



System i

Programming

i5/OS commands

Starting with INZPCS (Initialize Client Access/400)

Version 6 Release 1





System i

Programming

i5/OS commands

Starting with INZPCS (Initialize Client Access/400)

Version 6 Release 1

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 699.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

© **Copyright International Business Machines Corporation 1998, 2008. All rights reserved.**

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

Contents

Initialize System i Access (INZPCS) . . . 1

Error messages for INZPCS	1
Parameters	1
Keyboard type (KBDTYPE)	2
ASCII code page number (ASCII)	4
EBCDIC code page number (EBCDIC)	4
Language feature code (LANGUAGE)	5
Examples	6
Error messages	7

Initialize Physical File Mbr (INZPFM) . . . 9

Parameters	9
Physical file (FILE)	9
Member (MBR)	10
Initialize records as (RECORDS)	10
Total number of records (TOTRCDS)	10
Examples	10
Error messages	11

Initialize System (INZSYS) 13

Parameters	13
Message queue (MSGQ)	13
Examples	14
Error messages	14

Initialize Tape (INZTAP). 15

Parameters	15
Device (DEV).	15
New volume identifier (NEWVOL)	15
New owner identifier (NEWOWNID).	16
Volume identifier (VOL)	16
Check for active files (CHECK).	17
Tape density (DENSITY)	17
Code (CODE).	22
End of tape option (ENDOPT)	22
Clear (CLEAR)	22
Examples	22
Error messages	23

Iterate (ITERATE) 25

Parameters	25
Command label (CMDLBL)	25
Examples	25
Error messages	26

Copy From LDIF (LDIF2DB). 27

Parameters	27
LDIF stream file (LDIFSTMF)	27
Instance (INSTANCE)	28
Administrator (ADMIN)	28
Replicate imported data (REPLICATE)	28
Examples	28
Error messages	29

Leave (LEAVE) 31

Parameters	31
Command label (CMDLBL)	31
Examples	32
Error messages	32

Link/Unlink Data Definition (LNKDTADFN). 33

Parameters	33
Option (OPTION)	33
Data base file (FILE)	34
Data dictionary (DTADCT)	34
File definition (DFN)	34
Creation date (CRTDATE)	34
Examples	35
Error messages	35

Load or Unload Image Catalog (LODIMGCLG). 37

Parameters	37
Image catalog (IMGCLG).	38
Option (OPTION)	38
Virtual device (DEV)	38
Write protect (WRTPTC)	38
Examples	39
Error messages	40

Load/Unload/Mount IMGCLG Entry (LODIMGCLGE) 41

Parameters	41
Image catalog (IMGCLG).	42
Image catalog index (IMGCLGIDX)	42
Option (OPTION)	42
Image catalog volume (VOL)	42
Virtual device (DEV)	43
Examples	43
Error messages	43

Load Optical Firmware (LODOPTFMW) 45

Parameters	45
From stream file (FROMSTMF)	45
Device type (TYPE).	45
Device model (MODEL)	45
Examples	46
Error messages	46

Load Program Temporary Fix (LODPTF) 47

Parameters	47
Product (LICPGM)	47
Device (DEV).	48
PTF numbers to select (SELECT)	48
PTF numbers to omit (OMIT)	48
Superseded PTFs (SPRPTF)	48
Release (RLS).	49

Sequence number (SEQNBR)	49
End of media option (ENDOPT)	49
Path identifier (PATHID)	50
Save file (SAVF)	50
Copy PTF cover letter (COVER)	51
Copy PTF save file (CPYSAVF)	51
Examples	51
Error messages	52

Load Q/A Database (LODQSTDB) 55

Parameters	55
Q/A database (QSTDB)	55
Lib containing Q/A database (LIB)	56
Examples	56
Error messages	56

Load and Run (LODRUN) 57

Parameters	59
Device (DEV)	59
Sequence number (SEQNBR)	59
Volume identifier (VOL)	60
Directory (DIR)	60
Examples	60
Error messages	61

Create Directory (MD) 63

Parameters	64
Directory (DIR)	64
Public authority for data (DTAAUT)	64
Public authority for object (OBJAUT)	65
Auditing value for objects (CRTOBJAUD)	66
Scanning option for objects (CRTOBJSCAN)	66
Restricted rename and unlink (RSTDRNMUNL)	67
Examples	67
Error messages	68

Create Directory (MKDIR) 69

Parameters	70
Directory (DIR)	70
Public authority for data (DTAAUT)	70
Public authority for object (OBJAUT)	71
Auditing value for objects (CRTOBJAUD)	72
Scanning option for objects (CRTOBJSCAN)	72
Restricted rename and unlink (RSTDRNMUNL)	73
Examples	73
Error messages	74

Monitor Message (MONMSG) 75

Parameters	76
Message identifier (MSGID)	76
Comparison data (CMPDTA)	77
Command to execute (EXEC)	77
Examples	77
Error messages	78

Add Mounted FS (MOUNT) 79

Parameters	79
Type of file system (TYPE)	80
File system to mount (MFS)	80

Directory to mount over (MNTOVRDIR)	80
Mount options (OPTIONS)	81
Coded character set ID (CCSID)	83
Code page (CODEPAGE)	83
Examples	84
Error messages	85

Move Object (MOV) 87

Parameters	88
Object (OBJ)	88
To directory (TODIR)	88
To object (TOOBJ)	89
From CCSID (FROMCCSID)	89
To CCSID (TOCCSID)	89
Data Format (DTAFMT)	90
From code page (FROMCODPAG)	91
To code page (TOCODEPAGE)	91
Examples	92
Error messages	92

Move Document (MOVDOC) 95

Parameters	95
From document (FROMDOC)	95
From folder (FROMFLR)	96
To folder (TOFLR)	96
Rename (RENAME)	96
System object name (SYSOBJNAM)	96
Examples	97
Error messages	97

Move Object (MOVE) 99

Parameters	100
Object (OBJ)	100
To directory (TODIR)	100
To object (TOOBJ)	101
From CCSID (FROMCCSID)	101
To CCSID (TOCCSID)	101
Data Format (DTAFMT)	102
From code page (FROMCODPAG)	103
To code page (TOCODEPAGE)	103
Examples	104
Error messages	104

Move Object (MOV OBJ) 107

Parameters	109
Object (OBJ)	109
Object type (OBJTYPE)	109
To library (TOLIB)	110
From ASP device (ASPDEV)	110
To ASP device (TOASPDEV)	111
Examples	111
Error messages	112

Merge Message Catalog (MRGMSGCLG) 117

Parameters	117
Message catalog name (CLGFILE)	117
Source file path name (SRCFILE)	118
Message catalog CCSID (CLGCCSID)	118

Text 'description' (TEXT)	118
Source file CCSID (SRCCCSID)	118
Public authority for data (DTAAUT).	119
Public authority for object (OBJAUT)	120
Examples.	120
Error messages	120

Merge Message File (MRGMSGF) 121

Parameters	121
From message file (FROMMSGF)	122
To message file (TOMSGF)	122
Replaced message file (RPLMSGF)	122
Message IDs to select (SELECT)	123
Message IDs to omit (OMIT)	123
Examples.	123
Error messages	124

Merge TCP/IP Host Table (MRGTCPHT) 127

Parameters	127
From file (FROMFILE)	127
From member (FROMMBR)	128
File format (FILEFMT)	128
Replace host table (REPLACE).	129
Examples.	129
Error messages	129

Work with TCP/IP Network Sts (NETSTAT). 131

Parameters	131
Option (OPTION)	131
Examples.	131
Error messages	132

Start NSLOOKUP Query (NSLOOKUP) 133

Parameters	135
Query name (HOSTNAME)	135
Domain name server (DMNNAMSVR)	136
Examples.	136
Error messages	137

Run DNS Update (NSUPDATE) 139

Parameters	139
Batch input file (BCHFILE)	140
Show debug information (DEBUG)	140
Key name (KEYNAME)	141
Key file (KEYFILE)	141
Update timeout (TIMEOUT)	141
UDP timeout (UDPRTYITV)	142
UDP retry (UDPNBRRTY)	142
Network protocol (PROTOCOL)	142
Output file (TOSTMF)	142
Examples.	142
Error messages	145

Open Data Base File (OPNDBF) 147

Parameters	147
File (FILE)	148
Open option (OPTION)	148

Member to be opened (MBR)	148
Open file identifier (OPNID)	148
Access path to use (ACCPATH)	149
Limit to sequential only (SEQONLY)	149
Commitment control active (COMMIT).	149
Open scope (OPNSCOPE)	150
Duplicate key check (DUPKEYCHK)	150
Type of open (TYPE)	150
Examples.	150
Error messages	151

Open Query File (OPNQRYF). 153

Parameters	154
File specifications (FILE).	156
Open options (OPTION).	157
Format specifications (FORMAT)	157
Query selection expression (QRYSLT)	158
Key field specifications (KEYFLD)	159
Unique key fields (UNIQUEKEY)	160
Join field specifications (JFLD).	161
Join with default values (JDFTVAL)	163
Join file order (JORDER).	163
Grouping field names (GRPFLD)	164
Group selection expression (GRPSLT)	165
Mapped field specifications (MAPFLD).	165
Ignore decimal data errors (IGNDECERR).	169
Open file identifier (OPNID)	169
Limit to sequential only (SEQONLY)	169
Commitment control active (COMMIT).	169
Open scope (OPNSCOPE)	170
Duplicate key check (DUPKEYCHK)	170
Allow copy of data (ALWCPYDTA)	170
Performance optimization (OPTIMIZE)	171
Optimize all access paths (OPTALLAP).	172
Sort sequence (SRTSEQ)	172
Language ID (LANGID).	173
Final output CCSID (CCSID)	173
Type of open (TYPE)	173
Examples.	174
Error messages	178

Otherwise (OTHERWISE) 181

Parameters	181
Command (CMD)	181
Examples.	182
Error messages	182

Override with Data Base File (OVRDBF). 183

Parameters	183
File being overridden (FILE)	184
Overriding to data base file (TOFILE)	185
Overriding member (MBR)	185
Starting position in file (POSITION)	186
Record format lock (RCDFMTLCK)	187
Records to force a write (FRCRATIO)	188
Rcd format selector program (FMTSLR)	188
Maximum file wait time (WAITFILE)	189
Maximum record wait time (WAITRCD)	189
Records retrieved at once (NBRRCD)	190

EOF retry delay in sec (EOFDLY)	190
Record format level check (LVLCHK)	191
Check expiration date (EXPCHK)	191
Inhibit write (INHWRIT)	191
Secure from other overrides (SECURE)	191
Override scope (OVRSCOPE)	191
Share open data path (SHARE)	192
Open scope (OPNSCOPE)	192
Limit to sequential only (SEQONLY)	192
Distributed Data (DSTDTA)	194
Examples	194
Error messages	195

Override with Display File (OVRDSPF) 197

Parameters	197
File being overridden (FILE)	198
Overriding to display file (TOFILE)	198
Device (DEV)	199
Character identifier (CHRID)	199
Decimal format (DECFMT)	200
SFLEND text (SFLENDTXT)	200
User specified DBCS data (IGCDTA)	200
DBCS extension characters (IGCEXNCHR)	201
Maximum file wait time (WAITFILE)	201
Maximum record wait time (WAITRCD)	201
Record format level check (LVLCHK)	202
Secure from other overrides (SECURE)	202
Override scope (OVRSCOPE)	202
Data queue (DTAQ)	203
Share open data path (SHARE)	203
Open scope (OPNSCOPE)	203
Examples	204
Error messages	204

Override ICF Pgm Device Entry (OVRICFDEVE) 205

Parameters	205
Program device (PGMDEV)	206
Remote location (RMTLOCNAME)	206
Communication type (CMNTYPE)	207
Device (DEV)	208
Local location (LCLLOCNAME)	208
Mode (MODE)	208
Remote network identifier (RMTNETID)	209
Format select (FMTSLT)	209
Application identifier (APPID)	210
Batch activity (BATCH)	210
Host type (HOST)	210
End session with host (ENDSSNHOST)	210
Special host application (SPCHOSTAPP)	211
Initialize self (INZSELF)	211
Header processing (HDRPROC)	211
Message protection (MSGPTC)	211
Emulation device (EMLDEV)	212
Conversation type (CNVTYPE)	212
Blocking type (BLOCK)	213
Record length (RCDLEN)	214
Block length (BLKLEN)	214
Transmit in transparent mode (TRNSPY)	214
Data compression (DTACPR)	215

Truncate trailing blanks (TRUNC)	215
Overflow data (OVRFLWDTA)	215
Group separator type (GRPSEP)	216
Remote BSCCEL (RMTBSCCEL)	216
Initial connection (INLCNN)	216
Secure from other overrides (SECURE)	217
Override scope (OVRSCOPE)	217
Examples	217
Error messages	218

Override ICF File (OVRICFF) 219

Parameters	219
File being overridden (FILE)	220
Overriding to ICF file (TOFILE)	220
Acquire program device (ACQPGMDEV)	220
Maximum record length (MAXRCDLEN)	221
Maximum file wait time (WAITFILE)	221
Maximum record wait time (WAITRCD)	221
Record format level check (LVLCHK)	222
Secure from other overrides (SECURE)	222
Override scope (OVRSCOPE)	222
Data queue (DTAQ)	223
Share open data path (SHARE)	223
Open scope (OPNSCOPE)	223
Examples	224
Error messages	224

