specify the number of bytes in the probe IP packet.

Top

Output (OUTPUT)

Specifies where the results obtained from sending the probe packets is sent. Information is sent for each hop until the destination system is reached, including hop count, average round-trip time, IP address of the hop and host name of the hop.

***MSG** Results are output as messages sent to the job log of the job in which the command is issued.

***VERBOSE**
Results are output as messages sent to the job log of the job in which the command is issued. All responses received are displayed. Results are not limited to ICMP TIME_EXCEEDED and PORT_UNREACHABLE responses.

***DTAQ**
Results from probes are placed on the data queue specified by the Data Queue (DTAQ) parameter.

Top

Data queue (DTAQ)

Specifies the data queue on which entries are placed. When a data queue is specified, messages are not sent to the job log unless an error occurs.

Each queue entry contains the response to a probe if one was received or indicates that no probe response was received. The specified data queue must have a queue entry length of at least 32 characters and must exist when this command is issued.

Qualifier 1: Data queue

name Specify the name of the data queue.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched.

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

name Specify the name of the library to search.

Top

Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

***CALC**
The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. TRCTCPRTE (TRACEROUTE) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution is used in determining the IP address.

***IP4** Use the IP Version 4 host name resolution method.

***IP6** Use the IP Version 6 host name resolution method.

Top

Source IP address (LCLINTNETA)

Specifies how the source IP address in the probe packet is chosen.

***ANY** The source IP address in the probe packets is chosen by the system. The system may use any active local interface which can reach the remote system.

character-value
Specify the local interface to use as the source IP address.

Top

Base remote port (RMTPORT)

Specifies the base UDP port number used in probes.

33434 Use the default base UDP port number of 33434.

1-65535
Specify the base UDP port number to be used in probes.

Top

Lookup host names (NAMELOOKUP)

Specifies whether IP addresses will be resolved to the host name.

***YES** The address will be resolved to the host name.

***NO** The address will not be resolved to the host name.

Top

Probing protocol (PROBEPCL)

Specifies the protocol used when sending probe packets.

***ICMP**
The probes sent to the destination system are ICMP (Internet Control Message Protocol) Echo Request packets.

***UDP** The probes sent to the destination system are UDP (User Datagram Protocol) packets.

Top

Allow fragmentation (FRAGMENT)

Specifies how the setting of the ″Do Not Fragment″ option in the IP header of the probe packet is determined.

***TCPA**
The system sets the option based on the setting of the IP Path MTU Discovery TCP/IP attribute.

Note: Use the Change TCP/IP Attributes (CHGTCPA) command to change the value of this attribute.

***NO** The ″Do Not Fragment″ option is always specified.

***YES** The ″Do Not Fragment″ option is never specified.

Top

Examples

Example 1: Trace Entire Route

```
TRCTCPRTE   RMTSYS('130.14.3.5')
```

This command traces the entire route between the local 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)  PROBEPCL(*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 futher 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

```
TRCTCPRTE   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

```
TRCTCPRTE   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): *Name*	Optional, Positional 2
TIMLMT	Time limit	1-9999, *NOMAX	Optional

Top

© Copyright IBM Corp. 1998, 2006

Trace option setting (SET)

Specifies whether to start collecting statistics, end the collection of statistics, or delete previously collected usage statistics for an auxiliary storage pool (ASP).

Note: A value must always be specified for this parameter.

***ON** The tracing of the statistics will be started.

***OFF** The tracing of the statistics will be ended.

***CLEAR**
The statistics for the specified ASP will be cleared.

Top

ASP number (ASP)

Specifies the auxiliary storage pool (ASP) for which the ASP tracing function will be started, ended, or cleared.

Note: A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter.

Single values

***ALL** ASP tracing will be started, ended, or cleared for the system ASP (ASP number 1) and all basic ASPs (ASP numbers 2-32) defined to the system.

Other values (up to 32 repetitions)

1-32 Specify the number of the ASP for which ASP tracing is to be started, ended, or cleared.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device for which the tracing function will be started, ended, or cleared.

Note: A value must be specified for either the **ASP number (ASP)** parameter or the **ASP device (ASPDEV)** parameter.

Single values

***ALLAVL**
ASP tracing will be started, ended, or cleared for all ASP devices that currently have a status of 'Available'.

Other values (up to 32 repetitions)

name Specify the name of the independent ASP device for which ASP balancing is to be started.

Top

Time limit (TIMLMT)

Specifies the amount of time, in minutes, that the ASP balancing function will be allowed to run. When the time limit is reached the function will end. The trace function will not run across an IPL.

Note: A value must be specified for this parameter if *ON is specified for the **Trace option setting (SET)** parameter.

1-9999 Specify the number of minutes that the trace function will be allowed to run.

Top

Examples

Example 1: Start Trace for ASP 1

```
TRCASPBAL   ASP(1)  SET(*ON)  TIMLMT(9999)
```

This command allows the user to start the ASP tracing function for ASP 1. This function will run until the user ends the trace or 9999 minutes have passed.

Example 2: End Tracing for All ASPs

```
TRCASPBAL   ASP(*ALL)  SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP that currently has a trace running.

Example 3: Clear the Trace Data for ASP 1

```
TRCASPBAL   ASP(1)  SET(*CLEAR)
```

This command allows the user to clear the trace data for ASP 1.

Example 4: End Tracing for All ASP Devices

```
TRCASPBAL   ASPDEV(*ALLAVL)  SET(*OFF)
```

This command allows the user to end the ASP tracing function for each ASP device that currently has a trace running.

Top

Error messages

***ESCAPE Messages**

CPF1890
*ALLOBJ authority required for requested operation.

CPF18A9
ASP tracing for ASP &1 already started.

CPF18AA
ASP tracing not active for ASP &1.

CPF18AD
ASP &1 must contain more than a single unit.

CPF18AE
ASP &1 does not contain trace data.

CPF18B1
Trace function currently running for ASP &1.

CPF18B2
Balance function running for ASP &1.

CPF9829
Auxiliary storage pool &1 not found.

Top

Trace Connection (TRCCNN)

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

Parameters
Examples
Error messages

The Trace Connection (TRCCNN) command allows the tracing of encrypted data flowing over internet protocol (IP) and Secure Sockets Layer (SSL) connections. Specific types of traces are started and stopped by using this command.

TRCCNN uses the Trace Internal (TRCINT) command to collect the trace records and generate an intermediate spooled file named QPCSMPRT. The QPCSMPRT spooled file data is used to generate a spooled file named QSYSPRT. The user data for the QSYSPRT file is 'TRCCNN'.

You can use 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 Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- The following user profiles have private authorities to use the command:
  - QSRV
- When the **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 Operating System through iSeries 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

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
TRCTYPE	Trace type	Values (up to 2 repetitions): *IP, *SSL	Optional
TRCFULL	Trace full	***WRAP**, *STOPTRC	Optional
TRCTBL	Trace table name	*Character value*, ***GEN**	Optional
SIZE	Size	Single values: *MAX, *MIN Other values: *Element list*	Optional
	Element 1: Number of units	1-998000, **16000**	
	Element 2: Unit of measure	***KB**, *MB	
FMTDTA	Trace data to be formatted	72-99999, ***CALC**	Optional
CCSID	Coded character set identifier	1-65533, ***EBCDIC**, *ASCII	Optional
JOB	Job name	Single values: ***** Other values: *Qualified job name*	Optional
	Qualifier 1: Job name	*Name*	
	Qualifier 2: User	*Name*	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, *ONLY, ***LAST**	Optional
JOBSYSNAME	Job system name	*Name*, ***ONLY**, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: ***ONLY**, *LAST Other values: *Element list*	Optional
	Element 1: Creation date	*Date*	
	Element 2: Creation time	*Time*, ***ONLY**, *LAST	
TCPDTA	TCP/IP data	*Element list*	Optional
	Element 1: Protocol	*TCP, *UDP, *ICMP, *IGMP, *ARP, *ICMP6	
	Element 2: Local port	Values (up to 2 repetitions): 1-65535	
	Element 3: Remote port	Values (up to 2 repetitions): 1-65535	
	Element 4: Local IP address	*Character value*	
	Element 5: Remote IP address	*Character value*	
	Element 6: Line description	*Name*	
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**	

Keyword	Description	Choices	Notes
WCHLICLOG	Watch for LIC log entry	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Major code	*Character value*, *ALL	
	Element 2: Minor code	*Character value*, *ALL	
	Element 3: Comparison data	*Character value*, ***NONE**	
WCHTIMO	Length of time to watch	1-43200, ***NOMAX**	Optional
TRCPGM	Trace program	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Trace program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**	
TRCPGMITV	Time interval	1-9999, ***NONE**	Optional

Top

Trace option setting (SET)

Specifies whether tracing is started, stopped or ended. Also, you can select to format trace record data collected previously using the TRCCNN or TRCINT (Trace Internal) command.

This is a required parameter.

***ON** The collection of internal trace records is started for the trace types specified in the TRCTYPE parameter. If *GEN is specified in the TRCTBL parameter then the trace table name will be QTRCCNNxxxxxx where xxxxxx is the job number of the current job. Otherwise the trace table name will be the name specified on the TRCTBL parameter.

***OFF** Collection of trace records stops. A spooled file named QPCSMPRT is generated by the TRCINT command and contains the collected trace record data. TRCCNN formats this data in a second spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. The trace table is deleted after the spooled files are generated.

***END** Collection of trace records stops and the trace table is deleted. No spooled output is generated.

***FORMAT**
Formats trace data in a QPCSMPRT spooled file created by a previous invocation of TRCCNN or TRCINT. The formatted data is written to a spooled file named QSYSPRT. The user data for the QSYSPRT spooled file is 'TRCCNN'. Use the TRCTYPE parameter to specify which connection-related trace records to format. Use the JOB, SPLNBR, JOBSYSNAME and CRTDATE parameters to identify which QPCSMPRT file to use.

Top

Trace type (TRCTYPE)

If SET(*ON) is specified, identifies the types of trace records to start collecting. If SET(*FORMAT) is specified, identifies the types of collected trace records to format. Multiple trace types may be specified.

***IP** Trace IP (internet protocol) data.

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

Top

Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or stop tracing when the trace table is full.

***WRAP**
When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC**
Tracing is stopped when the trace table is full of trace records.

Top

Trace table name (TRCTBL)

Specifies the trace table to hold the collected trace data.

***GEN** The trace table name will be QTRCCNNxxxxxx where xxxxxx is the job number of the current job.

name Specify the name of the trace table to be used. If SET(*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created.

Top

Size (SIZE)

Specifies the size of the trace table. The amount of storage to be allocated can be specified in units of kilobytes (*KB) or megabytes (*MB). If the size is specified in kilobytes, the amount of storage allocated for the table will be rounded up to the nearest megabyte. Valid table size values range from one megabyte to 258048 megabytes.

Note: The amount of storage specified by this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage space is not dynamically allocated as it is needed. This storage space will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

Note: If tracing data over a gigabit Ethernet line the trace table size should be greater than 128000 kilobytes or 128 megabytes.

Single values

***MAX** The trace table is set to the maximum size of 258048 megabytes.

***MIN** The trace table is set to the minimum size of one megabyte.

Element 1: Number of units

16000 The trace table size is either 16000 kilobytes or 16000 megabytes, depending on the value specified for the second element of this parameter.

1-998000
Specify the size of the trace table in kilobytes or megabytes.

Element 2: Unit of measure

Specifies whether the value specified for the first element should be treated as number of kilobytes or number of megabytes.

***KB** The trace table size is specified in kilobytes.

***MB** The trace table size is specified in megabytes.

Top

Trace data to be formatted (FMTDTA)

Specifies the number of bytes of traced data to be formatted.

***CALC**
The system determines the number of bytes of data to be formatted.

72-99999
Specify the number of bytes of data to be formatted.

Top

Coded character set identifier (CCSID)

Specifies whether the extended binary-coded decimal interchange code (*EBCDIC- 37) or the American National Standard Code for Information Interchange (*ASCII- 819) character code or any other is used for the formated 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 QPCSMPRT 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 QPCSMPRT spooled file.

Qualifier 1: Job name

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

Qualifier 3: Number

000000-999999
Specify the system-assigned job number of the job that created the input QPCSMPRT spooled file.

Top

Spooled file number (SPLNBR)

Specifies the file number of the QPCSMPRT 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 QPCSMPR created by the specified job is used.

***ONLY**
Only one spooled file named QPCSMPRT 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 QPCSMPRT, an error message is issued.

1-999999
Specify the number of the QPCSMPRT 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 (TCPDTA)

Specifies whether a subset of TCP/IP and/or SSL trace data should be collected. Each parameter element is optional; if no element value is specified, no filtering of trace data is done for that element. For example, if *TCP is specified for element 1, only trace records where the TCP protocol is used are collected. If no value is specified for element 1, trace records using all TCP/IP protocols are collected.

Element 1: Protocol

Specify a TCP/IP protocol to be traced.

***TCP** Enable trace for transmission control protocol.

***UDP** Enable trace for user datagram protocol.

***ICMP**
Enable trace for internet control message protocol.

***IGMP**
Enable trace for internet group management protocol

***ARP** Enable trace for address resolution protocol. This will only apply for TCP/IP.

***ICMP6**
Enable trace for internet control message protocol version 6.

Element 2: Local port

1-65535
Specify one or two local port numbers for which trace data is collected.

Element 3: Remote port

1-65535
Specify one or two remote port numbers for which trace data is collected

Element 4: Local IP address

character-value
Specify a local internet protocol address.

Element 5: Remote IP address

character-value
Specify a remote internet protocol address.

Element 6: Line description

name Specify the name of a line description for which TCP/IP trace data is to be collected.

Top

Watch for message (WCHMSG)

Specifies up to five message identifiers which are to be watched for. If a value other than *NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace exit program is called; if no trace exit program is defined, the trace stops.

Single values

***NONE**
No messages will be watched for.

Element 1: Message identifier

name Specify the 7-character message identifier to be watched for.

Element 2: Comparison data

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data, 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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server 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.

Top

Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

***NOMAX**
There is no time limit for watching for a particular message or LIC log entry.

1-43200
Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.

- When the length of time to watch specified on WCHTIMO parameter is reached.

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