Override Message File (OVRMSGF) 225

Parameters	225
Message file being overridden (MSGF)	225
Overriding to message file (TOMSGF)	226
Secure from other overrides (SECURE)	226
Examples	226
Error messages	226

Override with Printer File (OVRPRTF) 229

Parameters	229
File being overridden (FILE)	234
Overriding to printer file (TOFILE)	234
Device (DEV)	234
Printer device type (DEVTYPE)	235
Page size (PAGESIZE)	235
Lines per inch (LPI)	236
Characters per inch (CPI)	237
Front margin (FRONTMGN)	237
Back margin (BACKMGN)	238
Overflow line number (OVRFLW)	238
Fold records (FOLD)	239
Unprintable character action (RPLUNPRT)	239
Align page (ALIGN)	240
Source drawer (DRAWER)	240
Output bin (OUTBIN)	240
Font specifications (FONT)	240
Form feed (FORMFEED)	241
Print quality (PRTQLTY)	241
Control character (CTLCHAR)	242
Channel values (CHLVAL)	243
Fidelity (FIDELITY)	243
Character identifier (CHRID)	244
Decimal format (DECFMT)	244

Font character set (FNTCHRSET)	245
Coded font (CDEFNT)	246
Page definition (PAGDFN)	246
Form definition (FORMDF)	247
AFP Characters (AFPCHARS)	248
Table Reference Characters (TBLREFCHR)	248
Degree of page rotation (PAGRTT)	249
Pages per side (MULTIUP)	249
Reduce output (REDUCE)	249
Print text (PRTTXT)	250
Hardware justification (JUSTIFY)	250
Print on both sides (DUPLEX)	250
Unit of measure (UOM)	250
Front side overlay (FRONTOVL)	251
Back side overlay (BACKOVL)	251
Convert line data (CVTLINDTA)	253
IPDS pass through (IPDSPASTHR)	253
User resource library list (USRRSCLIBL)	253
Corner staple (CORNERSTPL)	254
Edge stitch (EDGESTITCH)	255
Saddle stitch (SADLSTITCH)	256
Font resolution for formatting (RNTRSL)	257
Defer write (DFRWRT)	257
Spool the data (SPOOL)	258
Output queue (OUTQ)	258
Form type (FORMTYPE)	258
Copies (COPIES)	258
Page range to print (PAGERANGE)	259
Max spooled output records (MAXRCDS)	259
File separators (FILESEP)	259
Spooled output schedule (SCHEDULE)	259
Hold spooled file (HOLD)	260
Save spooled file (SAVE)	260
Output priority (on OUTQ) (OUTPTY)	260
User data (USRDTA)	260
Spool file owner (SPLFOWN)	261
User Defined Option (USRDFNOPT)	261
User Defined Data (USRDFNDTA)	262
User Defined Object (USRDFNOBJ)	262
Spool file name (SPLFNAME)	263
Expiration date for file (EXPDATE)	263
Days until file expires (DAYS)	264
User specified DBCS data (IGCDTA)	264
DBCS extension characters (IGCEXNCHR)	264
DBCS character rotation (IGCCHRRTT)	264
DBCS characters per inch (IGCCPI)	265
DBCS SO/SI spacing (IGCSOSI)	265
DBCS coded font (IGCCDEFNT)	265
To stream file (TOSTMF)	266
Workstation customizing object (WSCST)	266
Maximum file wait time (WAITFILE)	267
Record format level check (LVLCHK)	267
Secure from other overrides (SECURE)	267
Override scope (OVRSCOPE)	268
Share open data path (SHARE)	268
Open scope (OPNSCOPE)	268
Examples	268
Error messages	269

Override with Save File (OVRSAVF)	271
Parameters	271

File being overridden (FILE)	271
Save file (TOFILE)	272
Extend file (EXTEND)	272
Starting position in file (POSITION)	273
Maximum file wait time (WAITFILE)	273
Secure from other overrides (SECURE)	273
Override scope (OVRSCOPE)	274
Share open data path (SHARE)	274
Open scope (OPNSCOPE)	274
Examples	274
Error messages	275

Override with Tape File (OVRTAPF)	277
Parameters	277
File being overridden (FILE)	278
Overriding to tape file (TOFILE)	279
Device (DEV)	279
Volume identifier (VOL)	279
Tape reels specifications (REELS)	280
Sequence number (SEQNBR)	282
File label (LABEL)	282
Record length (RCDLEN)	282
Block length (BLKLEN)	283
Buffer offset (BUFOFSET)	283
Record block format (RCDBLKFMF)	284
Extend file (EXTEND)	285
Tape density (DENSITY)	286
Data compaction (COMPACT)	291
Code (CODE)	291
Creation date (CRTDATE)	291
File expiration date (EXPDATE)	291
End of tape option (ENDOPT)	292
User label program (USRLBLPGM)	292
Conversion table (TBL)	293
From CCSID (FROMCCSID)	293
To CCSID (TOCCSID)	293
User specified DBCS data (IGCDTA)	294
Maximum file wait time (WAITFILE)	294
Secure from other overrides (SECURE)	294
Override scope (OVRSCOPE)	294
Share open data path (SHARE)	295
Open scope (OPNSCOPE)	295
Examples	295
Error messages	296

Parameter Definition (PARM)	297
Parameters	297
Keyword (KWD)	299
Type of value (TYPE)	299
Value length (LEN)	301
Return value (RTNVAL)	302
Constant value (CONSTANT)	303
Restricted values (RSTD)	304
Default value (DFT)	304
Valid values (VALUES)	305
Relational expression (REL)	305
Range of values (RANGE)	306
Special values (SPCVL)	307
Single values (SNGVAL)	307
Minimum values required (MIN)	308

Maximum values allowed (MAX)	308
Allow unprintable characters (ALWUNPRT)	309
Allow variable names (ALWVAR)	309
Is PARM a program name (PGM)	309
Is PARM a data area name (DTAARA)	309
If a file parameter, how used (FILE)	310
Full field required (FULL)	310
Value an expression (EXPR)	310
Varying length (VARY)	311
Pass attribute byte (PASSATR)	311
Value to pass if unspecified (PASSVAL)	312
Case of value (CASE)	312
CCSID of value (CCSID)	313
List displacement (LISTDSPL)	313
Display input (DSPINPUT)	313
Choice text (CHOICE)	314
Choice program (CHOICEPGM)	315
Prompt control (PMTCTL)	315
Prompt control program (PMTCTLPGM)	316
Key parameter (KEYPARM)	316
Initial prompt length (INLPMTLEN)	316
Prompt specifications (PROMPT)	317
Examples	317
Error messages	318

Program (PGM) 319

Parameters	319
Parameter CL variable names (PARM)	320
Examples	320
Error messages	321

Verify TCP/IP Connection (PING) 323

Parameters	323
Remote system (RMTSYS)	323
Remote internet address (INTNETADR)	324
Address version format (ADRVERFMT)	324
Message mode (MSGMODE)	324
Packet length (in bytes) (PKTLEN)	325
Number of packets (NBRPKT)	325
Wait time (in seconds) (WAITTIME)	325
Local internet address (LCLINTNETA)	325
Type of service (TOS)	325
IP time to live (hop limit) (IPTTL)	326
Examples	326
Error messages	328

Configure PM Agent Line (PMLINMON). 329

Parameters	329
Examples	329
Error messages	329

Prompt Control Definition (PMTCTL) 331

Parameters	331
Controlling keyword (CTL)	331
Controlling conditions (COND)	331
Number of true conditions (NBRTRUE)	332
Logical relation (LGLREL)	332
Examples	332
Error messages	333

Position Data Base File (POSDBF) 335

Parameters	335
Open file identifier (OPNID)	335
File position (POSITION)	335
Examples	336
Error messages	336

Print Adopting Objects (PRTADPOBJ) 337

Parameters	338
User profile (USRPRF)	338
Changed report only (CHGRPTONLY)	338
Examples	338
Error messages	338

Print AFP Data (PRTAFPDTA) 341

Parameters	341
File (FILE)	341
Member (MBR)	342
Print device (DEV)	342
Form definition (FORMDF)	342
Number of copies (COPIES)	343
Starting page (STRPAGE)	343
Ending page (ENDPAGE)	343
Print fidelity (FIDELITY)	343
Examples	343
Error messages	344

Print Command Usage (PRTCMDUSG) 345

Parameters	345
Command (CMD)	345
Program (PGM)	346
Examples	347
Error messages	347

Print Communications Security (PRTCMNSEC) 349

Parameters	350
Changed report only (CHGRPTONLY)	351
Examples	351
Error messages	351

Print Communications Trace (PRTCMNTRC) 353

Parameters	353
Configuration object (CFGOBJ)	354
From stream file (FROMSTMF)	354
Type (CFGTYPE)	355
Output (OUTPUT)	355
File to receive output (OUTFILE)	355
Output member options (OUTMBR)	356
Character code (CODE)	356
Line description (SLTLIND)	356
Controller description (SLTCTLD)	356
Format SNA data only (FMTSNA)	357
Format RR, RNR commands (FMTRR)	357
Format TCP/IP data (FMTCIP)	357
Format LCP data (FMTLCP)	357
Format NCP data (FMTNCP)	358
Format TCP/IP data by address (TCPIPADR)	358

IP port number (SLTPORT)	358
Format UI data only (FMTUI)	359
Format MAC or SMT data only (FMTMAC)	359
Format Ethernet data only (FMTETH)	359
Format call control data (FMTCCD)	359
Format broadcast data (FMTBCD)	359
Exclude LMI data (EXCLMI)	360
Format LMI data only (FMTLMI)	360
Format HPR over IP data only (FMTHPRIP)	360
Format LDLC over IP data only (FMTLDLCIP)	360
Examples	360
Error messages	360

Print Device Addresses (PRTDEVADR) 363

Parameters	363
Controller description (CTLD)	363
Examples	363
Error messages	363

Print Directory Information (PRTDIRINF) 365

Parameters	365
Type of report (RPTTYPE)	366
Information file prefix (INFFILEPFX)	366
Information library (INFLIB)	366
Owners (OWNER)	366
Objects (OBJ)	367
Object types (OBJTYPE)	367
Directory subtree (SUBTREE)	367
Smallest size (MINSIZE)	368
Sort by (SORT)	368
Include path names (INCPATH)	368
Examples	368
Error messages	369

Print Document (PRTDOC) 371

Parameters	371
Document (DOC)	373
Folder (FLR)	373
Display print options (OPTIONS)	373
Printer file (PRTFILE)	374
File to receive output (OUTFILE)	374
Output member options (OUTMBR)	375
Current status value (CURSTS)	375
New status value (NEWSTS)	376
Type of data for output (OUTDTATYP)	376
Delete document (DLTDOC)	376
Output device (OUTPUT)	376
Print device (DEV)	377
Output queue (OUTQ)	377
Output file (SPLFILE)	377
Form type (FORMTYPE)	378
Print separator page (COVERPAGE)	378
Print quality (PRTQLTY)	378
Number of copies (COPIES)	379
Print on both sides (DUPLEX)	379
Automatic page binding (AUTOBIND)	379
Delay printing (HOLD)	379
Print document error log (PRTERRLOG)	380
Error log form type (ERRFORM)	380

Large print (LARGEPRINT)	380
Merge type (MRGTYPE)	380
Query (QRYDFN)	381
Data document (DTADOC)	381
Data folder (DTAFLR)	381
Data file (DTAFILE)	381
Data member (DTAMBR)	382
Multiple line report (MLTLINRPT)	382
Adjust line endings (ADJLINES)	382
Adjust page endings (ADJPAGES)	383
Allow widow lines (ALWWIDOW)	383
Renumber system page numbers (RENUMBER)	383
Print change symbols (PRTCHGSYM)	383
Change symbols to print (SYMBOLS)	384
Draft spacing (DRAFTSPACE)	384
Print line numbers (LINNBR)	384
Resolve instructions (RESOLVE)	384
Additional spaces to left (LEFTSPACES)	385
Character identifier (CHRID)	385
Save resolved output (SAVOUTPUT)	385
Resolved output document (SAVDOC)	385
Resolved output folder (SAVFLR)	386
Place on job queue (JOBQ)	386
Job description (JOBDESC)	386
Send completion message (SNDMSG)	386
Cancel on error (CNLERR)	387
Start page (STRPAGE)	387
End page (ENDPAGE)	387
Page ranges (PAGERANGE)	388
Number of labels across page (LBLACROSS)	388
Width of labels (LBLWIDTH)	389
Sheet feed labels (SHEETFEED)	389
Number of rows per sheet (LBLDOWN)	389
Shift left margin (SHFLEFTMAR)	389
Examples	390
Error messages	391

Print Disk Information (PRTDSKINF) 395

Parameters	395
Type of report (RPTTYPE)	396
ASP device (ASPDEV)	396
Libraries (LIB)	396
Owners (OWNER)	396
Folders (FLR)	397
Documents (DOC)	397
Objects (OBJ)	397
Object types (OBJTYPE)	397
Smallest size (MINSIZE)	398
Sort by (SORT)	398
Examples	398
Error messages	399

Print Error Log (PRTERRLOG) 401

Parameters	401
Type of log data to list (TYPE)	402
Logical device (DEV)	402
Resource name (RSRCNAME)	403
Error log identifier (ERRLOGID)	403
Output (OUTPUT)	403
Time period for log output (PERIOD)	403