```
1   Trace option setting   Input    Char(10)
2   Reserved               Input    Char(10)
3   Error detected         Output   Char(10)
4   Comparison data        Input    Char(*)
```

Allowed values for the ″Trace option setting″ parameter are:

***ON** The watch for trace facility is starting when the collection of trace information is started.

***MSGID**
A match on a message id specified on WCHMSG parameter occurred.

***LICLOG**
A match on a LIC log specified on the WCHLICLOG parameter occurred.

***CMPDATA**
The major and minor code of a LIC log matched, but the comparison data did not.

***INTVAL**
The time interval specified on TRCPGMITV parameter is elapsed.

***WCHTIMO**
The length of time to watch specified on WCHTIMO parameter is elapsed.

The ″Reserved″ parameter must be set to blanks.

Allowed values for the ″Error detected″ parameter are:

***CONTINUE**
The trace and the watch for trace event facility will continue running.

***STOP**
The trace and the watch for trace event facility will be ended.

***ERROR**
Error detected by customer trace program.

Allowed values for the ″Comparison data″ parameter when *MSGID is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(7)      Message ID
11   B     CHAR(9)      Reserved
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      Message comparison data
```

Allowed values for the ″Comparison data″ parameter when *LICLOG or *CMPDATA is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(4)      LIC Log major code
 8   8     CHAR(4)      LIC Log minor code
12   C     CHAR(8)      LIC Log identifier
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      LIC log comparison data
```

Allowed values for the ″Comparison data″ parameter when *ON, *INTVAL or *WCHTIMO is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information (always 4).
```

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.iseries.ibm.com/infocenter .

Single values

***NONE**
No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

Qualifier 1: Trace program

name Specify the name of the trace exit program.

Qualifier 2: Library

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

name Specify the name of the library where the user exit program is located.

Top

Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

***NONE**
No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

1-9999 Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

Examples

Example 1: Starting SSL Traces

```
TRCCNN   SET(*ON)  TRCTYPE(*SSL)
```

This command starts tracing for Secure Sockets Layer (SSL) connections.

Example 2: Starting IP Traces

```
TRCCNN   SET(*ON)  TRCTYPE(*IP)
```

This command starts tracing for connections at the internet protocol (IP) level.

Example 3: Stopping Traces and Clearing Trace Storage

```
TRCCNN   SET(*END)
```

This command stops all traces and deletes the trace table. No spooled output is generated.

Example 4: Printing Traces

```
TRCCNN   SET(*OFF)
```

This command stops all traces and generates a spooled file (QPCSMTRC) that contains the trace records collected by the TRCINT (Trace Internal) command, and a spooled file (QSYSPRT) that contains the formatted trace data.

Example 5: Formatting Trace Data from TRCINT Command

```
TRCINT   SET(*ON)  TRCTYPE(*SCK)
TRCINT   SET(*OFF)
TRCCNN   SET(*FORMAT)  TRCTYPE(*SSL)  JOB(*)  SPLNBR(*LAST)
```

The TRCINT (Trace Internal) commands are used to start collecting trace records related to all usage of sockets, and to stop collecting trace records and create a spooled file named QPCSMPRT. The TRCCNN command will use the trace information in the last spooled file named QPCSMPRT for the current job, and format the trace records related to SSL (Secure Sockets Layer) in a spooled file named QSYSPRT.

Example 6: Specifying a Trace Table

```
TRCCNN   SET(*ON)  TRCTYPE(*IP)  TRCTBL(USER)
```

This command starts tracing for connections at the internet protocol (IP) level and stores the trace data in the USER trace table.

Example 7: Specifying a Trace Table Size in Megabytes

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

This command starts tracing for connections at the internet protocol (IP) level and stores the data in a 20000-megabyte trace table.

Example 8: Specifying a CCSID for Trace Data

```
TRCCNN   SET(*OFF)  CCSID(*ASCII)
```

This command stops all traces and generates a spooled file (QSYSPRT). ASCII (819) CCSID will be used when formatting the trace data.

Example 9: Start a Trace and Watch for a Message to End the Trace

```
TRCCNN   SET(*ON) TRCTYPE(*IP)  WCHMSG((MCH2804))
         WCHMSGQ((*SYSOPR) (*JOBLOG))
         WCHJOB((*ALL/MYUSER/MYJOBNAME))
         TRCPGM(MYLIB/TRCEXTPGM)
```

This command starts tracing for connections at the internet protocol (IP) level. The trace will be ended when MCH2804 message is found on the System Operator message queue or within the *ALL/MYUSER/MYJOBNAME job log. Also, MYLIB/TRCEXTPGM is specified as a trace exit program.

Example 10: Start a Trace and Watch for a LIC Log Entry to End the Trace

```
TRCCNN   SET(*ON) TRCTYPE(*IP)
         WCHLICLOG(('99??' 9932 MYJOBNAME))
         WCHTIMO(*NOMAX)
```

This command starts tracing for connections at the internet protocol (IP) level. The trace will be ended when a Licensed Internal Code (LIC) log entry that has a major code starting with 99 and a minor code of 9932 is generated on the system. Also, the LIC log information should contain the text "MYJOBNAME". *NOMAX on WCHTIMO parameter indicates that the trace will be active until the event occurs or 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)
Threadsafe: No

Parameters
Examples
Error messages

The Trace Common Programming Interface Communications (TRCCPIC) command controls tracing of all CPI Communications that occur in the job in which the command is entered. The command sets a trace on or off, and traces (1) CPI Communications calls issued by a program and (2) data that is sent and received.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCCPIC command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restrictions: (1) The record format of the database output file must match the record format of the IBM-supplied output file, QACM0TRC. (2) The user must have specific authority from the security officer to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	***ON**, *OFF, *END	Optional, Positional 1
MAXSTG	Maximum storage to use	1-16000, **200**	Optional, Positional 2
TRCFULL	Trace full	***WRAP**, *STOPTRC	Optional, Positional 3
DTALEN	User data length	0-4096, **128**	Optional, Positional 4
OUTPUT	Output	***PRINT**, *OUTFILE	Optional
OUTFILE	Output file	*Qualified object name*	Optional
	Qualifier 1: Output file	*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	

Top

© Copyright IBM Corp. 1998, 2006

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.

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:

***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.

Top

Examples

Example 1: Starting Trace Operation

```
TRCCPIC   MAXSTG(350)  DTALEN(256)
```

This command traces the CPI Communications calls of the current job. The trace file contains 350KB of storage and wraps to the beginning if that amount of storage is filled with trace records. In addition, this command traces up to 256 bytes of user data on each input/output operation.

Example 2: Stopping Trace Operation

```
TRCCPIC   SET(*OFF)  OUTPUT(*OUTFILE)  OUTFILE(TRACELIB/CPICTRACE)
          OUTMBR(TRACEMBR)
```

This command stops the trace and directs the output to the database file CPICTRACE in library TRACELIB. The output is directed to the member TRACEMBR.

Top

Error messages

***ESCAPE Messages**

CPF2C90
Maximum storage specified too small.

CPF2C94
Error occurred during OUTFILE processing. Trace stopped.

CPF3B30
No CPI-Communications calls were run. Trace ended.

CPF3B31
Job is already being serviced or traced.

CPF3B32
Trace already off.

CPF3B33
Unexpected Trace CPI Communications error occurred.

CPF3B34
Cannot deactivate trace, trace started from another job.

CPF3548
Serviced job completed running.

CPF3936
Job being serviced ended before trace started.

CPF9847
Error occurred while closing file &1 in library &2.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

Top

Trace ICF (TRCICF)

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

Parameters
Examples
Error messages

The Trace Intersystems Communications Functions (TRCICF) command is used to start and stop the tracing of language operations and Intersystem Communications Functions (ICF) issued by your program. TRCICF can be started from the command entry display or from a CL program.

As trace records are collected, they are stored in an internal trace storage area. When the trace is ended, the trace records can be directed to a spooled output file or a database physical file.

If the Start Service Job (STRSRVJOB) command is entered before the TRCICF command, the job that is traced is the one specified on the STRSRVJOB command. The trace output from the serviced job is returned to the servicing job after the trace is set off or after the serviced job has ended.

Restrictions:

1. The record format of the database output file must match the record format of the IBM-supplied output file, QAIFTRCF.
2. The user must have specific authority from the security officer to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
SET	Trace option setting	***ON**, *OFF, *END	Optional, Positional 1
MAXSTG	Maximum storage to use	1-16000, **200**	Optional, Positional 2
TRCFULL	Trace full	***WRAP**, *STOPTRC	Optional, Positional 3
DTALEN	User data length	0-4096, **128**	Optional, Positional 4
OUTPUT	Output	***PRINT**, *OUTFILE	Optional
OUTFILE	Output file	*Qualified object name*	Optional
	Qualifier 1: Output file	*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	

Top

Trace option setting (SET)

Specifies whether an ICF trace is started, stopped, or ended.

© Copyright IBM Corp. 1998, 2006

***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 is 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 Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
2. The following user profiles have private authorities to use the command:
   - QSRV
3. When the **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 Operating System through iSeries 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	*Character value*, ***SYSDFT**	Optional
SIZE	Trace table size	Single values: ***NOCHG**, *MAX, *MIN Other values: *Element list*	Optional
	Element 1: Number of units	1-998000	
	Element 2: Unit of measure	***KB**, *MB	
TRCFULL	Trace full	***NOCHG**, *WRAP, *STOPTRC	Optional, Positional 4

© Copyright IBM Corp. 1998, 2006

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, *IDXMGT, *IFS, *IPCF, *ISCSI, *ISDN, *JRNMGT, *LNKTST, *LODDMP, *MCHOBS, *MITFMALL, *MITFMEVAL, *MITFMHEAP, *MITFMINT, *MITFMMI, *MITFMSTG, *MODMGT, *MOD2, *MSCP, *MSMCALL, *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, *QMGT, *QSMGT, *RCYMGT, *RMTSPT, *RSCMGT, *SCK, *SCKASCIO, *SCKNET, *SCKOTHER, *SCKOVRMPTN, *SCKOVRPEC, *SCKRSLV, *SCKSEL, *SCKSSL, *SCKSTDIO, *SIG, *SMBSVR, *SPCOBJMGT, *SRCSINK, *STGMGTALL, *STM, *SYNCMGT, *SYSCALL, *SYSJRNMGT, *TCPIP, *TLI, *TRXMGT, *VRTDEVMGT, *VRTIO	Optional, Positional 2
JOB	Job name	Single values: ***NOCHG** Other values (up to 10 repetitions): *Qualified job name*	Optional
	Qualifier 1: Job name	*Generic name, name*, *ALL	
	Qualifier 2: User	*Generic name, name*, *ALL	
	Qualifier 3: Number	000000-999999, *ALL	
SLTTHD	Thread ID to include	Single values: ***NOCHG**, *ALL, *SELECT Other values (up to 20 repetitions): *Hexadecimal value*	Optional
SVRTYPE	Server type	Single values: *ALL, ***NOCHG** Other values (up to 5 repetitions): *Generic name, name*	Optional
TASK	Task name	Single values: *ALL, ***NOCHG** Other values (up to 10 repetitions): *Generic name, name*	Optional
TASKNBR	Task number	Single values: *ALL, ***NOCHG** Other values (up to 10 repetitions): *Hexadecimal value*	Optional
SLTTRCPNT	Select trace points	Single values: ***NOCHG** Other values (up to 5 repetitions): *Element list*	Optional
	Element 1: Trace point qualifier	0-65535	
	Element 2: To trace point qualifier	0-65535	
OMTTRCPNT	Omit trace points	Single values: ***NOCHG** Other values (up to 5 repetitions): *Element list*	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: ***NOCHG** Other values (up to 4 repetitions): *Element list*	Optional
	Element 1: Trace point type	*Character value*	
	Element 2: Trace point qualifier	0-65535	
	Element 3: Trace point entry	1-65535	
	Element 4: Trace point entry offset	*Hexadecimal value*	
	Element 5: Trace point match value	*Character value*	
JOBTYPE	Job types	Single values: *NONE, *ALL Other values (up to 12 repetitions): ***DFT**, *ASJ, *BCH, *EVK, *INT, *MRT, *RDR, *SBS, *SYS, *WTR, *PDJ, *PJ, *BCI	Optional
JOBTRCITV	Job trace interval	0.1-9.9, **0.5**	Optional
TCPDTA	TCP/IP data	*Element list*	Optional
	Element 1: Protocol	*TCP, *UDP, *ICMP, *IGMP, *ARP, *ICMP6	
	Element 2: Local port	Values (up to 2 repetitions): 1-65535	
	Element 3: Remote port	Values (up to 2 repetitions): 1-65535	
	Element 4: Local IP address	*Character value*	
	Element 5: Remote IP address	*Character value*	
	Element 6: Line description	*Name*	
	Element 7: Line type	*PPP, *OPC	
SCKDTA	Sockets data	*Element list*	Optional
	Element 1: Address family	*INET, *UNIX, *INET6, *NETBIOS	
	Element 2: Socket type	*STREAM, *DGRAM, *RAW, *SEQPACKET	
	Element 3: Descriptor	Values (up to 2 repetitions): 0-65535	
	Element 4: Socket option	*SODEBUG	
DEV	Device	Single values: ***NONE** Other values (up to 16 repetitions): *Name*	Optional, Positional 3
CTL	Controller	Single values: ***NONE** Other values (up to 16 repetitions): *Element list*	Optional
	Element 1: Controller	*Name*	
	Element 2: Attached devices	***NODEV**, *ALLDEV	
LIN	Line	Single values: ***NONE** Other values (up to 16 repetitions): *Element list*	Optional
	Element 1: Line	*Name*	
	Element 2: Attached controllers	***NOCTL**, *ALLCTL	
NWI	Network interface	Single values: ***NONE** Other values (up to 16 repetitions): *Name*	Optional
NWS	Network server	Single values: ***NONE** Other values (up to 16 repetitions): *Name*	Optional
RSRCNAME	Resource name	Single values: ***NONE** Other values (up to 10 repetitions): *Name*	Optional
OUTDEV	Output device	*Name*	Optional
TASKINF	Task information	*ALL, ***TRCREF**	Optional
OUTPUT	Output	***PRINT**, *OUTFILE	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**	
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