Print format (PRTFMT parameter)	404
File to receive output (OUTFILE)	405
Output member options (OUTMBR)	405
Volume type (VOLTYPE)	406
Model number (MODEL)	406
Volume (VOL)	406
Volume statistical data (VOLSTAT)	406
Volume statistics type (VOLSTATTYP)	407
Error log entries to select (SELECT)	407
Sort by (SORT)	407
Examples	408
Error messages	408

Print Internal Data (PRTINTDTA) 411

Parameters	411
Type of data (TYPE)	412
Dump identifier (DMPID)	412
Time period for internal data (PERIOD)	412
Job name (JOB)	413
Thread ID to include (SLTTHD)	414
Server type (SVRTYPE)	414
Examples	415
Error messages	415

Print IP over SNA (PRTIPSCFG) 417

Parameters	417
Examples	417
Error messages	417

Print JOB Authority (PRTJOBDAUT) 419

Parameters	420
Library (LIB)	420
Changed report only (CHGRPTONLY)	421
Examples	421
Error messages	421

Print JVM Job (PRTJVMJOB) 423

Parameters	423
Job name (JOB)	423
Option (OPTION)	424
Examples	424
Error messages	424

Print PEX Report (PRTPEXRPT) 427

Parameters	427
Member (MBR)	428
Library (LIB)	429
Type (TYPE)	429
Output (OUTPUT)	429
File to receive output (OUTFILE)	430
Output member options (OUTMBR)	430
Trace options (TRACEOPT)	431
Select trace type (TRCTYPE)	432
Time period for report (PERIOD)	433
Select jobs (SLTJOB)	434
Omit jobs (OMTJOB)	435
Stats options (STATSOPT)	435
Profile options (PROFILEOPT)	436
Order (ORDER)	437
Number of threads (NBRTHD)	437

Task information (TASKINF)	438
Examples	438
Error messages	438

Print Profile Internals (PRTPRFINT) 441

Parameters	442
Select by (SELECT)	442
User profile (USRPRF)	443
Percent full (PCTFULL)	443
Examples	443
Error messages	443

Print Publicly Auth Objects (PRTPUBAUT) 445

Parameters	446
Object type (OBJTYPE)	447
Changed report only (CHGRPTONLY)	447
Library (LIB)	447
Print file authority (FILAUT)	448
Print command authority (CMDAUT)	448
Print program authority (PGMAUT)	449
Print JOB authority (JOBDAUT)	449
Folder (FLR)	449
Directory (DIR)	449
Search subdirectory (SCHSUBDIR)	450
Examples	450
Error messages	450

Print Private Authorities (PRTPVTAUT) 451

Parameters	453
Object type (OBJTYPE)	453
Changed report only (CHGRPTONLY)	453
Library (LIB)	454
Authority type (AUTTYPE)	454
Folder (FLR)	454
Print AUTL objects (AUTLOBJ)	454
Directory (DIR)	455
Search subdirectory (SCHSUBDIR)	455
Examples	455
Error messages	455

Print Queue Authority (PRTQAUT) 457

Parameters	457
Library (LIB)	458
Changed report only (CHGRPTONLY)	459
Examples	459
Error messages	459

Print Subsystem Description (PRTSBSDAUT) 461

Parameters	462
Library (LIB)	462
Changed report only (CHGRPTONLY)	463
Examples	463
Error messages	463

Print SQL Information (PRTSQLINF) 465

Parameters	465
Object (OBJ)	465

Object type (OBJTYPE)	466
Examples	466
Error messages	466
Print Stop Word List (PRTSWL)	467
Parameters	467
Language ID (LANGID)	467
Stop word list type (TYPE)	467
Examples	467
Error messages	468
Print System Information (PRTSYSINF)	469
Parameters	469
Examples	469
Error messages	469
Print System Security Attr (PRTSYSSECA)	471
Parameters	471
Examples	471
Error messages	471
Print Point-to-Point Profile (PRTTCPPTP)	473
Parameters	473
Configuration profile (CFGPRF)	473
Examples	473
Error messages	473
Print Trace Data (PRTRC)	475
Parameters	475
Data member (DTAMBR)	476
Data library (DTALIB)	476
Select jobs (SLTJOB)	476
Delete trace (DLTTRC)	477
Sort by (SORT)	477
Output (OUTPUT)	477
File to receive output (OUTFILE)	478
Output member options (OUTMBR)	478
Examples	479
Error messages	479
Print Trigger Programs (PRTRGPGM)	481
Parameters	481
Library (LIB)	481
Changed report only (CHGRPTONLY)	482
Examples	483
Error messages	483
Print User Objects (PRTUSROBJ)	485
Parameters	485
Library (LIB)	485
Changed report only (CHGRPTONLY)	486
Examples	486
Error messages	486
Print User Profile (PRTUSRPRF)	487

Parameters	488
Type of information (TYPE)	489
Select by (SELECT)	489
Special authorities (SPCAUT)	489
User class (USRCLS)	490
Examples	490
Error messages	491
Power Down System (PWRDWNSYS)	493
Parameters	494
How to end (OPTION)	494
Controlled end delay time (DELAY)	495
Restart after power down (RESTART)	495
IPL source (IPLSRC)	496
Image catalog (IMGCLG)	496
End subsystem option (ENDSBSOPT)	497
Timeout option (TIMOUTOPT)	497
Confirm (CONFIRM)	498
Examples	498
Error messages	499
Query Document Library (QRYDOCLIB)	501
Parameters	501
Query definition (QRYDFN)	503
In folder (FLR)	506
Search subfolder (SCHSUBFLR)	507
Query text (QRYTXT)	507
Language ID (TXTLANGID)	509
Document list (DOCL)	509
Text (TEXT)	509
File to receive output (OUTFILE)	509
Output member options (OUTMBR)	510
Type of data for output (OUTDTATYP)	510
User identifier (USRID)	511
Time limit (TIMLMT)	512
Selection limit (SELLMT)	512
Order by (ORDER)	512
Command character identifier (CMDCHRID)	514
Examples	515
Error messages	515
Query Distributions (QRYDST)	517
Parameters	517
Incoming or outgoing (OPTION)	518
User identifier (USRID)	518
Delete status (DLTSTS)	518
File to receive output (OUTFILE)	518
Output member options (OUTMBR)	519
Status (STATUS)	520
Command character identifier (CMDCHRID)	520
Examples	521
Error messages	521
Query Problem Status (QRYPRBSTS)	523
Parameters	523
Problem identifier (PRBID)	523
Origin (ORIGIN)	524
Remote control point (RMTCPNAME)	524
Remote network identifier (RMTNETID)	524

Service number (SRVID)	524
Branch number (BRANCH)	525
Country or region number (COUNTRY)	525
Auto problem create (AUTOPRCRT)	525
Examples	525
Error messages	526

Query TIE Files (QRYTIEF) 529

Parameters	529
Examples	529
Error messages	529

Start QSH (QSH) 531

Error messages for STRQSH	532
Parameters	532
Command (CMD)	532
Examples	533
Error messages	533

Qualifier Definition (QUAL) 535

Parameters	535
Type of value (TYPE)	536
Length specification (LEN)	537
Constant value (CONSTANT)	538
Restricted values (RSTD)	538
Default value (DFT)	539
Valid values (VALUES)	539
Relational expression (REL)	540
Range of values (RANGE)	540
Special values (SPCVAR)	540
Minimum values required (MIN)	541
Allow unprintable characters (ALWUNPRT)	541
Allow variable names (ALWVAR)	541
Full field required (FULL)	541
Value an expression (EXPR)	542
Varying length (VARY)	542
Pass attribute byte (PASSATR)	542
Display input (DSPINPUT)	543
Choice text (CHOICE)	543
Choice program (CHOICEPGM)	544
Initial prompt length (INLPMTLEN)	544
Prompt text or message ID (PROMPT)	545
Examples	545
Error messages	546

Reclaim Activation Group (RCLACTGRP) 547

Parameters	548
Activation group (ACTGRP)	548
Close option (OPTION)	548
Examples	548
Error messages	549

Reclaim DB Cross-Reference (RCLDBXREF) 551

Parameters	551
Option (OPTION)	551
Library (LIB)	552
Examples	552

Error messages	553
--------------------------	-----

Reclaim DDM Conversations (RCLDDMCNV) 555

Parameters	555
Examples	555
Error messages	555

Reclaim Document Lib Object (RCLDLO) 557

Parameters	557
Document library object (DLO)	558
In folder (FLR)	558
System object name (SYSOBJNAM)	559
System object attributes (SYSOBJATR)	559
Auxiliary storage pool ID (ASP)	559
File to receive output (OUTFILE)	559
Output member options (OUTMBR)	560
Examples	560
Error messages	562

Reclaim Library (RCLLIB) 563

Parameters	563
Library (LIB)	563
Examples	563
Error messages	564

Reclaim Object Links (RCLLNK) 565

Parameters	566
Object (OBJ)	566
Directory subtree (SUBTREE)	566
Damaged object option (DMGOBJOPT)	567
Examples	567
Error messages	568

Reclaim Objects by Owner (RCLOBJOWN) 571

Parameters	572
User profile (USRPRF)	572
Examples	572
Error messages	572

Reclaim Optical (RCLOPT) 573

Parameters	573
Optical media library (MLB)	573
Option (OPTION)	573
Volume identifier (VOL)	574
Rebuild directory index (DIR)	574
Examples	574
Error messages	575

Reclaim Resources (RCLRSC) 577

Parameters	578
Program level (LVL)	578
Close option (OPTION)	578
Examples	578
Error messages	581

Reclaim Spool Storage (RCLSPLSTG) 583

Parameters	583
Days (DAYS)	583
ASP group (ASPGRP)	584
Examples	584
Error messages	585

Reclaim Storage (RCLSTG) 587

Parameters	588
Estimate time required (ESTIMATE)	588
Select (SELECT)	589
Omit (OMIT)	589
ASP device or group (ASPDEV)	589
Examples	589
Error messages	591

Reclaim Temporary Storage (RCLTMPSTG) 593

Parameters	593
Library (LIB)	594
Days unused (DAYS)	595
Examples	595
Error messages	595

Receive Distribution (RCVDST). 597

Parameters	597
Distribution identifier (DSTID)	598
User identifier (USRID)	598
Document (DOC)	598
In folder (FLR)	599
File to receive output (OUTFILE)	599
Output member options (OUTMBR)	599
Type of data for output (OUTDTATYP)	600
Acknowledge receipt (ACKRCV)	601
Distribution ID extension (DSTIDEXN)	601
Keep in mail log (KEEP)	602
Command character identifier (CMDCHRID)	602
Examples	602
Error messages	603

Receive File (RCVF) 605

Parameters	605
Display device (DEV)	606
Record format (RCDFMT)	606
Open file identifier (OPNID)	606
Wait (WAIT)	606
Examples	607
Error messages	608

Receive Journal Entry (RCVJRNE) . . . 611

Parameters	612
Journal (JRN)	614
Program to receive entries (EXITPGM)	614
Journalled physical file (FILE)	617
Objects (OBJ)	618
Objects (OBJPATH)	620
Directory subtree (SUBTREE)	621
Name pattern (PATTERN)	622
Range of journal receivers (RCVRNG)	623

Starting large sequence number (FROMENTLRG)	624
Starting date and time (FROMTIME)	624
Ending large sequence number (TOENTLRG)	624
Ending date and time (TOTIME)	625
Number of journal entries (NBRENT)	625
Journal codes (JRNCDE)	626
Journal entry types (ENTTYP)	626
Job name (JOB)	627
Program (PGM)	627
User profile (USRPRF)	627
Commit cycle large identifier (CCIDLRG)	628
Dependent entries (DEPENT)	628
File identifier (OBJFID)	628
Object journal identifier (OBJJID)	629
Entry format (ENTFMT)	630
Format minimized data (FMTMINDTA)	633
Null value indicators length (NULLINDLEN)	633
Delay time (DELAY)	656
Block length (BLKLEN)	657
Journal entry format (JRNENTMFT)	657
Return pointers (RTNPTR)	658
Include entries (INCENT)	658
Starting sequence number (FROMENT)	659
Ending sequence number (TOENT)	659
Commit cycle identifier (CMTCYCID)	660
Examples	660
Error messages	661

Receive Message (RCVMSG). 663

Parameters	663
Call stack entry message queue (PGMQ)	664
Message queue (MSGQ)	666
Message type (MSGTYPE)	666
Message key (MSGKEY)	667
Wait time (WAIT)	668
Remove message (RMV)	668
Coded character set ID (CCSID)	669
Reject default reply (RJTDFTRPY)	669
CL var for KEYVAR (4) (KEYVAR)	670
CL var for 1st level text (MSG)	670
CL var for MSGLEN (5 0) (MSGLEN)	670
CL var for 2nd level text (SECLVL)	670
CL var for SECLVLEN (5 0) (SECLVLEN)	671
CL var for msg data (MSGDTA)	671
CL var for MSGDTALEN (5 0) (MSGDTALEN)	671
CL var for MSGID (7) (MSGID)	671
CL var for SEV (2 0) (SEV)	671
CL var for SENDER (80) (SENDER)	671
Sender format (SENDERFMT)	672
CL var for RTNTYPE (2) (RTNTYPE)	674
CL var for ALROPT (9) (ALROPT)	674
CL var for MSGF (10) (MSGF)	675
CL var for MSGFLIB (10) (MSGFLIB)	675
CL var for SNDMSGFLIB (10) (SNDMSGFLIB)	675
CL var for text CCSID (5 0) (TXTCCSID)	675
CL var for data CCSID (5 0) (DTACCSID)	676
Examples	676
Error messages	677

Receive Network File (RCVNETF). . . 681

Parameters	681
From file (FROMFILE)	682
To file (TOFILE)	682
Member to be received (FROMMBR)	682
To member (TOMBR).	683
Replace or add records (MBROPT)	683
File number (NBR)	683
User (USER).	683
From file type (FROMTYPE)	684
Examples.	684
Error messages	685

Receive TIE File (RCVTIEF) 687

Parameters	687
Library (LIB)	687
File type (TYPE)	687
Output (OUTPUT).	688
Maximum records (MAXRCDS)	688
Examples.	688
Error messages	688

Remove Directory (RD) 689

Parameters	690
Directory (DIR).	690
Directory subtree (SUBTREE)	690
Remove link (RMVLNK)	691
Examples.	691
Error messages	692

Rename Object (REN). 695

Parameters	695
Object (OBJ).	696
New object (NEWOBJ)	696
Examples.	696
Error messages	696

Appendix. Notices 699

Programming interface information	700
Trademarks	701
Terms and conditions.	702
Code license and disclaimer information	702

Initialize System i Access (INZPCS)

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

Parameters
Examples
Error messages

The Initialize System i Access (INZPCS) command allows you to establish an operating environment for System i Access applications by creating various control documents on the System i Access folders. These control documents include code page mapping tables, keyboard tables, and font files used for displaying information on the personal computer display.

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 INZPCS

*ESCAPE Messages

IWS16D0

Initialize System i Access (INZPCS command) failed.

IWS16DD

Error getting message &1 from message file &2 in library &3.

IWS16E1

INZPCS command successfully completed.

IWS16E2

Error retrieving data area &1 in library &2.

IWS16E3

Error creating data area QINZPCSDA in library QUSRSYS.

IWS16E4

Error updating data area QINZPCSDA in library QUSRSYS.

IWS16EE

Failed to delete data area &1 in library &2.

Top

Parameters

Keyword	Description	Choices	Notes
KBDTYPE	Keyboard type	*DFT, AGB, AGI, ALI, BGB, BLI, BRB, CAB, CAI, CLB, CSB, DMB, DMI, ESB, FNB, FNI, FAB, FAI, FQB, FQI, GKB, GNB, HNB, ICB, ICI, INB, INI, IRB, ITB, ITI, JEB, JEL, JKB, JPB, JUB, KAB, KOB, LAB, LTB, LVB, MKB, NCB, NEB, NEI, NWB, NWI, PKB, PLB, PRB, PRI, RCB, RMB, RUB, SKB, SPB, SPI, SQB, SSB, SSI, SWB, SWI, SFI, SGI, TAB, THB, TKB, TRB, UAB, UKB, UKI, USB, USI, VNB, YGI	Optional, Positional 1
ASCII	ASCII code page number	*DFT, 437, 720, 737, 775, 813, 819, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863, 864, 865, 866, 868, 869, 874, 891, 897, 903, 904, 912, 915, 916, 920, 921, 922, 1004, 1006, 1008, 1040, 1041, 1042, 1043, 1046, 1088, 1089, 1098, 1114, 1115, 1124, 1125, 1127, 1129, 1131, 1133	Optional

Keyword	Description	Choices	Notes
EBCDIC	EBCDIC code page number	* DFT , 037, 256, 273, 277, 278, 280, 284, 285, 290, 297, 420, 423, 424, 500, 833, 836, 838, 870, 871, 875, 880, 918, 1025, 1026, 1027, 1047, 1097, 1112, 1122, 1123, 1130, 1132	Optional
LANGUAGE	Language feature code	* DFT , *ALL, 2902, 2903, 2904, 2905, 2906, 2909, 2911, 2912, 2913, 2914, 2922, 2923, 2924, 2925, 2926, 2928, 2929, 2930, 2931, 2932, 2933, 2937, 2938, 2939, 2940, 2942, 2950, 2954, 2956, 2957, 2958, 2961, 2962, 2963, 2966, 2972, 2974, 2975, 2976, 2978, 2979, 2980, 2981, 2984, 2986, 2987, 2989, 2992, 2994, 2995, 2996, 2998	Optional

Top

Keyboard type (KBDTYPE)

Specifies the keyboard type used.

The possible values are:

***DFT** The default keyboard type is used. When the command is run initially, the default value comes from the system value QKBDTYPE. When the command is run after the initial time, the default takes the value specified at the previous running of the command.

keyboard-type

Specify the 3-character keyboard type to use. Values are listed in the Character Translation Table.

Character Translation Table

Identifier

Language

AGB	Austria/Germany
AGI	Austria/Germany Multinational
ALI	Albania
BGB	Bulgaria
BLI	Belgium Multinational
BRB	Brazil
CAB	Canadian/French
CAI	Canadian/French Multinational
CLB	Arabic X/Basic
CSB	Czech Republic
DMB	Denmark
DMI	Denmark Multinational
ESB	Estonia
FNB	Finland/Sweden
FNI	Finland/Sweden Multinational
FAB	France (Azerty)
FAI	France (Azerty) Multinational
FQB	France (Qwerty)

FQI France (Qwerty) Multinational
GKB Greece
GNB Greece
HNB Hungary
ICB Iceland
ICI Iceland Multinational
INB International
INI International Multinational
IRB Iran (Farsi)
ITB Italy
ITI Italy Multinational
JEB Japan (English)
JEI Japan (English) Multinational
JKB Japan (Kanji)
JPB Japan (Latin Extended)
JUB Japan (U.S. Basic)
KAB Japan (Katakana)
KOB Korea
LAB Laos
LTB Lithuania
LVB Latvia
MKB FYR Macedonia Former Yugoslavia Republic
NCB Hebrew
NEB Netherlands
NEI Netherlands Multinational
NWB Norway
NWI Norway Multinational
PKB Pakistan (Urdu)
PLB Poland
PRB Portugal
PRI Portugal Multinational
RCB Simplified Chinese
RMB Romania
RUB Russia
SKB Slovakia
SPB Spain
SPI Spain Multinational

SQB	Serbia (Cyrillic)
SSB	Spanish Speaking
SSI	Spanish Speaking Multinational
SWB	Sweden
SWI	Sweden Multinational
SFI	Switzerland/France Multinational
SGI	Switzerland/Germany Multinational
TAB	Traditional Chinese
THB	Thailand (used only with language 2924)
TKB	Turkey (Qwerty)
TRB	Turkey (F)
UAB	Ukraine
UKB	United Kingdom
UKI	United Kingdom Multinational
USB	United States/Canada
USI	United States/Canada Multinational
VNB	Vietnam
YGI	Languages of the Former Yugoslavia (Latin)

Top

ASCII code page number (ASCII)

Specifies the ASCII code page number to use.

Note: When you are running the INZPCS command for a double byte language, use the single byte code page for this language. INZPCS only needs to process single byte code pages. Double byte code page support is available without the use of INZPCS.

The possible values are:

***DFT** The default code page number is used. When the command is run initially, the default value is 437 for keyboard types USB and USI, and 850 for most others. When the command is run after the initial time, the default takes the value specified at the previous running of the command.

code-page-number

Specify the ASCII code page number to use.

Top

EBCDIC code page number (EBCDIC)

Specifies the EBCDIC (or host) code page number to use.

Note: When you are running the INZPCS command for a double byte language, use the single byte code page for this language. INZPCS only needs to process single byte code pages. Double byte code page support is available without the use of INZPCS.

***DFT** The default system code page number is used. When the command is run initially, the default value comes from the code page portion of the system value QCHRID. When the command is run after the initial time, the default takes the value specified at the previous running of the command.

code-page-number

Specify the EBCDIC (or host) page number to use.

Top

Language feature code (LANGUAGE)

Specifies the language feature identifier (ID) of the secondary language to be processed.

The possible values are:

***DFT** The primary language of System i Access should be processed.

language-feature-code

Specify the language feature code of the language to process. Values are listed in the Language Feature Identifier Table.

Language Feature Identifier Table

Identifier

Language

2902	Estonian
2903	Lithuanian
2904	Latvian
2905	Vietnamese
2906	Lao
2909	Belgian English
2911	Slovenian
2912	Croatian
2913	Macedonian
2914	Serbian
2922	Portuguese
2923	Dutch
2924	English
2925	Finnish
2926	Danish
2928	French
2929	German
2930	Japanese Universal
2931	Spanish
2932	Italian
2933	Norwegian

- 2937 Swedish
- 2938 English Uppercase/DBCS
- 2939 German-MNCS
- 2940 French-MNCS
- 2942 Italian-MNCS
- 2950 English Uppercase
- 2954 Arabic
- 2956 Turkish
- 2957 Greek
- 2958 Icelandic
- 2961 Hebrew
- 2962 Japanese (DBCS)
- 2963 Belgium Dutch
- 2966 Belgium French
- 2972 Thai
- 2974 Bulgarian
- 2975 Czech
- 2976 Hungarian
- 2978 Polish
- 2979 Russian
- 2980 Brazilian Portuguese
- 2981 Canadian French
- 2984 English U/L (DBCS)
- 2986 Korean (DBCS)
- 2987 Traditional Chinese (DBCS)
- 2989 Simplified Chinese (DBCS)
- 2992 Romanian
- 2994 Slovakian
- 2995 Albanian
- 2996 Portuguese-MNCS
- 2998 Farsi

[Top](#)

Examples

None

[Top](#)

Error messages

*ESCAPE Messages

IWS16D0

Initialize System i Access (INZPCS command) failed.

IWS16DD

Error getting message &1 from message file &2 in library &3.

IWS16E1

INZPCS command successfully completed.

IWS16E2

Error retrieving data area &1 in library &2.

IWS16E3

Error creating data area QINZPCSDA in library QUSRSYS.

IWS16E4

Error updating data area QINZPCSDA in library QUSRSYS.

IWS16EE

Failed to delete data area &1 in library &2.

Top

Initialize Physical File Mbr (INZPFM)

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

Parameters
Examples
Error messages

The Initialize Physical File Member (INZPFM) command initializes records in a member of a physical file to the specified type of record (either default or deleted records). If the initialized member is empty, records are added and initialized to the specified type; if the member is not empty, records of the specified type are added to the member. As many records are added as is necessary to make the total record count specified.

Restrictions:

- This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Physical file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST, *LAST</i>	Optional, Positional 2
RECORDS	Initialize records as	<i>*DFT, *DLT</i>	Optional, Positional 3
TOTRCDS	Total number of records	1-4294967288, <i>*NXTINCR</i>	Optional

Top

Physical file (FILE)

Specifies the physical file that contains the member to be initialized.

This is a required parameter.

Qualifier 1: Physical file

name Specify the name of 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 used to locate the file. If no library is specified as the current library, QGPL is used.

name Specify the name of the library to be searched.

Member (MBR)

Specifies the file member to be initialized.

*FIRST

The first member of the specified file is used.

*LAST

The last member of the specified physical file is initialized.

name Specify the name of the physical file member to be initialized.

Top

Initialize records as (RECORDS)

Specifies the type of records that are initialized or added to the specified member. The records in the member are initialized as default records or deleted records.

*DFT The records in the member are initialized as default records. If a default value was specified in the DDS (DFT keyword) for a field, that field is initialized to the specified default; otherwise, all numeric fields are initialized to zeros and all character fields are initialized to blanks.

*DLT The records in the member are initialized as deleted records. The records are not eligible for access, but simply hold a place in the file. Deleted records are changed to reuse the deleted space.

Top

Total number of records (TOTRCDS)

Specifies the total number of records in the member after it is initialized. If the value specified in this parameter causes the size of the file to be larger than the size specified when the file was created, a message is sent to the system operator's message queue (QSYSOPR). The operator can either continue or cancel the operation.

*NXTINCR

The number of records in the member is increased to extend the file to the next allocation amount added. If the member is empty, records are added to meet the initial allocation specified for the member. *NXTINCR is not valid if *NOMAX was specified for the **Member size (SIZE)** parameter, when the file is created.

1-4294967288

Specify the total number of records you want the the member to have. If the number of existing records in the member already meets or is larger than this number, no records are initialized; if the number is less than that specified, enough records are initialized to equal the total specified.

Top

Examples

```
INZPFM FILE(*CURLIB/INV) TOTRCDS(12000)
```

This command initializes as many as 12,000 records in the first member of the physical file named INV in the job's current library *CURLIB. Only the number of records are added that brings the total to 12,000 records in the member. Any records that are added are initialized to the default format. If a default value is specified in the DDS (DFT keyword) for a field, that field is initialized to the specified default; otherwise, all numeric fields are initialized to zeros and all character fields are initialized to blanks.

Top

Error messages

*ESCAPE Messages

CPF3130

Member &2 already in use.

CPF3131

Cannot initialize member &2 with default records.

CPF3132

TOTRCDS parameter value either missing or too small.

CPF3133

File &1 in library &3 contains no members.

CPF3134

Referential constraint error processing member &2.

CPF3136

File &1 in &3 not allowed on command.

CPF3137

No authority to clear, initialize, or copy member &2.

CPF3138

Check constraint error processing member &2.