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 QPCSMPRT or sent to a database file as indicated by **Output (OUTPUT)** parameter.

***END** Internal tracing ends and the internal trace records are deleted. No spooled output is generated. If a trace table name other than *SYSDFT is specified for the TRCTBL parameter, it will be automatically deleted.

***HOLD**
Internal traces are stopped, and the collected internal trace records are held in the trace table. Held records can be printed later if another Trace Internal (TRCINT) command is entered that specifies *OFF for this prompt; or, the held records can be put on tape or optical media if *SAVE is specified.

***SAVE**
Internal traces are stopped, and the trace records are written to a tape or optical device specified by the OUTDEV parameter.

***SIZE** The size of the trace table is changed. The new size is specified on the **Trace table size (SIZE)** parameter.

Top

Trace table name (TRCTBL)

Specifies the trace table to hold the collected trace data.

***SYSDFT**
The system default trace table is used.

name Specify the name of the trace table to be used. If SET(*ON) is specified and the name specified does not match an existing trace table, a new trace table by the specified name will be created.

Top

Trace table size (SIZE)

Specifies the size of the trace table. This parameter can be specified only when *SIZE is specified for the **Trace option setting (SET)** parameter or if SET(*ON) is specified and tracing is not currently active for the trace table specified (TRCTBL parameter).

Note: The storage indicated on this parameter is immediately allocated from the system auxiliary storage pool (ASP 1). This storage is not dynamically allocated as it is needed. This storage space will not be available for use by the system except to record trace-related information. Before specifying a large value on this parameter, the amount of free space in the system ASP should be checked. Use the Work with System Status (WRKSYSSTS) command to determine the amount of available free space in the system ASP. System performance degradation may result if the size of the free space in the system ASP is significantly reduced as a result of the value specified.

Note: If tracing data over a gigabit Ethernet line the trace table size should be greater than 128000 kilobytes or 128 megabytes.

Single values

***NOCHG**
The trace table size is not changed. If a new trace table is specified (TRCTBL parameter), a default size of 128 kilobytes will be used.

***MAX** The trace table is set to the maximum size of 258048 megabytes.

***MIN** The trace table is set to the minimum size of 128 kilobytes.

Element 1: Number of units

Specify the size of the trace table.

1-998000
Specify the size of the trace table in kilobytes or megabytes.

Element 2: Unit of measure

Specifies whether the value specified for the first element should be treated as number of kilobytes or number of megabytes.

***KB** The trace table size is specified in kilobytes. The valid range is 128 through 998000.

***MB** The trace table size is specified in megabytes. The valid range is 1 through 258048.

Top

Trace full (TRCFULL)

Specifies whether the trace records wrap (replace the oldest records with new records) or tracing stops when the trace table is full. This parameter can be specified when *ON is specified for the **Trace option setting (SET)** parameter.

***NOCHG**
The trace table full action is not changed. If a new trace table is specified (TRCTBL parameter), the default action is for trace records to wrap when the trace table becomes full.

***WRAP**
When the trace table is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC**
Tracing is stopped when the trace table is full of trace records.

Top

Trace type (TRCTYPE)

Specifies the type of traces to start. The two groups of trace types are:

- Component data trace codes cause active procedures to be traced within the system.
- General trace codes cause the iSeries instruction supervisor linkage, multiprogramming, transaction, or task and thread performance functions to be traced.

If trace code types are specified, *ON must be specified for the **Trace option setting (SET)** parameter. If a value other than *ON is specified for the SET parameter, TRCTYPE is ignored. Each trace type is identified by a special value or 6-digit code; all 6 digits must be specified. For a complete list of trace codes and special values, position the cursor on this parameter while prompting this command and press F4. Specify up to 50 types from the following code table:

TYPE OF TRACE	COMPONENT OR GENERAL TRACE CODE	SPECIAL VALUE
MI instruction supervisor linkage (SVL)	030000	*SVL
Multiprogramming level (MPL)	040000	*MPL
Transaction	080000	*TNS
Task/thread performance	090000	*TTPERF
Activation/Call	014400	*ACTCALL
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	*JRNMGT
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	015101	*MITFMEVAL

evaluation		
MI Transformer - heap operations	015102	*MITFMHEAP
MI Transformer - interpreter instructions	015103	*MITFMINT
MI Transformer - MI Instructions	015104	*MITFMMI
MI Transformer - storage management operations	015105	*MITFMSTG
Module management	013100	*MODMGT
Modula-2 run time support	012800	*MOD2
Machine services control point	010802	*MSCP
Main storage management -- calls	011102	*MSMCALL
Main storage management -- details	011103	*MSMDTL
NetBios on TCP/IP	015700	*NTBTCP
OptiConnect	015500	*OPC
Portable Application Solutions Environment	016100	*PASE
Performance collection services	016200	*PFRCOLSRV
Program binder	013200	*PGMBND
Program management	010300	*PGMMGT
Portability utilities	015200	*PORTUTIL
Process management	010500	*PRCMGT
Process table	015300	*PRCTBL
Pseudo terminal component	016500	*PSEUDOTERM
Power management	012900	*PWRMGT
Queue management	010700	*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 PEC	014202	*SCKOVRPEC
Sockets Berkeley Resolver APIs	014005	*SCKRSLV
Sockets Select API	014003	*SCKSEL
Secure Sockets Layer (SSL) APIs	014006	*SCKSSL
Sockets Standard Input/Output APIs	014001	*SCKSTDIO
Signals	015600	*SIG
SMB server	015800	*SMBSVR
Space object management	011500	*SPCOBJMGT
Source/sink (device support) management	010803	*SRCSINK
Storage management -- (all)	011105	*STGMGTALL
Streams	013900	*STM
Synchronization management	013700	*SYNCMGT
MI system call	015200	*SYSCALL
System journal management	014100	*SYSJRNMGT
Transmission Control Protocol/ Internet Protocol (TCP/IP)	013800	*TCPIP
Transport Layer Interface	016000	*TLI
Transaction Management	016800	*TRXMGT
Virtual I/O	016700	*VRTIO
Virtual terminal management	013000	*VRTDEVMGT

Note: If *TTPERF trace code is specified, it must be the first one in the TRCTYPE parameter list.

Top

Job name (JOB)

Specifies the jobs from which trace records are to be collected. Only trace records which are generated in the specified job(s) are collected. A list of up to ten qualified job names can be specified. The trace records will be collected if they were generated from a job that matches any of the qualified job name values.

This parameter can be specified when *ON is specified for the **Trace option setting (SET)** parameter. If a value other than *ON is specified on the SET parameter, JOB is ignored.

Single values

***NOCHG**
If any qualified job names had been specified for the JOB parameter on a previous TRCINT command for an active trace, the job name filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, *NOCHG will behave the same as *ALL/*ALL/*ALL for the JOB parameter.

Qualifier 1: Job name

***ALL** All trace records generated by the defined trace are collected, regardless of what job name the trace record was generated from.

generic-name
Specify the generic name of the job from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic job name specifies all jobs with job name that begin with the generic prefix.

name Specify the name of the job from which trace records are to be collected.

Qualifier 2: User

***ALL** All trace records generated by the defined trace are collected, regardless of what job user name the trace record was generated from.

generic-name
Specify the generic user name of the job from which trace records are to be collected. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic user name specifies all jobs with user names that begin with the generic prefix.

name Specify the name of the user of the job from which trace records are to be collected.

Qualifier 3: Number

***ALL** All trace records generated by the defined trace are collected, regardless of what job number the trace record was generated from. *ALL for the job number is considered to be a generic job specification because it will trace all jobs that meet the job name and job user name qualifiers that you specified.

000000-999999
Specify the job number to further qualify the job name and user name. You cannot specify a job number if a generic job name or generic user name is specified.

Top

Thread ID to include (SLTTHD)

Specifies the job threads from which trace records are to be collected. Only trace records which are generated in the specified thread(s) are collected. There can be only one job that has thread IDs associated with it. It must be the first qualified job name specified for the JOB parameter, and the job must be active.

Single values

***NOCHG**
If any thread identifiers had been specified for the SLTTHD parameter on a previous TRCINT command for an active trace, the thread ID filtering information is not changed. If no value was specified on a previous TRCINT command for an active trace, *NOCHG will behave the same as *ALL for this parameter.

***ALL** All trace records generated by the defined trace are collected, regardless of what thread ID the trace record was generated from.

***SELECT**
A list of thread identifiers is shown from which you can select up to twenty threads. Trace records from any of the selected thread identifiers are to be collected. *SELECT is only valid if the TRCINT command is run in an interactive job.

Other values

hexadecimal-value
Specify the identifier of the thread from which trace records are to be collected. Up to twenty thread identifiers can be specified.

Top

Server type (SVRTYPE)

Specifies the server type attribute for a job or task which is used to determine whether the trace record is collected. Only trace records which are generated in a job or task with the specified server type are collected. For a list of possible server types, see Work Management information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/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.

Top

Hardware Resource (RSRCNAME)

Specifies the names of the hardware resources for which the associated internal events are traced. This parameter can be specified only if 014700 or 011900 is specified on the **Trace type (TRCTYPE)** parameter.

Note: A combined total of 10 hardware resources (RSRCNAME) are allowed.

Single values

***NONE**
No hardware resources are traced by this command.

Other values

name Specify the names of up to 10 hardware resources for which the internal trace is started.

Top

Device (OUTDEV)

Specifies the tape device or optical device on which the held trace records are written. This parameter must be specified if *SAVE is specified for the **Trace option setting (SET)** parameter.

name Specify the name of the device description of the tape or optical device.

Top

Task information (TASKINF)

Specifies whether information for all licensed internal code (LIC) tasks is written to a spooled file or output device. This parameter can be specified if SET(*OFF) or SET(*SAVE) is specified.

***TRCREF**
Write information only for LIC tasks that were referenced by trace records in the specified trace table.

***ALL** Write information for all LIC tasks that were in existence while the trace was active.

Top

Output (OUTPUT)

Specifies whether the output from the command is printed with the job's spooled output or sent to a database file.

***PRINT**
The output is printed with the job's spooled output.

***OUTFILE**
The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**
The current library for the thread is used to locate the file. If no library is specified as the current library for the thread, the QGPL library is used.

name Specify the name of the library to be searched.

Note: If a new file is created, system file QASCTRCI in system library QSYS is used as a model.

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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server 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.

Top

Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a licensed internal code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

***NOMAX**
There is no time limit for watching for a particular message or LIC log entry.

1-43200
Specify the number of minutes that the trace will remain active while none of the watched for conditions have been met.

Top

Trace program (TRCPGM)

Specifies the program to be called for user-defined trace commands and procedures.

The trace program will be called:

- Before the application trace starts.
- After a match of a message identifier specified for the WCHMSG parameter, or a match of a Licensed Internal Code (LIC) log entry specified for the WCHLICLOG parameter occurs.
- When the time interval specified on the TRCPGMITV parameter is reached.
- When the length of time to watch specified on WCHTIMO parameter is reached.