CPF3140

Initialize or copy of member &2 canceled.

CPF3141

Member &2 not found.

CPF3142

File &1 in library &3 not found.

CPF3143

Increments not allowed for member &2.

CPF3144

Member &2 not cleared or initialized.

CPF3148

New records need too much space for member &2.

CPF3156

File &1 in library &3 in use.

CPF3157

Triggers prevent requested operation.

CPF3159

Member &2 saved with STG(*FREE).

- CPF3160**
Operation on member &2 ended. Entry cannot be journaled.
- CPF3179**
Cannot clear or initialize DDM file &1 in &3.
- CPF3180**
Member &2 not initialized.
- CPF32CF**
Distributed file error, reason code &3.
- CPF32C3**
Distributed file error, level ID mismatch
- CPF320B**
Operation was not valid for database file &1.
- CPF9801**
Object &2 in library &3 not found.
- CPF9810**
Library &1 not found.
- CPF9820**
Not authorized to use library &1.

Top

Initialize System (INZSYS)

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

Parameters
Examples
Error messages

The Initialize System (INZSYS) command initializes conversions done during installation procedures. This process is initiated during the first IPL after the software package is installed.

More information is available in the Installing, upgrading, or deleting i5/OS and related software book, SC41-5120.

Top

Parameters

Keyword	Description	Choices	Notes
MSGQ	Message queue	Single values: *SYSOPR Other values: <i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL , *CURLIB	

Top

Message queue (MSGQ)

Specifies the name and library of the message queue to which messages are sent and from which they are shown.

***SYSOPR**

Messages from the system operator message queue (QSYSOPR) are sent to the system operator.

message-queue-name

Specify the name of the message queue from which messages are shown.

The possible library values are:

***LIBL** The library list is used to locate the message queue.

***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, the QGPL library is used.

library-name

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

Top

Examples

INZSYS

This command initializes the conversions done during installation procedures.

[Top](#)

Error messages

*ESCAPE Messages

CPF372A

INZSYS or GO LICPGM currently running in another job.

CPF90E2

Error occurred for previous release file &1 in library &2.

CPF90E3

Error occurred for file &1 in library &2.

CPF90E4

System function in use. Reason code &1.

CPF90E8

Error occurred for file &1 in library &2.

CPF90E9

Data exists for more than one previous release.

[Top](#)

Initialize Tape (INZTAP)

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

Parameters
Examples
Error messages

The Initialize Tape (INZTAP) command is used to initialize magnetic tapes for use on the system. This command is used to initialize a tape with a standard volume label for standard label magnetic tape processing, or to initialize a tape with no labels for unlabeled magnetic tape processing.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i>	Required, Positional 1
NEWVOL	New volume identifier	<i>Character value</i> , *NONE, *CTGID	Optional, Positional 2
NEWOWNID	New owner identifier	<i>Character value</i> , *BLANK	Optional, Positional 3
VOL	Volume identifier	<i>Character value</i> , *MOUNTED	Optional
CHECK	Check for active files	*YES, *NO, *FIRST	Optional
DENSITY	Tape density	<i>Character value</i> , *DEVTYPE, *CTGTYPE, *FMT3480, *FMT3490E, *FMT3570, *FMT3570E, *FMT3590, *FMT3590E, *FMT3590H, *QIC120, *QIC525, *QIC1000, *QIC2GB, *QIC2DC, *QIC4GB, *QIC4DC, *QIC3040, *QIC5010, *MLR3, *SLR60, *SLR100, *FMT2GB, *FMT5GB, *FMT7GB, *FMT20GB, *FMT60GB, *ULTRIUM1, *ULTRIUM2, *VXA1, *VXA2, 1600, 3200, 6250	Optional
CODE	Code	*EBCDIC, *ASCII	Optional
ENDOPT	End of tape option	*REWIND, *UNLOAD	Optional
CLEAR	Clear	*NO, *YES	Optional

Top

Device (DEV)

Specifies the name of the device in which the volume being initialized for use is placed. Specify the name of the tape or media library device.

This is a required parameter.

Top

New volume identifier (NEWVOL)

Specifies the volume identifier for a tape being initialized for use as a standard labeled tape. If no volume identifier is specified, the tape is initialized for use as an unlabeled tape.

*NONE

The tape is initialized for use as an unlabeled tape. Only tape marks are used to indicate the beginning and end of each data file on it, and the beginning and end of the volume itself.

*CTGID

The tape is initialized as a standard labeled tape. The new logical volume identifier is the same as the external identifier of the tape cartridge. Each tape within a library device must have a unique external identifier.

character-value

Specify no more than 6 characters to identify the new volume. The identifier must contain only alphanumeric characters (A through Z, \$, #, @, and 0 through 9), and cannot have a prefix or contain blanks.

Top

New owner identifier (NEWOWNID)

Specifies the identifier of the tape owner to write in the volume label.

*BLANK

Text is not specified.

character-value

Specify no more than 14 characters that identify the owner of the tape. If fewer than 14 characters are specified, the field is left-justified and padded on the right with blanks.

Top

Volume identifier (VOL)

Specifies the existing volume identifier of the tape being initialized for use or indicates that the tape currently on the magnetic tape unit should be initialized for use.

Note: If the device specified is a media library device, or a virtual tape device, then the volume specified should be the cartridge identifier or virtual tape volume name to be mounted and used.

*MOUNTED

Any labeled or unlabeled volume that is placed in the specified tape device is initialized for use. To initialize a new or empty volume for use, *MOUNTED must be specified, and *NO must be specified on the **Check for active files** prompt (CHECK parameter). For a media library device, the volume to be used is the next cartridge in the category mounted by the Set Tape Category (SETTAPCGY) command. For a virtual tape device, the volume to be used is the currently mounted one, or if there is not a currently mounted volume, the next volume in loaded status in the image catalog will be used.

character-value

Specify the identifier of the labeled volume being initialized for use. This parameter value can be used only to initialize a tape for use that is already a labeled volume. If the tape on the specified device has a different volume identifier than the one specified or if it is an unlabeled volume, an error message is sent.

Top

Check for active files (CHECK)

Specifies whether a labeled tape volume should be checked for active data files before it is initialized for use. If an unlabeled volume is placed in the specified device, this parameter is ignored.

***YES** All data file labels on the tape are checked. If any active files are found, the operation is ended and an error message is sent.

***NO** Tape initialization continues with no checking for active files. To initialize a new or empty volume for use, ***NO** must be specified here and ***MOUNTED** must be specified on the **Volume identifier** prompt (VOL parameter).

***FIRST**

Only the first data file label on the tape is checked. If there are no data files on the volume or if the first data file has expired, the volume is initialized for use without checking for any other files on the tape. If the first data file has not expired, the operation is ended and an error message is sent.

Top

Tape density (DENSITY)

Specifies the recording format of the data to be written on the tape.

***DEVTYPE**

The highest capacity density or format supported by the tape device will be used.

Device

Highest capacity density or format

3480 *FMT3480

3490E *FMT3490E

3570-Bxx
*FMT3570

3570-Cxx
*FMT3570E

3580-001
*ULTRIUM1

3580-002
*ULTRIUM2

3580-003
*ULTRIUM3

3580-004
*ULTRIUM4

3590-Bxx
*FMT3590

3590-Exx
*FMT3590E

3590-Hxx
*FMT3590H

3592-E05
 *FMT3592A2

3592-J1A
 *FMT3592A1

4685-001
 *VXA2

5755 *ULTRIUM2

6258 *DAT72

6279 *VXA3

6344 *QIC2GB

6349 *QIC2GB

6369 *QIC2GB

6380 *QIC2GB

6381 *QIC2DC

6382 *QIC4DC

6383 *QIC5010

6384 *SLR60

6386 *MLR3

6387 *SLR100

6390 *FMT7GB

63B0 *VRT256K

7207-122
 *QIC4DC

7208-002
 *FMT2GB

7208-012
 *FMT5GB

7208-222
 *FMT7GB

7208-342
 *FMT20GB

7208-345
 *FMT60GB

9348 6250

*CTGTYPE

The highest capacity density or format supported by the device for the mounted cartridge type will be used. If the device does not support special cartridge type information, *DEVTYPE is used.

character-value

Specify the density or format to use.

1600 The data density on the tape volume is 1,600 bits per inch, which is used for 1/2 inch reel tapes.

3200 The data density on the tape volume is 3,200 bits per inch, which is used for 1/2 inch reel tapes.

6250 The data density on the tape volume is 6,250 bits per inch, which is used for 1/2 inch reel tapes.

***DAT72**

The format of this tape is DAT72. It is used by 4mm cartridge tape devices that can store 36 gigabytes of data on a standard length cartridge.

***DDS3**

The format of this tape is DDS3. It is used by 4mm cartridge tape devices that can store 12 gigabytes of data on a standard length cartridge.

***DDS4**

The format of this tape is DDS4. It is used by 4mm cartridge tape devices that can store 20 gigabytes of data on a standard length cartridge.

***FMT3480**

The format of this tape is FMT3480. The data density on this tape volume is formatted to support a 3480 device. This density is used for 1/2 inch cartridge tapes.

***FMT3490E**

The format of this tape is FMT3490E. The data density on this tape volume is formatted to support a 3490E device. This density is used for 1/2 inch cartridge tapes.

***FMT3570**

The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.

***FMT3570E**

The format of this tape is FMT3570E. The data format is written on the tape volume with a 3570E device.

***FMT3590**

The format of this tape is FMT3590. The data format is written on the tape volume with a 3590 device. This density is used for 1/2 inch cartridge tapes.

***FMT3590E**

The format of this tape is FMT3590E. The data format is written on the tape volume with a 3590E device. This density is used for 1/2 inch cartridge tapes.

***FMT3590H**

The format of this tape is FMT3590H. The data format is written on the tape volume with a 3590H device. This density is used for 1/2 inch cartridge tapes.

***FMT3592A1**

The format of this tape is FMT3592A1. It is used by 3592 tape devices that can store 300 gigabytes of data on a standard length cartridge.

FMT3592A1E

The format of this tape is FMT3592A1E. It is used by 3592 tape devices that can store 300 gigabytes of encrypted data on a standard length cartridge.

***FMT3592A2**

The format of this tape is FMT3592A2. It is used by 3592 tape devices that can store 500 gigabytes of data on a standard length cartridge.

FMT3592A2E

The format of this tape is FMT3592A2E. It is used by 3592 tape devices that can store 500 gigabytes of encrypted data on a standard length cartridge.

***QIC120**

The format of this tape is QIC120, which is used for 1/4 inch cartridge tapes that can hold 120 megabytes of data.

***QIC525**

The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

***QIC1000**

The format of this tape is QIC1000, which is used for 1/4 inch cartridge tapes that can hold 1200 megabytes of data.

***QIC2GB**

The format of this tape is QIC2GB. It is used by 1/4 inch tape devices which can store 2.5 gigabytes of data on a standard length QIC2GB cartridge.

***QIC2DC**

The format of this tape is QIC2DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC2GB format.

***QIC4GB**

The format of this tape is QIC4GB. It is used by 1/4 inch tape devices which can store 4 gigabytes of data on a standard length QIC4GB cartridge.

***QIC4DC**

The format of this tape is QIC4DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC4GB format.

***QIC3040**

The format of this tape is QIC3040, which is used for 1/4 inch minicartridge tapes that can hold 840 megabytes of data.

***QIC5010**

The format of this tape is QIC5010, which is used for 1/4 inch cartridge tapes that can hold 13.5 gigabytes of data.

***MLR3**

The format of this tape is MLR3. It is used by 1/4 inch tape devices which can store 25 gigabytes of data on a standard length MLR3 cartridge.

***SLR60**

The format of this tape is SLR60. It is used by 1/4 inch tape devices which can typically store 60 gigabytes of compacted data on a standard length SLR60 cartridge.

***SLR100**

The format of this tape is SLR100. It is used by 1/4 inch tape devices which can typically store 100 gigabytes of compacted data on a standard length SLR100 cartridge.

***FMT2GB**

The format of this tape is FMT2GB, which is used for 8 millimeter cartridge tapes that can hold 2 gigabytes of data.

***FMT5GB**

The format of this tape is FMT5GB, which is used for 8 millimeter cartridge tapes that can hold 5 gigabytes of data.

***FMT7GB**

The format of this tape is FMT7GB, which is used for 8 millimeter cartridge tapes that can hold 7 gigabytes of data.

***FMT20GB**

The format of this tape is FMT20GB. It is used by 8 millimeter tape devices that can store 20 gigabytes of data on a standard length cartridge.

***FMT60GB**

The format of this tape is FMT60GB. It is used by 8 millimeter tape devices that can store 60 gigabytes of data on a standard length cartridge.

***ULTRIUM1**

The format of this tape is ULTRIUM1. It is used by 1/2 inch cartridge tape devices that can store 100 gigabytes of data on a standard length cartridge.

***ULTRIUM2**

The format of this tape is ULTRIUM2. It is used by 1/2 inch cartridge tape devices that can store 200 gigabytes of data on a standard length cartridge.

***ULTRIUM3**

The format of this tape is ULTRIUM3. It is used by 1/2 inch cartridge tape devices that can store 400 gigabytes of data on a standard length cartridge.

***ULTRIUM4**

The format of this tape is ULTRIUM4. It is used by 1/2 inch cartridge tape devices that can store 800 gigabytes of data on a standard length cartridge.

***VRT32K**

The format of the volume is VRT32K. It is used to write data to a virtual volume using a maximum data block size of 32KB. Volumes written using this format can be duplicated to all supported tape devices.

***VRT64K**

The format of the volume is VRT64K. It is used to write data to a virtual volume using a maximum data block size of 64KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 64KB or greater.

***VRT240K**

The format of the volume is VRT240K. It is used to write data to a virtual volume using a maximum data block size of 240KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 240KB or greater.

***VRT256K**

The format of the volume is VRT256K. It is used to write data to a virtual volume using a maximum data block size of 256KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 256KB or greater.

***VXA1**

The format of this tape is VXA1. It is used by VXA cartridge tape devices that can store 33 gigabytes of data on a standard length cartridge.

***VXA2**

The format of this tape is VXA2. It is used by VXA cartridge tape devices that can store 80 gigabytes of data on a standard length cartridge.

***VXA3**

The format of this tape is VXA3. It is used by VXA cartridge tape devices that can store 160 gigabytes of data on a standard length cartridge.

Note: Self-configured tape devices may define additional valid values for the density parameter. Use System i5 Navigator (Configuration and Service) (Hardware) (Tape Devices) (Tape Libraries) (Tape Resources) (Properties) or (Configuration and Service)(Hardware) (Tape Devices) (Stand-Alone Devices) (Properties) to find additional valid density values for a specific device, or use the F4=Prompt key on the "Tape density" field of the CL command to see a list of all valid density values for the attached tape devices.

Code (CODE)

Specifies the character code in which the volume label is written. All data that is not save data written after the label must be in the same code; codes cannot be intermixed on a tape that is not a save tape. If the tape is being initialized for use as an unlabeled tape with *NONE or no volume identifier specified on the **New volume identifier** prompt (NEWVOL parameter), this parameter is ignored.

*EBCDIC

The volume label is written in EBCDIC and is an IBM standard label; all additional data must also be written in EBCDIC.

*ASCII

The volume label is written in ASCII and is an ANSI standard label; all additional data must also be written in ASCII.

Top

End of tape option (ENDOPT)

Specifies whether the tape is rewound only or rewound and unloaded after the operation ends.

*REWIND

The tape is automatically rewound, but not unloaded, after the operation has ended.

*UNLOAD

The tape is automatically rewound and unloaded after the operation ends.

Top

Clear (CLEAR)

Specifies whether all previous labels and data are deleted from the tape when it is initialized. If the volume must be cleared of all data, it is spaced from the location of the initializing volume label or tape markers to the end of the tape marker.

*NO Existing data is not deleted. Even though the existing data is not deleted, the data on the volume is not accessible after the volume has been initialized for use.

*YES After the beginning of the tape has been initialized for use, the rest of the data on the tape is deleted. The *YES value is needed only if there are security concerns with the old data. If *YES is selected, the initialize operation can take a long time.

Top

Examples

```
INZTAP  DEV(TAPE1) NEWVOL(T00100) CHECK(*NO) CODE(*ASCII)
        ENDOPT(*UNLOAD)
```

This command initializes the volume on the tape device named TAPE1 using the ASCII character code. Its new volume identifier is T00100, regardless of whether it contains a valid volume identifier or files that have not ended (active field). Once the volume has been initialized, the tape is rewound and unloaded. Any previous data beyond the new volume label is not deleted, but is no longer accessible.

Top

Error messages

*ESCAPE Messages

CPF67A0

Volume ID does not match cartridge ID

CPF6702

Error processing volume on device &1.

CPF6708

Command ended due to error.

CPF6715

Error at beginning of tape on device &1.

CPF6718

Cannot allocate device &1.

CPF6720

Incorrect volume &2 found on device &1.

CPF6721

Device &1 not a tape device.

CPF6722

End of tape found on device &1.

CPF6745

Device &1 not a media library device.

CPF6750

NEWVOL(*NONE) not valid for device &1.

CPF6751

Load failure occurred on device &4.

CPF6754

Active file &4 found on volume &2.

CPF6760

Device &1 not ready.

CPF6762

Wrong type of cartridge in device &1.

CPF6763

Wrong type of cartridge in device &1.

CPF6768

Volume on device &1 is write protected.

CPF676B

Volume on device &4 is write protected.

CPF6772

Volume on device &1 cannot be processed.

CPF6774

New volume &2 is a nonstandard labeled tape. Volume not prepared.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

Iterate (ITERATE)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The Iterate (ITERATE) command interrupts the processing of commands in the associated DOWHILE, DOUNTIL, or DOFOR loop and passes control to the associated ENDDO. The conditional part of the DOWHILE, DOUNTIL, or DOFOR will be evaluated and processing resume accordingly.

By specifying the optional command label, processing will skip to the ENDDO of the associated Do command group.

Restrictions:

- This command is valid only in a CL program or ILE CL procedure.
- This command is valid only inside a DOWHILE, DOUNTIL, or DOFOR command group.

Top

Parameters

Keyword	Description	Choices	Notes
CMDLBL	Command label	<i>Simple name</i> , <u>*CURRENT</u>	Optional, Positional 1

Top

Command label (CMDLBL)

The label must be within the same program as the ITERATE command and be a label on an active DOWHILE, DOUNTIL, or DOFOR group. A CL variable name cannot be used to specify the label name.

*CURRENT

Iterates on the innermost loop surrounding this ITERATE command.

simple-name

Specify the label name of the surrounding DOWHILE, DOUNTIL, or DOFOR command which is being iterated.

Top

Examples

```
DCL VAR(&INT) TYPE(*INT) LEN(2)
DCL VAR(&NAME) TYPE(*CHAR) LEN(10)
:
DOUNTIL COND(&INT *GT 100)
: (group of CL commands)
```

```
IF COND(&NAME *EQ *NONE) THEN(ITERATE)
CHGVAR VAR(&INT) VALUE(&INT + 1)
: (group of CL commands)
ENDDO /* ITERATE passes control to here */
```

Whenever the IF command evaluates the value of &NAME to be equal to *NONE the ITERATE is processed. Control will pass to the ENDDO command, the condition specified on the associated DOUNTIL is evaluated. If the value of &INT is 100 or less, the loop will be processed again. If the value of &INT is 101 or greater, control passes to the associated ENDDO.

[Top](#)

Error messages

None

[Top](#)

Copy From LDIF (LDIF2DB)

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

Parameters
Examples
Error messages

The Copy From LDIF (LDIF2DB) command is used to copy directory entries from a LDAP Data Interchange Format (LDIF) stream file to the directory for a Directory Server instance. The Directory Server provides a Lightweight Directory Access Protocol (LDAP) server on i5/OS. The command can only be used to add new entries to the directory, it cannot change or delete entries already in the directory.

Restriction: You must do or satisfy one of the following conditions to use this command:

- Have all object (*ALLOBJ) and input/output system configuration (*IOSYSCFG) special authorities.
- Supply the administrator distinguished name and password.
- Be a Directory Services administrator. The caller is a Directory Services administrator if the Directory Services server has been configured to grant administrator access to authorized users and the caller is authorized to the 'Directory Services Administrator' function of the operating system.

Top

Parameters

Keyword	Description	Choices	Notes
LDIFSTMF	LDIF stream file	<i>Path name</i>	Required, Positional 1
INSTANCE	Instance	<i>Name, <u>QUSRDIR</u></i>	Optional, Positional 2
ADMIN	Administrator	<i>Element list</i>	Optional
	Element 1: Distinguished name	<i>Character value</i>	
	Element 2: Password	<i>Character value</i>	
REPLICATE	Replicate imported data	<i><u>*YES</u>, *NO</i>	Optional

Top

LDIF stream file (LDIFSTMF)

Specifies the integrated file system path to the LDAP Data Interchange Format (LDIF) stream file.

This is a required parameter.

path-name

Specify the path name of the LDIF stream file that contains the directory entries to be copied into the directory for the Directory Server instance.

Top

Instance (INSTANCE)

Specifies the Directory Server instance name whose directory the directory entries are copied to.

QUSRDIR

The name of the system default Directory Server instance.

name Specify a Directory Server instance name. The name has a minimum of one character and a maximum of eight characters.

Top

Administrator (ADMIN)

Specifies the Directory Server administrator. If not specified, the user must have *ALLOBJ and *IOSYSCFG special authorities.

Element 1: Distinguished name

character-value

Specify the distinguished name for the Directory Server administrator, for example, cn=administrator. A maximum of 50 characters is allowed.

Element 2: Password

character-value

Specify the password for the Directory Server administrator. The password is case sensitive and must be enclosed in apostrophes. A maximum of 50 characters is allowed.

Top

Replicate imported data (REPLICATE)

Specifies whether copied directory entries should be replicated to replica Directory Servers. This option could be used, for example, when initializing an additional master server so that it does not attempt to replicate data to Directory Servers already containing these directory entries.

*YES Copied directory entries are replicated.

*NO Copied directory entries are not replicated.

Top

Examples

Example 1: Copy to the QUSRDIR Directory

```
LDIF2DB      LDIFSTMF('/1dap/qusrdir.ldif') INSTANCE(QUSRDIR)
```

This command copies the directory entries from the **qusrdir.ldif** stream file in the **ldap** directory to the Directory Server directory for the QUSRDIR instance. The user running the command this way must have all object (*ALLOBJ) and input/output system Configuration (*IOSYSCFG) special authorities.

Example 2: Copy to the QUSRDIR Directory and Replicate

```
LDIF2DB      LDIFSTMF('/1dap/qusrdir.ldif') INSTANCE(QUSRDIR)
              REPLICATE(*YES) ADMIN('cn=admin' 'secret')
```

This command copies the directory entries from the **qusrdir.ldif** stream file in the **ldap** directory to the Directory Server directory for the QUSRDIR instance. The copied data will be replicated to the replica Directory Servers.

Example 3: Copy to the DOGGIES Directory

```
LDIF2DB      LDIFSTMF('/ldap/doggies.ldif') INSTANCE(DOGGIES)
              ADMIN('cn=fluffy' 'poodle')
```

This command copies the directory entries from the **doggies.ldif** stream file in the **ldap** directory to the Directory Server directory for the DOGGIES instance.

Top

Error messages

*ESCAPE Messages

GLD0202

Administrator DN or password not correct.

GLD0213

Error opening or creating file.

GLD0215

Directory server instance &1 not found.

GLD0218

Not enough authority or incorrect distinguished name and password specified.

GLD0225

&1 items added to directory, &2 items not added.

GLD0226

Client request rejected. Directory server is read only.

Top

Leave (LEAVE)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The Leave (LEAVE) command ends the processing of commands in the associated DOWHILE, DOUNTIL, or DOFOR loop and passes control to the first command following the associated ENDDO command.

The following command sequence shows this flow.

```
L1:   DOWHILE &LGL1
      ...
L2:   DOWHILE &LGL2
      ...
      IF &LGL3 (LEAVE CMDLBL(L1))
      IF &LGL4 LEAVE
      ...
      ENDDO
      /* Here if &LGL4 evaluates to true */
      ...
      ENDDO
      /* Here if &LGL3 evaluates to true */
      ...
```

Restrictions:

- This command is valid only in a CL program or ILE CL procedure.
- This command is valid only inside a DOWHILE, DOUNTIL, or DOFOR command group.

Top

Parameters

Keyword	Description	Choices	Notes
CMDLBL	Command label	Simple name, <u>*CURRENT</u>	Optional, Positional 1

Top

Command label (CMDLBL)

The label must be within the same program as the LEAVE command and be a label on an active DOWHILE, DOUNTIL, or DOFOR group. A CL variable name cannot be used to specify the label name.

*CURRENT

Leaves the innermost loop surrounding this LEAVE command.

simple-name

Specify the label name of the surrounding DOWHILE, DOUNTIL, or DOFOR command which is being ended.

Examples

Example 1: Leave Simple DOFOR Loop

```
DCL VAR(&INT) TYPE(*INT) LEN(2)
DCL VAR(&NAME) TYPE(*CHAR) LEN(10)
:
DOFOR VAR(&INT) FROM(0) TO(10)
: (group of CL commands)
IF COND(&NAME *EQ *NONE) THEN(LEAVE)
: (group of CL commands)
ENDDO
```

The LEAVE command interrupts processing of the active DOFOR group and processing continues with command following the ENDDO.

Example 2: Leave with Nested Loops

```
DCL VAR(&INT) TYPE(*INT) LEN(2)
DCL VAR(&NAME) TYPE(*CHAR) LEN(10)
DCL VAR(&LGL) TYPE(*LGL) VALUE('1') /* True */
:
LOOP1: DOFOR VAR(&INT) FROM(0) TO(10)
: (group of CL commands)
LOOP2: DOUNTIL COND(&LGL)
: (group of CL commands)
IF COND(&NAME *EQ *NONE) THEN(LEAVE CMDLBL(LOOP1))
: (group of CL commands)
ENDDO /* DOUNTIL */
: (group of CL commands)
ENDDO /* DOFOR */
```

The LEAVE command interrupts processing of both the active DOUNTIL and DOFOR groups and processing continues with command following the ENDDO matching the DOFOR command.

Error messages

None

Link/Unlink Data Definition (LNKDTADFN)

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

Parameters
Examples
Error messages

The Link Data Definition (LNKDTADFN) command links or unlinks file definitions in a dictionary, program-described files, or externally-described files.