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

```
1   Trace option setting   Input    Char(10)
2   Reserved               Input    Char(10)
3   Error detected         Output   Char(10)
4   Comparison data        Input    Char(*)
```

Allowed values for the ″Trace option setting″ parameter are:

***ON** The watch for trace facility is starting when the collection of trace information is started.

***MSGID**
A match on a message id specified on WCHMSG parameter occurred.

***LICLOG**
A match on a LIC log specified on the WCHLICLOG parameter occurred.

***CMPDATA**
The major and minor code of a LIC log matched, but the comparison data did not.

***INTVAL**
The time interval specified on TRCPGMITV parameter is elapsed.

***WCHTIMO**
The length of time to watch specified on WCHTIMO parameter is elapsed.

The ″Reserved″ parameter must be set to blanks.

Allowed values for the ″Error detected″ parameter are:

***CONTINUE**
The trace and the watch for trace event facility will continue running.

***STOP**
The trace and the watch for trace event facility will be ended.

***ERROR**
Error detected by customer trace program.

Allowed values for the ″Comparison data″ parameter when *MSGID is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(7)      Message ID
11   B     CHAR(9)      Reserved
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      Message comparison data
```

Allowed values for the ″Comparison data″ parameter when *LICLOG or *CMPDATA is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information
 4   4     CHAR(4)      LIC Log major code
 8   8     CHAR(4)      LIC Log minor code
12   C     CHAR(8)      LIC Log identifier
20  14     BINARY(4)    Offset to comparison data
24  18     BINARY(4)    Length of comparison data
 *   *     CHAR(*)      LIC log comparison data
```

Allowed values for the ″Comparison data″ parameter when *ON, *INTVAL or *WCHTIMO is specified for the ″Trace option setting″ parameter will be the following structure:

```
OFFSET     TYPE         FIELD
Dec Hex
 0   0     BINARY(4)    Length of trace information (always 4).
```

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.iseries.ibm.com/infocenter .

Single values

***NONE**
No trace exit program is defined. If a watched for message or licensed internal code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

Qualifier 1: Trace program

name Specify the name of the trace exit program.

Qualifier 2: Library

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

name Specify the name of the library where the user exit program is located.

Top

Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

***NONE**
No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

1-9999 Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

Examples

Example 1: Starting Component Data Traces and Call Traces

```
TRCINT   SET(*ON)  TRCTYPE(010100 010400 050500 051200)
```

This command starts component data traces and call traces for resource management and database. Database operations associated with database files are used to collect component data trace records.

Example 2: Tracing Lines and Controllers

```
TRCINT   SET(*ON)  TRCTYPE(*SRCSINK) TRCTBL(*SYSDFT)
         DEV(WS1 WS2 WS3)  CTL((C1) (C2))  LIN((L1) (L2))
```

This command starts component data traces for source/sink management (device support) operations involving the devices WS1, WS2, and WS3, lines L1 and L2, and controllers C1 and C2.

Example 3: Stopping Traces and Clearing Trace Table

```
TRCINT   SET(*END)  TRCTBL(*SYSDFT)
```

This command stops all traces and deletes the trace records from the system default trace table.

Example 4: Tracing Communications Trace Service Function

```
TRCINT   SET(*ON)  TRCTYPE(*CMNTRC)
```

This command starts component data traces for the communications trace service function.

Example 5: Using Job Filtering Capability

```
TRCINT   SET(*ON)  TRCTBL(MYFTPTRACE)
         TRCTYPE(*TCPIP)  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
TCPDTA only valid if *TCPIP or *SMBSVR or *FRCA or one or more of the socket TRCTYPE is specified.

CPD3687
SCKDTA only valid if one or more of the socket TRCTYPE values is specified.

CPD3688
Job types (JOBTYPE) parameter only valid with TRCTYPE(*MPL).

CPD3689
Job trace interval (JOBTRCITV) parameter only valid with TRCTYPE(*MPL)

CPD368A
Cannot change trace point selection criteria for active trace table.

CPD36C0
OUTDEV parameter only valid with SET(*SAVE).

CPD36C1
SIZE parameter only valid with SET(*ON) or SET(*SIZE).

CPD36CD
TASKINF parameter only valid with SET(*OFF) or SET(*SAVE).

CPD3983
Range of parameter SIZE not valid.

CPD3990
User number qualifier not valid.

CPD3991
Job, thread identifier or task not active.

CPF3515
Too many trace requests or objects.

CPF3516
Trace table is full.

CPF3517
Cannot specify *SELECT for the thread ID to include.

CPF3518
End time and date earlier than start time and date.

CPF3659
Total of specified CTL, DEV, LIN, NWI, and NWS greater than allowed.

CPF3679
Service function returned completion code &1 qualifier &2.

CPF3683
Error occurred trying to open printer file.

CPF3684
Error occurred while trying to close a print file.

CPF3685
Error occurred while data being put to print file.

CPF3686
Service function ended with error message.

CPF3687
Error occurred while trying to open file.

CPF3688
Error occurred while tape or optical file being closed.

CPF3689
Error occurred while writing data to tape or optical device.

CPF368A
Trace table size not changed.

CPF3692
Error occurred while trying to write data to tape or optical device.

CPF3693
Service function ended because error occurred.

CPF3694
Cannot start service function.

CPF3695
No trace tables exist.

CPF3696
No traces recorded.

CPF3697
Trace type parameter value missing.

CPF7A11
Trace table &1 not found.

CPF7A13
Trace table cannot be created.

CPF7A15
Trace buffer must be cleared.

CPF7A17
Trace already is active.

CPF7A1A
Cannot change trace point selection criteria for active trace table.

CPF7A1C
IP address not valid.

CPF98A2
Not authorized to &1 command.

Top

Trace Job (TRCJOB)

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

Parameters
Examples
Error messages

The Trace Job (TRCJOB) command controls traces of original program model (OPM) programs and Integrated Language Environment (ILE) procedure calls and returns that occur in the current job or in the job being serviced as a result of the Start Service Job (STRSRVJOB) command directed to that job. The command, which sets a trace on or off, can trace module flow, operating system data acquisition (including CL command traces), or both.

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	***ON**, *OFF, *END	Optional, Positional 1
TRCTYPE	Trace type	***ALL**, *FLOW, *DATA	Optional, Positional 2
MAXSTG	Maximum storage to use	1-16000, **4096**	Optional, Positional 3
TRCFULL	Trace full	***WRAP**, *STOPTRC	Optional, Positional 4
EXITPGM	Program to call before trace	Single values: ***NONE** Other values: *Qualified object name*	Optional
	Qualifier 1: Program to call before trace	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
SLTPRC	Select procedures to trace	Single values: ***ALL**, *NONE Other values (up to 50 repetitions): *Element list*	Optional
	Element 1: Program	*Qualified object name*	
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
	Element 2: Type	***PGM**, *SRVPGM	
SLTTHD	Thread ID to include	Single values: ***ALL**, *SELECT Other values (up to 20 repetitions): *Hexadecimal value*	Optional
OUTPUT	Output	***PRINT**, *OUTFILE	Optional
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	

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.

Top

Examples

Example 1: Tracing Flow of Control

```
TRCJOB   TRCTYPE(*FLOW)  MAXSTG(40)
```

This command traces the flow of the current job. Trace records are collected for each OPM program and ILE procedure call and return that occurs in the job. The trace file contains 40K of storage and wraps (oldest records are replaced by new records) if that amount of storage is filled with trace records.

Example 2: Stopping the Trace Operation

```
TRCJOB   SET(*OFF)  OUTPUT(*OUTFILE)
         OUTFILE(QGPL/TRCJOB)  OUTMBR(TRCDTA)
```

This command stops the trace and directs the output to the database file QGPL/TRCJOB. The output is directed to the member TRCDTA.

Example 3: Tracing One Thread

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

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

Top

Error messages

***ESCAPE Messages**

CPF2C94
Error occurred during OUTFILE processing. Trace stopped.

CPF2C95
Trace already active.

CPF2C96
Trace already off.

CPF3510
User exit program not found in specified library.

CPF3511
Trace already active.

CPF3512
Trace already off.

CPF3513
Cannot set Trace Off, trace started from another job.

CPF3521
Not enough storage for the trace table.

CPF3530
Conflicting entries in index QSERVICE.

CPF3542
Job not traced because it is being serviced.

CPF3548
Serviced job completed running.

CPF3675
Cannot allocate QSYS library.

CPF3909
Service command will not be processed.

CPF3918
Service request canceled.

CPF3925
Cannot open file &1.

CPF3936
Job being serviced ended before trace started.

CPF3950
Error message &2 received for file &1. Request ended.

CPF3951
File &1 cannot be overridden by file name &2.

CPF3957
Not authorized to use exit program library &2.

CPF3969
Error during close of file &1. Output may not be complete.

CPF6611
Error occurred during OUTFILE processing, trace ended.

CPF6801
Command prompting ended when user pressed &1.

CPF9810
Library &1 not found.

CPF98A4
Thread restrictions exist for some other process.

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.

The possible values are:

***RESULTS**
All clauses are traced before processing. Tracing operates as if the TRACE ?R instruction was used from within the REXX procedure.

***ALL** All clauses are traced before processing. Tracing operates as if the TRACE ?A instruction was used from within the REXX procedure.

***COMMANDS**
All host commands are traced before processing and any error return code is displayed. Tracing operates as if the TRACE ?C instruction was used from within the REXX procedure.

***ERROR**
Any host command resulting in an error return code is traced after processing. Tracing operates as if the TRACE ?E instruction was used from within the REXX procedure.

***FAILURE**
Any host command resulting in a failure is traced after processing together with the return code from the command. Tracing operates as if the TRACE ?F instruction was used from within the REXX procedure.

***INTERMEDIATES**
All clauses are traced before processing. Intermediate results during evaluation of expressions and substituted names are also traced. Tracing operates as if the TRACE ?I instruction was used from within the REXX procedure.

***LABELS**
Labels passed during processing are traced. Tracing operates as if the TRACE ?L instruction was used from within the REXX procedure.

© Copyright IBM Corp. 1998, 2006

***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, 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 iSeries Navigator's Application Administration support. For a given application, there could be only one trace active at a time on the system. The user must have *USE authority to the line, network interface, or network server to be traced.

When the **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 iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_ALLOBJ_TRACE_ANY_USER, can also be used to change the list of users that are allowed to perform trace operations.

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 iSeries 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

© Copyright IBM Corp. 1998, 2006

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	Required, Positional 1
SET	Trace option setting	***ON**, *OFF, *END, *CHK	Optional, Positional 2
MAXSTG	Maximum storage for trace	1-16000, ***APP**	Optional, Positional 3
TRCFULL	Trace full action	***WRAP**, *STOPTRC	Optional, Positional 4
ADLTRC	Additional traces	Single values: ***NONE** Other values (up to 11 repetitions): *CMNARB, *CMNTRC, *DEVD, *DEVMGR, *JOB, *SOCKETS, *SRCSINK, *SYSARB, *TCPIP, *USER, *USERDEVD	Optional, Positional 5
TRCPGM	Trace program	Single values: ***NONE** Other values: *Qualified object name*	Optional, Positional 6
	Qualifier 1: Trace program	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
TITLE	Trace title	*Character value*, ***DFT**	Optional, Positional 7
USER	User profile	*Simple name*	Optional, Positional 8
MAILADR	Recipient mail address	*Character value*	Optional, Positional 9
HOST	Recipient host name	*Character value*	Optional, Positional 10
RMTNETADR	Remote network address	*Element list*	Optional, Positional 11
	Element 1: Address family	***INET**	
	Element 2: IP address	*Character value*	
	Element 3: Subnet mask	*Character value*, **'255.255.255.255'**	
	Element 4: Port number	1-65535, ***ANY**	
LCLNETADR	Local network address	*Element list*	Optional, Positional 12
	Element 1: Address family	***INET**, *UNIX	
	Element 2: IP address or UNIX path	*Character value*	
	Element 3: Subnet mask	*Character value*, **'255.255.255.255'**	
	Element 4: Port number	1-65535, ***ANY**	
DEVD	Device description	*Generic name, name*	Optional, Positional 13
DEVTYPE	Device type	Single values: *DSP, *PRT Other values (up to 6 repetitions): 5251, 5291, 5292, 3196, 3488, 3487, 3179, 3180, 5555, 3477, 3277, 3278, 3279, V100, 3812, 5553	Optional, Positional 14
TRCPNT	Trace point	Values (up to 12 repetitions): *Character value*	Optional, Positional 15
ARGLIST	Argument list	*Character value*	Optional, Positional 16
VPNSVR	Virtual private network server	Values (up to 2 repetitions): *KEYMGR, *CNNMGR	Optional, Positional 17

Keyword	Description	Choices	Notes
CERTTYPE	Certificate services type	***ALL**, *DCM, *KEYMGR, *SSL, *OBJSIGN, *OTHER	Optional, Positional 18
DNS	Domain name service	***NO**, *YES	Optional, Positional 19
PPPCNNPRF	PPP connection profile	*Character value*	Optional, Positional 20
TCPTRCDTA	TCP/IP data to trace	***PPPALL**, *LCPNCP	Optional, Positional 21
QOSTRCTYPE	QOS trace type	***ALL**, *POLICYD, *RSVPD	Optional, Positional 22
HTTPSVR	HTTP server instance	*Character value*	Optional, Positional 23
TRCLVL	Trace level	***ERROR**, *INFO, *VERBOSE	Optional, Positional 24
PKTTRCPNT	Packet rules trace points	***TRAFFIC**, *LOAD	Optional, Positional 25
CFGOBJ	Configuration object	*Name*	Optional, Positional 26
CFGTYPE	Type	*LIN, *NWI, *NWS	Optional, Positional 27
WCHMSG	Watch for message	Single values: ***NONE** Other values (up to 5 repetitions): *Element list*	Optional, Positional 28
	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, Positional 29
	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, Positional 30
	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, Positional 31
	Element 1: Major code	*Character value*, *ALL	
	Element 2: Minor code	*Character value*, *ALL	
	Element 3: Comparison data	*Character value*, ***NONE**	
WCHTIMO	Length of time to watch	1-43200, ***NOMAX**	Optional, Positional 32
TRCPGMITV	Time interval	1-9999, ***NONE**	Optional, Positional 33

Keyword	Description	Choices	Notes
JOB	Jobs	Single values: * Other values (up to 16 repetitions): *Element list*	Optional, Positional 34
	Element 1: Job name	*Qualified job name*	
	Qualifier 1: Job name	*Generic name, name*, *ALL	
	Qualifier 2: User	*Generic name, name*, *ALL	
	Qualifier 3: Number	000001-999999, ***ALL**	

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.

Top

Trace option setting (SET)

Specifies whether the collection of trace information starts, stops, or status is presented.

***ON** The collection of trace information is started.

***OFF** The collection of trace information is stopped and the trace information is written to spooled printer files of the user. For PPP traces, the trace files are also included in the OUTQ for the designated PPP profile.

***END** Tracing is ended and all trace information is deleted. No trace information output is created.

***CHK** The status of tracing for the specified application is checked. Messages are returned indicating whether or not tracing is active for the specified TCP/IP application, the command parameters specified from the last time that TRCTCPAPP was started for this application and other information related to the collection of trace information.

Top

Maximum storage for trace (MAXSTG)

Specifies the maximum amount of storage in kilobytes (K) used for collected trace information.

***APP** Each application type defines a default buffer size.

- *FTP - 4096K bytes per job
- *SMTPCLT - 4096K bytes per job
- *SMTPSVR - 4096K bytes per job
- *TELNET - 16000K bytes per job
- *VTAPI - 16000K bytes per job
- *CENTRAL - 16000K bytes per job
- *RMTCMD - 16000K bytes per job
- *SIGNON - 16000K bytes per job
- *DTAQ - 16000K bytes per job
- *NETPRT - 16000K bytes per job
- *SVRMAP - 16000K bytes per job
- *DATABASE - 16000K bytes per job
- *DDM - 16000K bytes per job

- *VPN - 16000K bytes per job
- *PKTRULES - 16000K bytes per job
- *L2TP - 4096K bytes per job
- *CERTSRV - 16000K bytes per job
- *PPP - 4096K bytes per job
- *QOS - 4096K bytes per job
- *NTP - 4096K bytes per job
- *HTTP - 16000K bytes per job
- *DIRSRV - 300K bytes per job

1-16000
Specify the maximum amount of storage, in kilobytes, used to store trace records (one kilobyte 1024 bytes).

Top

Trace full action (TRCFULL)

Specifies whether the trace records wrap (replace oldest records with new records) or whether the trace stops when all of the storage specified by the MAXSTG parameter has been used.

***WRAP**
When the trace buffer is full, the trace wraps to the beginning. The oldest trace records are written over by new ones as they are collected.

***STOPTRC**
Tracing stops when the trace buffer is full of trace records.

Top

Additional traces (ADLTRC)

Specifies additional trace(s) to be started. When the TRCTCPAPP command is invoked interactively, the user will be prompted for any options to change on each of the selected traces. This parameter is valid for all applications.

Note: Specifying *CMNARB, *DEVD, *DEVMGR, *JOB, *SYSARB, *USER or *USERDEVD 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 iSeries Information Center at http://www.eserver.iseries.com/infocenter.

***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.

***DEVD**
All jobs associated with the device specified for the Device description (DEVD) 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 iSeries Information Center at http://www.eserver.iseries.com/infocenter.

***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.

***USERDEVD**
All user jobs associated with the device specified for the Device description (DEVD) 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 or WCHLICLOG parameter is specified, the trace program will be called:
  - After a match of a message identifier specified on WCHMSG parameter or LIC log specified on WCHLICLOG parameter occurred.
  - When the time interval specified on the TRCPGMITV parameter is reached.
  - When the length of time to watch specified on WCHTIMO parameter is reached.

For SET(*OFF), the trace program will be called:
- Before the LIC traces, if requested, end.
- After the communications trace, if requested, ends.
- After the application trace ends.

For SET(*END), the trace program will be called:
- After the LIC and communications traces, if requested, end.
- After the application trace ends.

When the TRCTCPAPP CPP detects an error with the trace program, it will display the TCP4537 diagnostic message. If watch for trace event facility is active, the trace and the watch for trace event facility will be ended, and CPI3999 message will be sent to the system operator message queue with reason code 04.

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

- 1 Trace option setting Input Char(10)
- 2 Application Input Char(10)
- 3 Error detected Output Char(10)
- 4 Comparison data Input Char(*)

The possible values for the **Trace option setting (SET)** parameter are:

- *ON, The collection of trace information is started.
- *OFF, The collection of trace information is stopped and the trace information is written to spooled printer files of the user.
- *END, Tracing is ended and all trace information is deleted. No trace information output is created.
- *MSGID, A match on a message ID specified on WCHMSG parameter occurred.
- *LICLOG, A match on a LIC log specified on the WCHLICLOG parameter occurred.
- *CMPDATA, The major and minor code of a LIC log matched, but the comparison data did not.
- *INTVAL, The time interval specified on TRCPGMITV parameter is elapsed.
- *WCHTIMO, The length of time to watch specified on WCHTIMO parameter is elapsed.

The possible values for the ″Application″ parameter are the same as the values for the APP parameter on the TRCTCPAPP command.

The possible values for the ″Error detected″ parameter are:

- *ERROR, Error detected by customer trace program.
- *CONTINUE, The trace and the watch for trace event facility will continue running.
- *STOP, The trace and the watch for trace event facility will be ended.

The possible values for the ″Comparison data″ parameter when *MSGID is specified on ″Trace option setting″ parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information

4 4 CHAR(7) Message ID

11 B CHAR(9) Reserved

20 14 BINARY(4) Offset to comparison data

24 18 BINARY(4) Length of comparison data

* * CHAR(*) Message comparison data

The possible values for the ″Comparison data″ parameter when *LICLOG or *CMPDATA is specified on ″Trace option setting″ parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information

4 4 CHAR(4) LIC Log major code

8 8 CHAR(4) LIC Log minor code

12 C CHAR(8) LIC Log identifier

20 14 BINARY(4) Offset to comparison data

24 18 BINARY(4) Length of comparison data

* * CHAR(*) LIC log comparison data

The possible values for the ″Comparison data″ parameter when *ON, *OFF, *END, *INTVAL or *WCHTIMO is specified on ″Trace option setting″ parameter will be the following structure:

Offset Type Field

Dec Hex

0 0 BINARY(4) Length of trace information (always 4 at this time).

For more information on the trace exit program interface, refer to the System API Reference information in the iSeries Information Center at http://www.ibm.com/eserver/iseries/infocenter.

Single values

***NONE**
No user supplied trace program will be called. If a watched for message or Licensed Internal Code (LIC) log entry is added, or if the specified watch time limit is exceeded, the trace function ends.

Qualifier 1: Trace program

name Specify the name of the trace program to be called.

Qualifier 2: Library

***LIBL** The library list is used to locate the program.

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

trace-program-library
Specify the name of the library where the program is located.

Top

Trace title (TITLE)

Specifies the title that is printed on each page of the spooled file which contains the collected trace information. This parameter is only valid when SET(*OFF) is specified.

***DFT** The default trace description title ″TRCTCPAPP Output″ is used.

character-value
Specify up to 50 characters to be used as the title on each page of the trace output spooled file.

Top

User profile (USER)

Only trace information associated with a specific user profile will be collected. This parameter is only valid when APP(*FTP) is specified, 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

***INET** The filter for AF_INET address family.

Element 2: IP address

character-value
Specify the remote TCP/IP address for which trace information is to be collected.

Element 3: Subnet mask

255.255.255.255
Tracing will be done for only the IP address specified as the second element of this parameter.

character-value
Specify the subnet mask for which trace information is to be collected.

Element 4: Port number

***ANY** The TCP/IP port number defaults to *ANY which implies traffic associated with any port on the remote system (and qualified by the IP address and subnet mask) will be traced.

1-65535
Specify the port number to be used. If a number is specified, a subnet mask value must also be specified.

Top

Local network address (LCLNETADR)

The user may limit the amount of information collected by entering an address family, local TCP/IP address, subnet mask and port number. This parameter is only valid when APP(*DDM), APP(host server), APP(*TELNET) or APP(*VTAPI) is specified.

Element 1: Address family

***INET** The filter for AF_INET address family.

***UNIX**
The filter for AF_UNIX address family. Note that *UNIX is a valid choice for only APP(*DDM) or APP(host server).

Element 2: IP address or UNIX path

character-value
When *INET is specified for element 1 of this parameter, specify the local TCP/IP address for which trace information is to be collected.

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

Element 3: Subnet mask

255.255.255.255
Tracing will be done for only the IP address specified as the second element of this parameter.

character-value
Specify the subnet mask for which trace information is to be collected.

Element 4: Port number

***ANY** The TCP/IP port number defaults to *ANY which implies traffic associated with any port on the local system (and qualified by the IP address and subnet mask) will be traced.

1-65535
Specify the port number to be used. If a number is specified, a subnet mask value must also be specified.

Top

Device description (DEVD)

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, TGVTIN, TGVTINI, TGVTM or TGVTOUT.

For host/DDM server trace points, specify 'Qcccxxxx' where 'ccc' is the component ID of the host/DDM server and 'xxxx' defines the specific trace point.

Top

Argument list (ARGLIST)

Only trace information associated with this specific argument list will be included in the trace information collected. The argument list contains data like the debug level and special trace requests. This parameter is only valid when APP(*VPN), APP(*QOS), APP(*PKTRULES), APP(*PPP) or APP(*DIRSRV) is specified.

character-value
Specify the argument list (up to 255 characters).

QOS allows the following argument list values:

lvl=1 The lvl=1 argument logs errors that are associated with system operations. One example might be that the system is out of memory. The result of these types of errors is that the QoS server will not be able to run.

lvl=2 The lvl=2 argument includes all lvl=1 information. In addition, the lvl=2 argument logs internal errors identified with the operation of the QOS server. The usual cause of these types of errors is that unexpected errors have been encountered in a server operation. A lvl=2 error is considered a condition for an APAR.

lvl=3 The lvl=3 argument includes all lvl=1 and lvl=2 information. In addition, the lvl=3 argument logs the basic operational activities of the QoS server. Examples might be the loading of rules or the sending of a STRTCPSVR SERVER(*QOS) RESTART(*QOS) command.

lvl=4 The lvl=4 argument includes all lvl=1, lvl=2 and lvl=3 information. In addition, the lvl=4 argument logs all traced activities of the QoS server.

Top

Virtual private network server (VPNSVR)

Specifies whether the trace information is to be collected for the VPN key manager or the VPN connection manager. If no value is specified for this parameter, trace information is to be collected for both the VPN key manager and the VPN connection manager. This parameter is only valid when APP(*VPN) is specified.

***KEYMGR**
Filtering of trace information is done to include the VPN key manager.

***CNNMGR**
Filtering of trace information is done to include the VPN connection manager.

Top

Certificate services type (CERTTYPE)

Only trace information associated with a specific certificate services type will be included in the captured trace information. This parameter is only valid when APP(*CERTSRV) is specified.

***ALL** No filtering of trace information is done for certificate services type.

***DCM** Filtering of trace information is done to include only the DCM certificate services type.

***KEYMGR**
Filtering of trace information is done to include only the VPN key manager certificate services type.

***SSL** Filtering of trace information is done to include only the SSL certificate services type.

***OBJSIGN**
Filtering of trace information is done to include only the OBJSIGN certificate services type.

***OTHER**
Filtering of trace information is done to include only a certificate services type not listed above.

Top

Domain name service (DNS)

Specifies whether only trace information associated with domain name service (DNS) resolution will be collected. This parameter is only valid when APP(*SMTPCLT) is specified.

***NO** No filtering of trace information is done for DNS resolution.

***YES** Trace information includes only trace points associated with DNS resolution.

Top

PPP connection profile (PPPCNNPRF)

Trace information associated with a specific PPP connection profile will be collected. The default trace information provided is one joblog and one connection dialog spooled file (containing the PPP dialog trace) for each connection started by the PPP connection profile, one copy of the PPP profile settings, and one copy of the line description used by the profile. When selected by the user there could also be one SRCSINK component trace per connection, one Communications trace per connection and a single TCPIP component trace. This parameter is required when APP(*PPP) is specified.

character-value
Specify the PPP connection profile for which trace information is to be collected.

Top

TCP/IP data to trace (TCPTRCDTA)

Specifies what additional data will be collected when ADLTRC(*TCPIP) is selected. This parameter is only valid when APP(*PPP) is specified. If APP(*PPP) is specified and ADLTRC(*TCPIP) is not specified, this parameter will be ignored.

***PPPALL**
All data on the PPP connection will be traced.

***LCPNCP**
Only LCP and NCP data on the PPP connection will be traced.

Top

QOS trace type (QOSTRCTYPE)

Only trace information associated with a specific QOS trace type will be included in the trace information collected. This parameter is only valid when APP(*QOS) is specified.

***ALL** Filtering of trace information is done to include both servers.

***POLICYD**
Filtering of trace information is done to include the QOS policy server.

***RSVPD**
Filtering of trace information is done to include the RSVP (Resource reSerVation Protocol) server.

Top

HTTP server instance (HTTPSVR)

This parameter will determine which HTTP server instance to trace. It is only valid and required when APP(*HTTP) is specified.

Top

Trace level (TRCLVL)

Specifies the level of granularity of the service trace. This parameter is only valid when APP(*HTTP) is specified.

***ERROR**
The service trace will contain trace records for all error return codes or exception conditions.

***INFO**
The service trace will contain *ERROR level trace records as well as trace records for entry and exit points from application level APIs and API parameters.

***VERBOSE**
The service trace will contain *INFO level trace records as well as trace records for debugging control flow or data corruption.

Top

Packet rules trace points (PKTTRCPNT)

Specifies a keyword that represents trace point values which will be displayed when the Trace Internal (TRCINT) panel is displayed. This parameter is only valid when APP(*PKTRULES) and ADLTRC(*TCPIP) are specified.

***TRAFFIC**
The following trace points for packet filter evaluation will be displayed on the Trace Internal panel: 8110-8111, 8120-8123 and 8420.

***LOAD**
The following trace points for the audit and load of rules will be displayed on the Trace Internal panel: 8100-8105 and 8430-8438.

Top

Configuration object (CFGOBJ)

Specifies the configuration object to trace. The object can be either a line description, a network interface description, or a network server description. This parameter is only valid when ADLTRC(*CMNTRC) is specified.

name Specify the name of the configuration object to be traced.

Top

Type (CFGTYPE)

Specifies the type of configuration description to trace. This parameter is only valid when ADLTRC(*CMNTRC) is specified.

***LIN** The type of configuration object is a line description.

***NWI** The type of configuration object is a network interface description.

***NWS** The type of configuration object is a network server description.

Top

Watch for message (WCHMSG)

Specifies up to five message identifiers which are to be watched for. If a value other than *NONE is specified, you must specify where to watch for the message on the WCHMSGQ parameter. When the watched for message is added to the specified message queue or log, the trace exit program is called; if no trace exit program is defined, the trace stops.

Single values

***NONE**
No messages will be watched for.

Element 1: Message identifier

name Specify the 7-character message identifier to be watched for.

Element 2: Comparison data

Specify comparison data to be used if a message matching the specified message ID is added to the specified message queue or log. If the message data, 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 fields compared of a watched for log entry, the watch condition is considered to be true. This text is case sensitive. The LIC log fields which can be compared are TDE number, task name, server type, job name, user ID, job number, thread ID, exception ID, LIC module compile binary timestamp, LIC module offset, LIC module RU name, LIC module name, LIC module entry point name. The comparison data cannot be used to match across two fields, and can match an entire field or a substring of any field.

When watching for an exception ID, all four hexadecimal digits of the exception ID must be specified. Also, the prefix MCH may be specified if you want to compare only against the exception ID field and avoid possible substring matches with the other fields.

Top

Length of time to watch (WCHTIMO)

Specifies the time limit, in minutes, for watching for a message or a Licensed Internal Code (LIC) log entry. When the specified amount of time has elapsed, the trace exit program is called (if one was specified on the TRCPGM parameter), the trace is ended, and message CPI3999 is sent to the system operator message queue.

***NOMAX**
There is no time limit for watching for a particular message or LIC log entry.

1-43200
Specify the number of minutes that the trace will be active while none of the watched for conditions have been met. You can specify up to 43200 minutes (30 days).

Top

Time interval (TRCPGMITV)

Specifies how often the trace exit program will be called.

***NONE**
No time interval is specified. The trace exit program will not be called because a time interval has elapsed.

1-9999 Specify the interval of time, in seconds, of how often the trace exit program will be called. This must be less than the amount of time specified for the **Length of time to watch (WCHTIMO)** parameter.

Top

Job name (JOB)

Specifies which jobs are to be traced. This parameter is only valid 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(*INET
                      '9.130.69.22'
                      '255.255.255.255' 8471)
            ADLTRC(*CMNTRC)  TRCPGM(PGMLIB/PROG1)
            CFGOBJ(TESTLIN)  CFGTYPE(*LIN)
```