Restriction: A file cannot be linked if it is already linked. However, a definition can be linked to several files at the same time.

Note: If file text and the file definition are not the same, a new version of the definition is created.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	*LINK, *UNLINK	Required, Positional 1
FILE	Data base file	Single values: *ALL Other values: <i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DTADCT	Data dictionary	<i>Name</i>	Optional
DFN	File definition	<i>Name</i>	Optional
CRTDATE	Creation date	<i>Date</i> , *FIRST	Optional

Top

Option (OPTION)

Specifies the action that is performed on the program-described file, externally-described file, or file definition.

Note: Externally-described files can only be unlinked.

This is a required parameter.

The possible values are:

***LINK**

The program-described file or file definition is linked.

***UNLINK**

The program-described file, externally-described file, or the file definition is unlinked.

Top

Data base file (FILE)

Specifies the name and library of the program-described file or externally-described file to be linked or unlinked.

This is a required parameter.

The possible file values are:

***ALL** All program-described files linked to definitions in the specified dictionary are unlinked. This value is valid only if ***UNLINK** is also specified for the **Option** prompt (OPTION parameter) and a dictionary name is also specified for the **Data dictionary** prompt (DTADCT parameter). This value is not valid for externally-described files.

library-name/file-name

Specify the name of the database file being linked or unlinked.

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 file. 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

Data dictionary (DTADCT)

Specifies the name of the dictionary that contains the file definition being linked or unlinked to the program-described file. A name is required if ***LINK** is specified for the **Option** prompt (OPTION parameter), or if ***ALL** is specified for the **Data base file** prompt (FILE parameter) and ***UNLINK** is specified for the **Option** prompt (OPTION parameter).

Top

File definition (DFN)

Specifies the name of the file definition being linked to the program-described file. This parameter is not applicable if ***UNLINK** is specified for the **Option** prompt (OPTION parameter).

Top

Creation date (CRTDATE)

Specifies the creation date of the file definition being linked to the program-described file. This information is ignored if ***UNLINK** is specified for the **Option** prompt (OPTION parameter).

The possible values are:

***FIRST**

The file definition with the specified definition name and the earliest creation date is used.

creation-date

Specify the creation date of the file definition being linked to the program-described file.

Examples

```
LNKDTADFN  OPTION(*LINK)  FILE(MYLIB/MYFILE)
           DTADCT(MINE)  DFN(MYDEF)
```

This command links definition MYDEF in dictionary MINE, to the program-described database file MYFILE located in library MYLIB.

Error messages

*ESCAPE Messages

CPF2E9B

Definition &1 not found.

CPF2FE0

Error occurred while opening dictionary &1.

CPF2FE1

Error occurred while closing dictionary &1.

CPF2FE2

Dictionary &1 currently in use.

CPF2FE3

System cross reference file is in error.

CPF2FE4

System cross reference file not available.

CPF2F02

Not authorized to use dictionary &1.

CPF2F07

Dictionary &1 in error.

CPF2F08

Dictionary &1 not found.

CPF2F6A

File &2 in &3 not valid for LNKDTADFN.

CPF2F6C

All files were not unlinked.

CPF2F61

File &2 in &3 currently in use.

CPF2F7B

File &2 not linked. Record lengths not equal.

CPF2F7C

Start key position &1 splits field &2.

CPF2F7D

End key position &1 splits field &2.

- CPF2F7F**
File &2 in &3 is already linked.
- CPF2F76**
Only file definitions for physical files can be linked.
- CPF2F77**
File not keyed. Cannot link to keyed file definition.
- CPF2F78**
Definition &1 in error.
- CPF2F79**
Key fields do not match.
- CPF2F80**
File &2 in &3 is not linked.
- CPF9812**
File &1 in library &2 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9845**
Error occurred while opening file &1.
- CPF9846**
Error while processing file &1 in library &2.
- CPF9847**
Error occurred while closing file &1 in library &2.

Top

Load or Unload Image Catalog (LODIMGCLG)

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

Parameters
Examples
Error messages

The Load or Unload Image Catalog (LODIMGCLG) command is used to associate an image catalog and its images to a virtual device. The status of the image catalog will be changed based on the value specified for the **Option (OPTION)** parameter as follows:

*LOAD

This will cause the status of the image catalog to change to Ready. All image catalog entries that are in mounted or loaded status will be loaded in the specified virtual device. The allow save attribute will be set to not allow save for all image catalog files.

*UNLOAD

This will cause the status of the image catalog to change to Not ready. All image catalog entries are removed from the specified virtual device. The allow save attribute will be set to allow save for all image catalog entries.

Only one image catalog can be associated with a virtual device. If the virtual device already has an image catalog associated with it, you can use OPTION(*UNLOAD) to unload the current image catalog.

Restrictions:

- The following authorities are required to load an image catalog:
 1. Execute (*EXECUTE) authority to library QUSRSYS.
 2. Use (*USE) authority to the image catalog.
 3. Use (*USE) authority to the virtual device description.
 4. Execute (*X) authority to each directory in the image catalog path name.
 5. If WRTPTC(*ALL) or WRTPTC(*NONE) is specified, change (*CHANGE) authority to the image catalog is required.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
OPTION	Option	*LOAD , *UNLOAD	Optional, Positional 3
DEV	Virtual device	<i>Name</i>	Optional, Positional 2
WRTPTC	Write protect	*DFT , *ALL, *NONE	Optional

Top

Image catalog (IMGCLG)

Specifies the image catalog to be loaded or unloaded.

This is a required parameter.

name Specify the name of the image catalog.

Top

Option (OPTION)

Specifies whether the image catalog is to be loaded or unloaded.

*LOAD

The image catalog will be loaded into the virtual device.

*UNLOAD

The image catalog will be unloaded from the virtual device.

Top

Virtual device (DEV)

Specify the device that the image catalog is to be loaded into or unloaded from.

This is a required parameter.

name Specify the name of the virtual device.

Top

Write protect (WRTPTC)

Specifies whether to write protect based on each catalog entry's assigned value or to force all catalog entries to be write protected when the image catalog is loaded. The write protect value for each catalog entry will be returned to its previous value when the image catalog is unloaded from the device.

This parameter is valid only when parameter OPTION(*LOAD) is specified. When parameter OPTION(*UNLOAD) is specified the WRTPTC parameter is ignored.

*DFT Each image catalog entry's current write protect value will be used when the image catalog is loaded in the virtual device.

The default write protect values for dependent and reference image catalogs are unique and are defined as follows:

1. Optical dependent image catalogs
 - All entries will have their write protected value set on when the image catalog is loaded in the virtual device.
2. Optical reference image catalogs
 - All entries will have their write protected value set on when the image catalog is loaded in the virtual device.
3. Tape dependent image catalogs
 - All entries will have their write protected value set on when the image catalog is loaded in the virtual device.
4. Tape reference image catalogs

- All entries will use the current write protect value when the image catalog is loaded in the virtual device.

***ALL** All entries will be write protected when the image catalog is loaded in the virtual device.

***NONE**

The write protect value will be set off for all entries when the image catalog is loaded in the virtual device.

The *NONE write protect values for dependent and reference image catalogs are unique and are defined as follows:

1. Optical dependent image catalogs

- All entries will have their write protected value set on when the image catalog is loaded in the virtual device.

2. Optical reference image catalogs

- All entries will have their write protected value set on when the image catalog is loaded in the virtual device.

3. Tape dependent image catalogs

- All entries will have their write protected value set on when the image catalog is loaded in the virtual device.

4. Tape reference image catalogs

- All entries will have their write protect value set off when the image catalog is loaded in the virtual device.

Top

Examples

Example 1: Loading an Image Catalog

```
LODIMGCLG  IMGCLG(MYCLG)  DEV(OPTVRT01)  OPTION(*LOAD)
```

This command loads image catalog **MYCLG** into device **OPTVRT01**. Write protection is set based on the current value for each catalog entry. This example assumes that **MYCLG** is not a dependent or reference catalog.

Example 2: Loading an Image Catalog as Read-only

```
LODIMGCLG  IMGCLG(MYCLG)  DEV(OPTVRT01)  OPTION(*LOAD)
           WRTPTC(*ALL)
```

This command loads image catalog **MYCLG** into device **OPTVRT01**. Write protection is set on for all catalog entries.

Example 3: Unloading an Image Catalog

```
LODIMGCLG  IMGCLG(MYCLG)  OPTION(*UNLOAD)
```

This command unloads image catalog **MYCLG** from the associated virtual device.

Example 4: Loading a Tape Image Catalog

```
LODIMGCLG  IMGCLG(MYCLG)  DEV(TAPVRT01)  OPTION(*LOAD)
```

This command loads image catalog **MYCLG** into device **TAPVRT01**.

[Top](#)

Error messages

*ESCAPE Messages

CPFBC10

Image catalog &1 not loaded to device &2.

CPFBC11

Image catalog &1 not unloaded from device &2.

CPFBC45

Image catalog &1 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9820

Not authorized to use library &1.

[Top](#)

Load/Unload/Mount IMGCLG Entry (LODIMGCLGE)

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

Parameters
Examples
Error messages

The Load/Unload/Mount Image Catalog Entry (LODIMGCLGE) command is used to change the status of an image catalog entry in an image catalog. The status of the image catalog entry will be changed based on the value specified for the **Option (OPTION)** parameter as follows:

*LOAD

This will cause the status of the image catalog entry to change to Loaded.

*UNLOAD

This will cause the status of the image catalog entry to change to Unloaded.

*MOUNT

This will cause the status of the image catalog entry to change to Mounted.

Only one image catalog entry can be in a mounted status. If OPTION(*MOUNT) is specified, an existing entry in mounted status will be changed to a loaded status.

Restrictions:

- The following authorities are required to load an image catalog entry:
 1. Execute (*EXECUTE) authority to library QUSRSYS.
 2. Use (*USE) authority to the image catalog being loaded or unloaded.
 3. Use (*USE) authority to the virtual device description.
 4. Execute (*X) authority to each directory in the image catalog path name.
- If a device of type 632B and model 003 is specified on the Virtual device (DEV) parameter the command will be restricted to only *MOUNT and *LOAD requests.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	Name, *DEV	Required, Positional 1
IMGCLGIDX	Image catalog index	1-256, <u>*FIRST</u> , *LAST, *NEXT, *VOL	Optional, Positional 2
OPTION	Option	<u>*MOUNT</u> , *LOAD, *UNLOAD	Optional, Positional 3
VOL	Image catalog volume	Character value	Optional
DEV	Virtual device	Name	Optional

Top

Image catalog (IMGCLG)

Specifies the image catalog to be used.

This is a required parameter.

***DEV** Use the virtual device specified on the Virtual device (DEV) parameter to locate the image catalog image.

name Specify the name of the image catalog.

Top

Image catalog index (IMGCLGIDX)

Specifies the index number of the image catalog entry whose status is to be changed.

***FIRST**
The first image catalog entry in the image catalog.

***LAST**
The last image catalog entry in the image catalog.

***NEXT**
The next image catalog entry in loaded status will be mounted. This value is only allowed when OPTION(*MOUNT) is specified.

***VOL** Specifies to load/unload/mount the entry by specifying a volume name.

1-256 Specify the image catalog index number to be used.

Top

Option (OPTION)

Specifies the new status of the image catalog entry.

***MOUNT**
The image catalog entry will be mounted into the image catalog.

***LOAD**
The image catalog entry will be loaded into the image catalog.

***UNLOAD**
The image catalog entry will be unloaded from the image catalog.

Top

Image catalog volume (VOL)

Specifies the volume name of the entry to be unloaded, loaded, or mounted.

name Specify the volume name of the entry to be unloaded, loaded, or mounted. For optical image catalogs, the first volume that matches the volume specified will have its status changed.

Top

Virtual device (DEV)

Specify the name of the virtual device.

name Specify the name of the virtual device that has an image catalog image or a device type 632B model 003 that has image catalog image loaded from a network file server to be loaded or mounted.

Top

Examples

Example 1: Mounting an Image Catalog Entry

```
LODIMGCLGE  IMGCLG(MYCLG)  IMGCLGIDX(*FIRST)  OPTION(*MOUNT)
```

This command mounts the first image catalog entry in image catalog **MYCLG**. If there is an image mounted, it will be changed to a loaded status.

Example 2: Loading an Image Catalog Entry

```
LODIMGCLGE  IMGCLG(MYCLG)  IMGCLGIDX(5)  OPTION(*LOAD)
```

This command loads the image catalog entry associated with index number **5** in image catalog **MYCLG**.

Example 3: Unloading an Image Catalog Entry

```
LODIMGCLGE  IMGCLG(MYCLG)  IMGCLGIDX(*LAST)  OPTION(*UNLOAD)
```

This command unloads the last image catalog entry in image catalog **MYCLG**.

Example 4: Mounting a Tape Image Catalog Entry by Volume

```
LODIMGCLGE  IMGCLG(TAPECLG)  IMGCLGIDX(*VOL)
              VOL(TAP002)  OPTION(*MOUNT)
```

This command mounts the tape image catalog entry associated with volume **TAP002** in image catalog **TAPECLG**.

Example 5: Mounting an Optical Image Catalog Entry by Volume

```
LODIMGCLGE  IMGCLG(*DEV)  DEV(OPTVRT01)  IMGCLGIDX(1)  OPTION(*MOUNT)
```

This command mounts the optical volume at index 1 of device **OPTVRT01**.