This command will start tracing for the database host server. Tracing information associated with the AF_INET address family, a local TCP/IP address of 9.130.69.22, a subnet mask of 255.255.255.255, port number of 8471 and trace points of QZDA1050 and QZDA1060 will be collected. A communication trace will be included in the trace information. TESTLIN is the name of the configuration object to trace. This object is a line (*LIN) description. Trace program PROG1 in library PGMLIB, with its user-defined trace commands and procedures, will be called. Tracing for the other TCP applications is not affected.

Example 2: Check Status of Database Tracing

```
TRCTCPAPP   APP(*DATABASE)  SET(*CHK)
```

This command is used to check the status of the tracing for the database host server job. Assume that the last command entered was from ″Example 1″ above. The format of the response to this command would be a set of messages that would look similar to the following:

```
TCP45B7 TRCTCPAPP APP(*DATABASE) SET(*ON)
                  TRCPNT(QZDA1050 QZDA1060)
                  LCLNETADR(*INET '9.130.69.22'
                            '255.255.255.255' 8471)
                  MAXSTG(*DFT) TRCFULL(*WRAP) ADLTRC(*CMNTRC)
                  TRCPGM(PGMLIB/PROG1) CFGOBJ(TESTLIN)
                  CFGTYPE(*LIN)
TCP45B1 Tracing active for *DATABASE at 20:15:14 on 03/15/01 by
        043432/TRCUSER/QPADEV000B.
TCP45B2 Data capture begun for *DATABASE.
TCP45B3 Data buffer wrapped for *DATABASE.
```

Example 3: Ending Database Connection Tracing

```
TRCTCPAPP   APP(*DATABASE)  SET(*OFF)
```

This command first ends any currently active application trace for the database host server, followed by ending the TCP/IP component trace. If tracing was active, output trace records are formatted and placed into a spool file. A similar message will be found in the user's joblog:

```
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 4: Starting Trace for Packet Rules

```
TRCTCPAPP   APP(*PKTRULES)  SET(*ON)
            ARGLIST('DebugLvl=1 TraceLvl=2')
            ADLTRC(*TCPIP)  PKTTRCPNT(*LOAD)
```

This command will start tracing for packet rules. Tracing information associated with the specific argument list will be collected. A component trace will be included in the trace information, using trace points of 8100-8105 and 8430-8438. Tracing for the other TCP applications is not affected.

Example 5: Starting Trace for FTP

```
TRCTCPAPP   APP(*FTP)  SET(*ON)
            RMTNETADR(*INET '9.130.69.16' '255.255.255.255' 5)
```

This command will start tracing for the FTP server. Tracing information associated with the AF_INET address family, a remote TCP/IP address of 9.130.69.16, a subnet mask of 255.255.255.255 and port number of 5 will be collected. Tracing for the other TCP applications is not affected.

Example 6: Starting Trace for TELNET

```
TRCTCPAPP   APP(*TELNET)  SET(*ON)  DEVD(QPADEV*)
```

This command will start tracing for the TELNET server. Trace information will be collected for all device descriptions with names that begin with ″QPADEV″. The user must have authority to these specific device descriptions. Tracing for the other TCP applications is not affected.

Example 7: 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.

Top

Error messages

***ESCAPE Messages**

TCP4595

Trace not started.

Top

Trace TCP/IP Route (TRCTCPRTE)

Where allowed to run: All environments (*ALL)
Threadsafe: 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	*Character value*	Required, Positional 1
RANGE	Range of hops to probe	*Element list*	Optional
	Element 1: Starting probe TTL (hop limit)	1-255, **1**	
	Element 2: Maximum probe TTL (hop limit)	1-255, **30**	
PROBES	Probes sent per hop	1-64, **3**	Optional
WAITTIME	Response wait time	1-120, **3**	Optional
PKTLEN	Packet length (in bytes)	40-65535, **40**	Optional
OUTPUT	Output	***MSG**, *VERBOSE, *DTAQ	Optional
DTAQ	Data queue	*Qualified object name*	Optional
	Qualifier 1: Data queue	*Name*	
	Qualifier 2: Library	*Name*, *CURLIB, ***LIBL**	
ADRVERFMT	Address version format	***CALC**, *IP4, *IP6	Optional
LCLINTNETA	Source IP address	*Character value*, ***ANY**	Optional
RMTPORT	Base remote port	1-65535, **33434**	Optional
NAMELOOKUP	Lookup host names	***YES**, *NO	Optional
PROBEPCL	Probing protocol	***ICMP**, *UDP	Optional
FRAGMENT	Allow fragmentation	***TCPA**, *NO, *YES	Optional

© Copyright IBM Corp. 1998, 2006

Top

Remote system (RMTSYS)

Specifies the remote system name (255 characters) or IP address of the destination system.

character-value
Specify the remote system name or IP address. Either a valid IP Version 4 or IP Version 6 address will be accepted.

Top

Range of hops to probe (RANGE)

Specifies the range of hop systems from which probe responses are expected. Each probe specifies a TTL (Time To Live) integer value. This TTL value is the maximum number of hops the probe can traverse. For example, a probe packet with a TTL of 3 can pass through at most 3 hop systems before the hop system discards the probe and sends information back to the system from which the probe originated.

Element 1 specifies the first TTL value sent in probe packets. Element 2 specifies the last TTL value sent in probe packets. Trace information is generated from each hop system which discards a probe packet because the TTL value in the probe is reached or when the destination system is reached.

Element 1: Starting probe TTL (hop limit)

1 The default starting hop is 1.

1-255 Specify the first hop limit TTL number used for probe packets.

Element 2: Maximum probe TTL (hop limit)

30 The default ending hop is 30.

1-255 Specify the maximum number of hops a probe can traverse to reach the destination system.

Top

Probes sent per hop (PROBES)

Specifies the number of probe packets sent to each hop system for each probe TTL (hop limit) value in the range specified by the RANGE parameter.

3 The default number of probes is three.

1-64 Specify the number of probes to send.

Top

Response wait time (WAITTIME)

Specifies the maximum time, in seconds, to wait for a response from a hop system to each probe.

3 Wait up to 3 seconds for a response.

1-120 Specify the maximum number of seconds to wait for a response.

Top

Packet length (in bytes) (PKTLEN)

Specifies the total length, in bytes, of the IP packet sent for each probe.

40 The probe packet length is 40 bytes.

40-65535
Specify the number of bytes in the probe IP packet.

Top

Output (OUTPUT)

Specifies where the results obtained from sending the probe packets is sent. Information is sent for each hop until the destination system is reached, including hop count, average round-trip time, IP address of the hop and host name of the hop.

***MSG** Results are output as messages sent to the job log of the job in which the command is issued.

***VERBOSE**
Results are output as messages sent to the job log of the job in which the command is issued. All responses received are displayed. Results are not limited to ICMP TIME_EXCEEDED and PORT_UNREACHABLE responses.

***DTAQ**
Results from probes are placed on the data queue specified by the Data Queue (DTAQ) parameter.

Top

Data queue (DTAQ)

Specifies the data queue on which entries are placed. When a data queue is specified, messages are not sent to the job log unless an error occurs.

Each queue entry contains the response to a probe if one was received or indicates that no probe response was received. The specified data queue must have a queue entry length of at least 32 characters and must exist when this command is issued.

Qualifier 1: Data queue

name Specify the name of the data queue.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched.

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

name Specify the name of the library to search.

Top

Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

***CALC**
The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. TRCTCPRTE (TRACEROUTE) will first use IP Version 4 host name resolution in determining the IP address. If that fails, IP Version 6 host name resolution is used in determining the IP address.

***IP4** Use the IP Version 4 host name resolution method.

***IP6** Use the IP Version 6 host name resolution method.

Top

Source IP address (LCLINTNETA)

Specifies how the source IP address in the probe packet is chosen.

***ANY** The source IP address in the probe packets is chosen by the system. The system may use any active local interface which can reach the remote system.

character-value
Specify the local interface to use as the source IP address.

Top

Base remote port (RMTPORT)

Specifies the base UDP port number used in probes.

33434 Use the default base UDP port number of 33434.

1-65535
Specify the base UDP port number to be used in probes.

Top

Lookup host names (NAMELOOKUP)

Specifies whether IP addresses will be resolved to the host name.

***YES** The address will be resolved to the host name.

***NO** The address will not be resolved to the host name.

Top

Probing protocol (PROBEPCL)

Specifies the protocol used when sending probe packets.

***ICMP**
The probes sent to the destination system are ICMP (Internet Control Message Protocol) Echo Request packets.

***UDP** The probes sent to the destination system are UDP (User Datagram Protocol) packets.

Top

Allow fragmentation (FRAGMENT)

Specifies how the setting of the ″Do Not Fragment″ option in the IP header of the probe packet is determined.

***TCPA**
The system sets the option based on the setting of the IP Path MTU Discovery TCP/IP attribute.

Note: Use the Change TCP/IP Attributes (CHGTCPA) command to change the value of this attribute.

***NO** The ″Do Not Fragment″ option is always specified.

***YES** The ″Do Not Fragment″ option is never specified.

Top

Examples

Example 1: Trace Entire Route

```
TRCTCPRTE   RMTSYS('130.14.3.5')
```

This command traces the entire route between the local 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)  PROBEPCL(*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 futher 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

```
TRCTCPRTE   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

```
TRCTCPRTE   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

Remove Mounted FS (UNMOUNT)

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

Parameters
Examples
Error messages

The Remove Mounted File System (UNMOUNT) command will make a previously mounted file inaccessible within the integrated file system name space. The file system to be made inaccessible can be a user-defined file system (*UDFS) on the local system, a remote file system accessed through a Network File System server (*NFS), or a local or remote NetWare file system (*NETWARE). If any of the objects in the file system are in use, the command will return an error message to the user. Note that if any part of the file system has itself been mounted over, then this file system cannot be unmounted until it is uncovered.

This command can also be issued using the following alternative command name:

- RMVMFS

For more information about Network File System commands, see Network File System book, SC41-5714

Restrictions:

1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS, *NETWARE, *ALL	Required, Positional 1
MNTOVRDIR	Directory mounted over	*Path name*, *ALL	Optional
MFS	Mounted file system	*Path name*	Optional

Top

Type of file system (TYPE)

Specifies the type of file system to be unmounted.

***NFS** The file system to be unmounted is a Network File System. When *NFS is specified, a directory must be specified for the **Directory mounted over (MNTOVRDIR)** parameter.

***UDFS**
The file system to be unmounted is a user-defined file system. When *UDFS is specified, either the MNTOVRDIR or the **Mounted file system (MFS)** parameter may be specified.

***NETWARE**
The file system to be unmounted is a NetWare file system. When *NETWARE is specified, a directory must be specified for the MNTOVRDIR parameter.

***ALL** File systems of all types are to be unmounted. If *ALL is specified, a value must be specified for the MNTOVRDIR parameter, and that value may be *ALL.

© Copyright IBM Corp. 1998, 2006

This is a required parameter.

Top

Directory mounted over (MNTOVRDIR)

Specifies the path name of the directory that was mounted over ('covered') by a previous ADDMFS (Add Mounted File System) or MOUNT command.

'directory-path-name'
The specified directory that was previously mounted over will be uncovered. If TYPE(*ALL) was specified, all file systems mounted over the specified directory will be unmounted. If a specific file system type was specified for the **Type of file system (TYPE)** parameter, the file system mounted most recently over the specified directory will be unmounted only if it matches the specified TYPE value.

***ALL** All directories that were previously mounted over will be uncovered. If *ALL is specified, *ALL must be specified for the TYPE parameter.

This is a required parameter.

Top

Mounted file system (MFS)

Specifies the path name of the file system to be unmounted. This parameter can only be used to unmount a Block Special File (*BLKSF), when *UDFS is specified for the **Type of file system (TYPE)** parameter.

Top

Examples

The alternative command name for UNMOUNT is RMVMFS. The following examples use the alternative command name, but UNMOUNT can be replaced directly for RMVMFS in all of them.

Example 1: Unmounting a Directory

```
RMVMFS   TYPE (*NFS)  MNTOVRDIR('/tools')
```

This command unmounts a Network File System that is accessible on directory */tools*.

Example 2: Unmounting a User-Defined File System

```
RMVMFS   TYPE(*UDFS)  MFS('/DEV/QASP02/CUST1UDFS')
```

This command unmounts the user-defined file system */dev/qasp02/custudfs*.

Top

Error messages

***ESCAPE Messages**

CPFA0A9
Object not found. Object is &1.

CPFA1B8
*IOSYSCFG authority required to use &1.

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	*Qualified object name*	Optional, Positional 1
	Qualifier 1: Data base file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
MBR	Member	*Name*, ***FIRST**	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.

© Copyright IBM Corp. 1998, 2006

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)
Threadsafe: No

Parameters
Examples
Error messages

The Update Program (UPDPGM) command can be used to replace modules of an Integrated Language Environment (ILE) bound program with other modules on the system, without requiring you to change or recompile the bound program. Modules being replaced must be module objects (*MODULE) on the system.

Other jobs running the bound program can run while the program is being updated with this command. The currently running bound program is moved to library QRPLOBJ and an updated version of the bound program is inserted into the library of the bound program. Current activations of the program will continue running, using the version of the program in the QRPLOBJ library.

Restrictions:

- You must have use (*USE) and add (*ADD) authorities to the library of the bound program.
- You must have *USE, object management (*OBJMGT), and object existence (*OBJEXIST) authorities to the bound program.
- You must be the owner of the bound program, or a member of a group who is the owner of the bound program, or be a user with all object (*ALLOBJ) special authority.
- You must have *USE authority to the following:
  - *MODULE objects specified on the **Module (MODULE)** parameter, and execute (*EXECUTE) authority to the library that the module resides in.
  - *SRVPGM objects specified on the **Bind service program (BNDSRVPGM)** parameter.
  - *BNDDIR objects specified on the **Binding directory (BNDDIR)** parameter, and *EXECUTE authority to the library, and all objects used to resolve external symbols for these *BNDDIR objects, and their libraries.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***USRLIBL**, *CURLIB	
MODULE	Module	Single values: *NONE Other values (up to 300 repetitions): *Qualified object name*	Required, Positional 2
	Qualifier 1: Module	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
RPLLIB	Replacement library	Single values: ***ONLY**, *FIRST, *MODULE Other values: *Qualifier list*	Optional
	Qualifier 1: Replacement library	*Name*	

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
BNDSRVPGM	Bind service program	Single values: ***NONE** Other values (up to 300 repetitions): *Qualified object name*	Optional
	Qualifier 1: Bind service program	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**	
SRVPGMLIB	Bound *SRVPGM library name	Single values: ***SAME**, *LIBL Other values: *Qualifier list*	Optional
	Qualifier 1: Bound *SRVPGM library name	*Name*	
BNDDIR	Binding directory	Single values: ***NONE** Other values (up to 300 repetitions): *Qualified object name*	Optional
	Qualifier 1: Binding directory	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
ACTGRP	Activation group	*Name*, ***SAME**	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	***NONE**, *BASIC, *EXTENDED, *FULL	Optional

Top

Program (PGM)

Specifies the bound program that is to be updated.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the bound program that is to be updated.

Qualifier 2: Library

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

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

name Specify the name of the library where the bound program is located.

Top

Module (MODULE)

Specifies the names of the existing *MODULE objects that are to replace the modules of the same name in the bound program. If two or more modules of the bound program have the same name, the **Replacement library (RPLLIB)** parameter indicates which is to be replaced.

If the library of the module being replaced is different from the library of the replacing module, the module's library after the update will remain the library the module was in when the program was first

created. If the RPLLIB parameter is required to determine which module to replace, the value to be entered in the RPLLIB parameter for this module will not change due to the update. Up to 300 names can be specified.

This is a required parameter.

Single values

***NONE**
No modules are specified.

Qualifier 1: Module

***ALL** All modules of the same name, to which the user has authority, replace the modules of the bound program.

generic-name
Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound program.

name Specify the name of the module that replaces a module of the bound program.

Qualifier 2: Library

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

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

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Specify the name of the module that replaces a module of the bound program.

Top

Replacement library (RPLLIB)

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

***ONLY**
The bound program contains only one module of the specified name and it is replaced. If two or more modules of the bound program have the specified name, an exception is signaled and the bound program is not updated.

***FIRST**
The first module of the specified name in the module list of the bound program is replaced.

***MODULE**
The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

name Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

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 program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound program. Up to 300 names can be specified.

Single values

***NONE**
No service programs are examined during symbol resolution.

Qualifier 1: Bind service program

***ALL** All service programs are examined during symbol resolution.

generic-name
Specify the generic name of the service programs to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

name Specify the name of the service program to examine during symbol resolution.

Qualifier 2: Library

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

name Specify the name of the library to be searched. QTEMP is not a valid library name for this parameter.

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 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:

1. 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

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

© Copyright IBM Corp. 1998, 2006

Update Service Program (UPDSRVPGM)

Where allowed to run: All environments (*ALL)
Threadsafe: 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 QRPLOBJ 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 QRPLOBJ 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	*Qualified object name*	Required, Positional 1
	Qualifier 1: Service program	*Name*	
	Qualifier 2: Library	*Name*, ***USRLIBL**, *CURLIB	
MODULE	Module	Single values: *NONE Other values (up to 300 repetitions): *Qualified object name*	Required, Positional 2
	Qualifier 1: Module	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
EXPORT	Export	***CURRENT**, *SRCFILE, *ALL	Optional

© Copyright IBM Corp. 1998, 2006

Keyword	Description	Choices	Notes
SRCFILE	Export source file	*Qualified object name*	Optional
	Qualifier 1: Export source file	*Name*, **QSRVSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Export source member	*Name*, ***SRVPGM**	Optional
RPLLIB	Replacement library	Single values: ***ONLY**, *FIRST, *MODULE Other values: *Qualifier list*	Optional
	Qualifier 1: Replacement library	*Name*	
BNDSRVPGM	Bind service program	Single values: ***NONE** Other values (up to 300 repetitions): *Qualified object name*	Optional
	Qualifier 1: Bind service program	*Generic name, name*, *ALL	
	Qualifier 2: Library	*Name*, ***LIBL**	
SRVPGMLIB	Bound *SRVPGM library name	Single values: ***SAME**, *LIBL Other values: *Qualifier list*	Optional
	Qualifier 1: Bound *SRVPGM library name	*Name*	
BNDDIR	Binding directory	Single values: ***NONE** Other values (up to 300 repetitions): *Qualified object name*	Optional
	Qualifier 1: Binding directory	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB, *USRLIBL	
ACTGRP	Activation group	*Name*, ***SAME**	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	***NONE**, *BASIC, *EXTENDED, *FULL	Optional

Top

Service program (SRVPGM)

Specifies the service program that is to be updated.

This is a required parameter.

Qualifier 1: Service program

name Specify the name of the bound service program that is to be updated.

Qualifier 2: Library

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

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

name Specify the name of the library where the bound service program is located.

Top

Module (MODULE)

Specifies the existing *MODULE objects that are to replace the modules of the same name in the bound program. Up to 300 names can be specified. If two or more modules of the bound program have the same name, the **Replacement library (RPLLIB)** parameter indicates which is to be replaced.

If the library of the module being replaced is different from the library of the replacing module, the module's library after the update will remain the library the module was in when the service program was first created. If the RPLLIB parameter is required to determine which module to replace, the value to be entered in the RPLLIB parameter for this module will not change due to the update.

This is a required parameter.

Single values

***NONE**
No modules are specified.

Note: This value can be specified when the module is not changing, but you are updating the service program (the BNDSRVPGM parameter) or the binding directory (BNDDIR parameter) to examine for exports. The existing module is used.

Qualifier 1: Module

***ALL** All modules of the same name to which the user has authority replace the modules of the bound service program.

generic-name
Specify the generic name of the modules that replace the modules of the bound program. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All modules with names that begin with the generic name, and for which the user has authority, replace the modules of the bound service program.

name Specify the name of the module that replaces a module of the bound service program.

Qualifier 2: Library

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

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

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Top

Export (EXPORT)

Specifies the variables and procedures that are to be exported from the updated service program. This parameter also specifies whether new signatures identifying the sequence of exports in the service program are created.

***CURRENT**
The variables, procedures, and signatures currently exported from the service program continue to be exported. No new signatures are created.

Note: If a variable or procedure that is currently exported is not available for export after the update, an exception is signaled and the service program is not updated.

***SRCFILE**
The source file member identified by the **Source file (SRCFILE)** and **Source member (SRCMBR)** parameters contains EXPORT statements that identify the data and procedures to export from the service program. If the specified source file differs from the one used to create the service program, a new signature or set of signatures may be created.

Note: If a signature is lost, some current clients of the service program may not be able to use the service program without binding again.

***ALL** All variables and procedures exported from the modules of the service program are exported from the updated service program.

If the number or names of the variables and procedures exported before the service program update differs from those exported after the service program update, a new signature is created.

Note: If a new signature is created, all clients of the service program must be bound again before they can use the service program.

Top

Export source file (SRCFILE)

Specifies the source file containing the specifications for exporting variables and procedures from this bound service program.

Qualifier 1: Export source file

QSRVSRC
The source file name is QSRVSRC.

name Specify the name of the source file containing the specifications for exporting variables and procedures.

Qualifier 2: Library

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

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

name Specify the name of the library to be searched.

Top

Export source member (SRCMBR)

Specifies the member in the file specified for the **Export source file (SRCFILE)** parameter that contains the specifications for exporting variables and procedures from this bound service program.

***SRVPGM**
The source file member has the same name as the service program being updated.

name Specify the name of the member that contains the specifications for exporting.

Top

Replacement library (RPLLIB)

Specifies the method used to select the module to be replaced when two or more modules of the bound program have the name specified on the MODULE parameter.

***ONLY**
The bound service program contains only one module of the specified name and it is replaced. If two or more modules of the bound service program have the specified name, an exception is signaled and the bound service program is not updated.

***FIRST**
The first module of the specified name in the module list of the bound service program is replaced.

***MODULE**
The module that originated from the same library as the specified module is replaced. If no module of the specified name originally came from the same library as the replacing module, no module is replaced and an exception is signaled.

name Specify the name of the originating library of the module to be selected for replacement. If no module of the specified name originated in the specified library, no module is replaced.

Top

Bind service program (BNDSRVPGM)

Specifies the service program to examine for exports if import requests to resolve external symbols cannot be met by the modules and service programs of the updated bound service program. If the specified service program can resolve external symbols, it is added to the service programs that are bound to the bound service program. Up to 300 names can be specified.

Single values

***NONE**
No service programs, except those in the bound service program being updated, are examined during symbol resolution.

Qualifier 1: Bind service program

***ALL** All service programs are examined during symbol resolution.

generic-name
Specify the generic name of the service programs to examine during symbol resolution. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. All service programs with names that begin with the generic name, and for which the user has authority, are examined during symbol resolution.

name Specify the name of the service program to examine during symbol resolution.

Qualifier 2: Library

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

name Specify the name of the library to be searched. QTEMP is not a valid library name for this parameter.

Top

Bound *SRVPGM library name (SRVPGMLIB)

Specifies the library name used to resolve to currently bound service programs. A value other than *SAME for this parameter can be specified if the program attribute ALWLIBUPD is *YES.

***SAME**
Use the library name where the service program (*SRVPGM) is currently bound from.

***LIBL** All libraries in the job's library list are searched until the first match is found for each bound *SRVPGM. The first occurrence of a *SRVPGM is used to resolve to currently bound service programs and *LIBL is saved to be used at run time. If no match is found in the job's library list, the *SRVPGM currently bound to the program is used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself.

Note: The service programs that came from the implicit binding directories (system-supplied service programs) are not changed.

name Specify the name of the library to be used first to resolve to all currently bound service programs. If a bound *SRVPGM does exist in the library specified on SRVPGMLIB parameter, that *SRVPGM from that library is used instead of the currently bound *SRVPGM and the library name specified on the SRVPGMLIB parameter is saved to be used at run time. If a *SRVPGM does not exist in the library specified on the SRVPGMLIB parameter, the *SRVPGM currently bound to the program is used. You must have *USE authority to the *SRVPGM objects in the library specified and *EXECUTE authority to the library itself.

Top

Binding directory (BNDDIR)

Specifies the binding directory to examine for exports if import requests to resolve external symbols cannot be met by (1) the modules and service programs of the updated bound program or by (2) the service program specified on the BNDSRVPGM parameter. If a module or service program listed in the specified binding directory can resolve external symbols, it is added to the modules or service programs that are bound to the bound service program. Up to 300 names can be specified.

Single values

***NONE**
No binding directories, except those in the bound service program being updated, are examined during symbol resolution.

Qualifier 1: Binding directory

name Specify the name of the binding directory to be used during symbol resolution.

Qualifier 2: Library

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

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

***USRLIBL**
Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Top

Activation group (ACTGRP)

Specifies the activation group to be used for the updated service program.

***SAME**
The activation group is not changed. Specify this value if the service program was given *CALLER activation group at the time it was created.

name Specify the name of the activation group that is associated with this called service program. If the service program was given a named activation group at the time it was created, the name of that activation group can be changed to another named activation group.

Note: Changing the activation group name can affect the behavior of the program (or service program). Refer to the ILE Concepts book, SC41-5606 for detailed information on the behavior of named activation groups.

Top

Creation options (OPTION)

Specifies the options to be used when the service program object is updated.

You can specify up to 6 values for this parameter.

Creating Program Objects

***GEN** An updated program object is created.

***NOGEN**
An updated program object is not created.

Duplicate Procedure Names

***DUPPROC**
During symbol resolution, the procedures that are exported from the modules and service programs need not be unique. The first procedure of the specified modules and service programs that matches the import request is exported.

***NODUPPROC**
During symbol resolution, each procedure that is exported from the modules and service programs must be unique.

Duplicate Variable Names

***DUPVAR**
During symbol resolution, the variables that are exported from the modules and service programs need not be unique. The first variable of the specified modules and programs that matches the import request is exported.

***NODUPVAR**
During symbol resolution, each variable that is exported from the modules and service programs must be unique.

Issuing Diagnostic Messages

***WARN**
The appropriate diagnostic messages are signaled. Also, if you specify duplicate procedures or variables (*DUPPROC or *DUPVAR) and duplicates are found, a diagnostic message is issued indicating what duplicates were found.

***NOWARN**
No information or diagnostic messages are issued.

Trimming Marooned Modules

A **marooned module** is a module of the bound program being updated. This module was originally bound into the bound program from a binding directory to resolve one or more imports. Imports are not resolved for this program update.

***NOTRIM**
Marooned modules are not removed from the bound program.

Note: Bound service programs may grow significantly over time when *NOTRIM is specified.

***TRIM**
Marooned modules are removed from the bound program.

Note: If marooned modules are removed from the bound program during this program update, the exports that the modules contain are not available for other program updates.

Resolving References (Imports)

***RSLVREF**
All imports must resolve to exports for the bound service program to be updated.

***UNRSLVREF**
All imports do not need to resolve to exports for the bound service program to be updated.

Note: If this command contains an import that does not resolve, an exception will be generated when the command is run.

Top

Listing detail (DETAIL)

Specifies the level of detail of the binder listing to be printed. The printer file *LIBL/QSYSPRT is used to create the listing.

***NONE**
No binder listing is printed.

***BASIC**
The brief summary table, the options passed to this command, and some processing statistics are printed.

***EXTENDED**
The extended summary table and the binding information listing are printed, in addition to the information provided in the *BASIC listing (the brief summary table, the options passed to this command, and some processing statistics).

***FULL** The cross-reference listing is printed, in addition to the information provided in the *EXTENDED listing (the extended summary table, the binding information listing, the brief summary table, the options passed to this command, and some processing statistics).

Top

Examples

```
UPDSRVPGM   SRVPGM(WORKDOC)  MODULE(BIN/TASKTWO)
            RPLLIB(*MODULE)
```

This command replaces the module named TASKTWO in the service program object named WORKDOC with another module named TASKTWO in the library BIN, only if the module being replaced was originally from the library BIN.

Top

Error messages

***ESCAPE Messages**

CPF223D
Not authorized to update &1 in &2 type *&3.

CPF223E
Authority check for use adopted authority attribute failed.

CPF5CA7
SRVPGMLIB must be *SAME when ALWLIBUPD is *NO.

CPF5CE1
Service program &1 not updated.

CPF5CE2
Unexpected error occurred during program or service program update.

CPF5D1C
Update of service program &1 in library &2 not allowed.

Top

Update System Information (UPDSYSINF)

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

Parameters
Examples
Error messages

The Update System Information (UPDSYSINF) command updates various system information that was gathered using the Retrieve System Information (RTVSYSINF) 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	*Name*	Required, Positional 1
TYPE	Type of information	***ALL**, *EDTD, *NETA, *RPYLE, *SRVATR, *SRVPVD, *SYSVAL	Optional, Positional 2

Top

Library (LIB)

Specifies the library where the system information was placed by the Retrieve System Information (RTVSYSINF) command.

This is a required parameter.

The possible values are:

library-name
Specify the library name where the system information exists.

Top

© Copyright IBM Corp. 1998, 2006

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 (RTVSYSINF) command.

***EDTD**
Update all edit descriptions saved on the source system by the Retrieve System Information (RTVSYSINF) command.

***NETA**
Update all network attributes saved on the source system by the Retrieve System Information (RTVSYSINF) command.

***RPYLE**
Update all reply list entries saved on the source system by the Retrieve System Information (RTVSYSINF) command.

***SRVATR**
Update all service attributes saved on the source system by the Retrieve System Information (RTVSYSINF) command.

***SRVPVD**
Update all service providers saved on the source system by the Retrieve System Information (RTVSYSINF) command.

***SYSVAL**
Update all system values saved on the source system by the Retrieve System Information (RTVSYSINF) command.

Top

Examples

```
UPDSYSINF   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 OS/400 function.

Top

Verify APPC Connection (VFYAPPCCNN)

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

Parameters
Examples
Error messages

The Verify APPC Connection (VFYAPPCCNN) 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	*Character value*	Required, Positional 1
MODE	Mode	*Communications name*, ***NETATR**	Optional, Positional 2
RMTUSER	Remote user ID	*Character value*, ***NONE**, *CURRENT	Optional
RMTPWD	Remote password	*Character value*, ***NONE**	Optional
MSGMODE	Message mode	***VERBOSE**, *QUIET	Optional
PKTLEN	Packet length (in bytes)	0-32763, **100**	Optional
NBRITER	Number of iterations	1-32767, **2**	Optional
NBRPKT	Number of packets	1-32767, **1**	Optional
ECHO	Echo	***YES**, *NO	Optional
WAITTIME	Wait time (in seconds)	2-3600, **10**, *NOMAX, *NOWAIT	Optional

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format nnnnnnnn.cccccccc, where nnnnnnnn is the network identifier (ID) and cccccccc is the remote location name. If only the remote location name is specified, the local network ID (LCLNETID) network attribute is used as the value of the network identifier (ID).

Top

Mode (MODE)

Specifies the name of the mode to be used for the APPC conversation.

The possible values are:

© Copyright IBM Corp. 1998, 2006

***NETATR**
The mode in the network attributes is used.

mode-name
Specify a mode name. Specify BLANK for a mode name consisting of eight blank characters.

Note: SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

Remote user ID (RMTUSER)

Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMTPWD(*NONE) is not valid.

The possible values are:

***NONE**
No user ID is sent. If security on the target system is configured to require a user ID, the command will fail.

***CURRENT**
The user ID of the job (signed-on user) using this command is sent.

remote-user-identifier
Specify a user ID to use that exists on the target system. If a user ID is specified and password security is active on the target system, a password must be specified.

Top

Remote password (RMTPWD)

Specifies the password sent to the target system.

The possible values are:

***NONE**
The system does not send a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, the command will fail.

password
Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. The password may or may not be substituted across the communication line depending on whether the remote system supports password substitution.

Top

Message mode (MSGMODE)

Specifies the amount of information displayed by the command.

The possible values are:

***VERBOSE**
Display verification message after each iteration.

***QUIET**
Display only initial and summary messages.

Top

Packet length (in bytes) (PKTLEN)

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

The possible values are:

100 The packet length is 100 bytes.

packet-length
Specify the length of the packet. Valid values range from 0 through 32763 bytes.

Top

Number of iterations (NBRITER)

Specifies the number of iterations. For each iteration, the specified number of data packets are exchanged between the local and remote systems.

The possible values are:

2 Two iterations are performed.

number-of-iterations
Specify the number of iterations. Valid values range from 1 through 32767.

Top

Number of packets (NBRPKT)

Specifies the number of packets that are sent by the local system for each iteration before giving the target system permission to send.

The possible values are:

1 One packet is sent for each iteration.

number-of-packets
Specify the number of packets that are sent for each iteration. Valid values range from 1 through 32767.

Top

Echo (ECHO)

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

The possible values are:

***YES** Packets are echoed back from the remote location to the local location.

***NO** Packets are sent from the local location to the remote location only; packets are not echoed back to the local location.

Top

Wait time (in seconds) (WAITTIME)

Specifies the time in seconds to wait for the return (echo) before declaring the remote location to be unreachable.

The possible values are:

10 The system waits 10 seconds.

***NOMAX**
The system waits forever.

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

time-to-wait-for-reply
Specify the length of time in seconds. Valid values range from 2 through 3600 seconds.

Top

Examples

Example 1: Verify an APPC Connection

```
VFYAPPCCNN   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

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY8809
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

© Copyright IBM Corp. 1998, 2006

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2

IBM
Infoprint
InfoWindow
iSeries
LPDA
OfficeVision
i5/OS
Print Services Facility
RPG/400
SystemView
System/36
TCS
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

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 AND FITNESS FOR A PARTICULAR PURPOSE

Printed in USA