Top

Error messages

*ESCAPE Messages

CPFBC46

Catalog entry at index &1 not loaded.

CPFBC47

Catalog entry at index &1 not unloaded.

CPFBC48

Catalog entry at index &1 not mounted.

CPFBC0D

Catalog entry at index &1 not loaded.

CPFBC0E

Catalog entry at index &1 not unloaded.

CPFBC0F

Catalog entry at index &1 not mounted.

CPFBC45

Image catalog &1 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9820

Not authorized to use library &1.

Top

Load Optical Firmware (LODOPTFMW)

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

Parameters
Examples
Error messages

The Load Optical Firmware (LODOPTFMW) command loads optical device firmware into a staging area that can be accessed by the Optical Hardware Service Manager. The Optical Hardware Service Manager must be used to load the firmware onto the device.

Restrictions:

1. This command is shipped with *EXCLUDE public authority.
2. You must have an active optical device although it does not need to be the destination device for the firmware. The system will find the varied on device.

Top

Parameters

Keyword	Description	Choices	Notes
FROMSTMF	From stream file	<i>Path name</i>	Required, Positional 1
TYPE	Device type	<i>Character value</i>	Required, Positional 2
MODEL	Device model	<i>Character value</i>	Required, Positional 3

Top

From stream file (FROMSTMF)

Specifies the stream file that contains the firmware to be read into cache.

Top

Device type (TYPE)

Specifies the optical media library type that the firmware will eventually be loaded on.

Top

Device model (MODEL)

Specifies the model the firmware is for.

Top

Examples

Example 1: Specify a Media Library

```
LODOPTFMW FROMSTMF('/MyDir/MyFile') TYPE(399F)
           MODEL(200)
```

This command loads the optical device firmware from MyFile into system storage for a 399F Optical Media Library Model 200.

Example 2: Specify an Optical Drive

```
LODOPTFMW FROMSTMF('/MyDir/MyFile') TYPE(399F)
           MODEL(002)
```

This command loads the optical device firmware from MyFile into system storage for a 14x optical drive.

Top

Error messages

*ESCAPE Messages

OPT1114

Error accessing stream file.

OPT1121

Length beyond end of buffer.

OPT1681

An active optical device was not found.

OPT1815

Internal program error occurred.

OPT1821

Error occurred on optical device &1.

Top

Load Program Temporary Fix (LODPTF)

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

Parameters
 Examples
 Error messages

The Load Program Temporary Fix (LODPTF) command loads program temporary fixes (PTFs) for a specified product from a tape, optical device, or save file into the product PTF library. Each PTF contains one or more objects, including programs, that can be applied to a product by the Apply Program Temporary Fix (APYPTF) command.

Only the PTFs for a single product can be loaded at one time. Specific PTFs can be selected or omitted when loading PTFs for the specified product. PTFs that are currently applied are not loaded.

Restrictions:

- To use this command, you must be signed on as QSRV, or have all object (*ALLOBJ) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	Character value	Required, Positional 1
DEV	Device	Name, <u>*SERVICE</u> , *SAVF	Optional
SELECT	PTF numbers to select	Single values: <u>*ALL</u> Other values (up to 50 repetitions): Character value	Optional
OMIT	PTF numbers to omit	Values (up to 50 repetitions): Character value	Optional
SPRPTF	Superseded PTFs	<u>*APYPERM</u> , *NOAPY	Optional
RLS	Release	Character value, <u>*ONLY</u>	Optional
SEQNBR	Sequence number	1-16777215, <u>*SEARCH</u>	Optional
ENDOPT	End of media option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
PATHID	Path identifier	1-9999, <u>*FIRST</u> , *SELECT	Optional
SAVF	Save file	Qualified object name	Optional
	Qualifier 1: Save file	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
COVER	Copy PTF cover letter	*SRVATT, <u>*YES</u> , *NO, *ONLY	Optional
CPYSAVF	Copy PTF save file	<u>*SRVATT</u> , *YES, *NO	Optional

Top

Product (LICPGM)

Specifies the 7-character identifier of the product for which the PTFs are loaded.

This is a required parameter.

Top

Device (DEV)

Specifies the device from which the PTFs are loaded. The device name must be known on the system by a device description.

*SERVICE

The PTFs that were sent from the service support system are loaded.

***SAVF** The PTFs are loaded from a save file. If *SAVF is specified, a value for the **Save file (SAVF)** parameter is required.

name Specify the name of the tape or optical device that is used to load the PTFs.

Top

PTF numbers to select (SELECT)

Specifies which of the PTFs for the specified product are loaded. The **PTF numbers to omit (OMIT)** parameter cannot be specified if single PTF numbers are specified for this parameter.

Note: Permanently removed PTFs are ignored when SELECT(*ALL) and DEV(*SERVICE) are specified. To load removed PTFs, specify the PTF number on this parameter.

Single values

***ALL** All the PTFs for the specified product are loaded.

Other values (up to 50 repetitions)

character-value

Specify the PTF identification numbers of the single PTFs that are loaded.

Top

PTF numbers to omit (OMIT)

Specifies that all PTFs except for those specified in this parameter are loaded. A maximum of 50 PTF numbers can be specified.

Note: The OMIT parameter cannot be specified if single PTF numbers are specified on the **PTF numbers to select (SELECT)** parameter.

character-value

Specify the PTF identification numbers of the single PTFs that are omitted (not loaded).

Top

Superseded PTFs (SPRPTF)

Specifies the operation that is performed for temporarily applied PTFs that are being superseded by PTFs encountered by this load operation.

*APYPERM

For the specified product, any PTFs that are temporarily applied, and are superseded by PTFs contained on the PTF media, are automatically permanently applied before loading the superseding PTFs. If the superseded PTFs have any prerequisite PTFs, they are also permanently applied by this operation.

***NOAPY**

The load operation stops when temporarily applied PTFs are being superseded by PTFs contained on the PTF medium. The temporarily applied PTFs that are being superseded must be permanently applied (APYPTF command) or removed (RMVPTF command) before the LODPTF command can be processed again.

Top

Release (RLS)

Specifies the release level of the PTFs being loaded.

***ONLY**

This value is valid only if only one release of the product's base option is installed on the system. PTFs for all installed options of the product will be loaded regardless of the release-level of the option.

character-value

Specify the release level in VxRyMz format where Vx is the version number, Ry is the release number, and Mz is the modification level. The variables x and y can be a number from 0 through 9, and the variable z can be a number from 0 through 9 or a letter from A through Z.

If the release-level specified is the release-level of the base option of the product, PTFs for all installed options of the product are loaded regardless of the release-level of the option.

If the release-level specified is not the release-level of the base option of the product, only PTFs for the options installed at that release-level are loaded.

Top

Sequence number (SEQNBR)

Specifies the sequence number on the tape volume where the load operation begins to load the PTF data. This parameter is valid only if a tape device name is specified on the **Device (DEV)** parameter.

***SEARCH**

The tape volume is searched for the first PTF file for the specified product. The first PTF file found is loaded.

1-16777215

Specify the sequence number of the PTF file being loaded. This sequence number must exist on the tape.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically performed on the tape or optical volume after the PTF operation ends.

Note: This parameter is valid only if a tape or optical device name is specified on the DEV parameter. For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

***REWIND**

The tape is automatically rewound, but not unloaded, after the operation has ended.

***LEAVE**

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

***UNLOAD**

The tape is automatically rewound and unloaded after the operation ends. Some optical devices will eject the volume after the operation ends.

Top

Path identifier (PATHID)

Specifies the number that identifies a file on the optical media that contains the PTFs to be loaded. The PTF files for each product and release that exist on the optical media have a path identifier number to allow the files to be processed in a specific order. Only the PTFs from the specified path identifier are loaded on your system.

Note: This parameter is valid only if an optical device name is specified on the DEV parameter.

***FIRST**

The optical media is searched for the first PTF file for the specified product and release, according to the search dependency specified on the SELECT parameter.

- When a specific PTF identifier is specified on the SELECT parameter, the first occurrence of the specified PTF is loaded.
- When *ALL is specified on the SELECT parameter, the existing PTF file with the lowest path identifier is loaded.

***SELECT**

A list of the PTF files that exist on the optical media that match the product and release is shown. You can select the specific file from which PTFs are loaded. This value cannot be selected in a batch environment.

1-9999 Specify the path identifier number of the existing PTF file from which to load the PTF data.

Top

Save file (SAVF)

Specifies the save file from which the PTFs are loaded. This parameter is valid only if *SAVF is specified on the Device (DEV) parameter.

Qualifier 1: Save file

name Specify the name of the save file from which the PTFs are loaded.

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 save file. 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 save file is located.

Top

Copy PTF cover letter (COVER)

Specifies whether to copy the cover letter for the PTF into a physical file. This parameter is valid only when a tape or optical device name is specified on the **Device (DEV)** parameter.

***YES** After the PTF is loaded, the cover letter is copied into a physical file.

***NO** The cover letter is not copied into a physical file.

***ONLY**

The cover letter is copied into a physical file but PTF is not loaded. If the SEQNBR parameter is specified, it must contain the sequence number of the file that contains the PTF.

***SRVATT**

Use the Copy PTFs (CPYPTF) service attribute to determine if the cover letter for the PTF should be copied into a physical file. The Display Service Attributes (DSPSRVA) command displays information about how the system is set up. This includes whether PTF save files and cover letters will be copied into *SERVICE when PTFs are loaded. The Change Service Attributes (CHGSRVA) command can be used to change the CPYPTF service attribute.

Top

Copy PTF save file (CPYSAVF)

Specifies whether to copy PTF save files into *SERVICE when PTFs are loaded. PTF save files must be in *SERVICE when distributing PTFs to other systems or when using the Save System Information (SAVSYSINF) command. This parameter is valid only when a tape or optical device name is specified on the **Device (DEV)** parameter.

***SRVATT**

Use the Copy PTFs (CPYPTF) service attribute to determine if PTF save files will be copied into *SERVICE when PTFs are loaded. The Display Service Attributes (DSPSRVA) command displays information about how the system is set up. This includes whether PTF save files and cover letters will be copied into *SERVICE when PTFs are loaded. The Change Service Attributes (CHGSRVA) command can be used to change the CPYPTF service attribute.

***YES** PTF save files that do not already exist are copied into *SERVICE when PTFs are loaded.

***NO** PTF save files are not copied into *SERVICE when PTFs are loaded.

Top

Examples

Example 1: Omitting PTFs

```
LODPTF LICPGM(5761SS1) OMIT(SI00003 SI00008 SI00014)
```

This command loads all of the PTFs from the service support system (*SERVICE) for the product 5761SS1 except SI00003, SI00008, and SI00014. The Apply Program Temporary Fix (APYPTF) command can then be used to apply these PTFs to the 5761SS1 product.

Example 2: Selecting PTFs

```
LODPTF LICPGM(5761SS1) DEV(OPT01) SELECT(SI00009 SI00010)
```

This command loads the PTFs named SI00009 and SI00010 from the optical device named OPT01. The Apply Program Temporary Fix (APYPTF) command can then be used to apply these PTFs to the 5761SS1 product.

Top

Error messages

*ESCAPE Messages

CPF35AA

Licensed internal code PTF &2 already applied.

CPF35AB

Licensed Internal Code fix &2 not applied.

CPF35AE

Duplicate PTF &1 found.

CPF35A0

Cannot allocate library &1.

CPF35A1

Wrong copy of Licensed Internal Code in use.

CPF35A2

Required hardware changes not installed for PTF &2.

CPF35A3

Licensed Internal Code fix &2 not temporarily applied.

CPF35A5

Licensed Internal Code fix &2 not permanently applied.

CPF35A6

Language option &1 not installed for licensed program.

CPF35A8

No PTFs to be loaded.

CPF35A9

Error occurred while processing Licensed Internal Code fix.

CPF35CC

Library required for PTF operation already exists.

CPF35CF

PTF &1-&2 not applied.

CPF35C1

LODPTF ended. No more storage available.

CPF35C9

PTF &1-&2 &3 not valid.

CPF35EB

Multiple releases of product &1 installed.

CPF35E3

Interface error detected.

CPF35FA

PTF &1-&2 not applied.

CPF35F4
Error occurred during cover letter processing.

CPF35F6
MPTFI for library &1 deleted and created.

CPF354A
Cannot specify *SELECT for the path identifier.

CPF354C
Cannot process PTF files on optical volume.

CPF354D
Device &1 not allowed.

CPF354E
No file selected.

CPF354F
Required PTF file cannot be processed.

CPF355B
Multiple releases for product &1 found on media.

CPF355C
No PTFs found in path identifier &1.

CPF3558
Cannot allocate &1 in &3 type *&2.

CPF3564
PTF &1-&2 damaged.

CPF358A
Release not valid.

CPF3586
List of PTFs not correct.

CPF3587
PTFs not loaded.

CPF3590
PTF &1-&2 &3 not loaded.

CPF3598
PTF function already in process.

CPF3606
Product &1 &2 not installed.

CPF361D
Apply order of PTFs cannot be determined.

CPF3612
Library &1 not found.

CPF3616
No PTFs loaded.

CPF3619
PTFs for release &1 found on device.

CPF3657
PTFs not loaded because error occurred.

CPF3693

Service function ended because error occurred.

CPF3924

PTF not loaded.

CPF3931

Required programs not found. PTF incomplete.

CPF3945

Records of PTF activity for licensed program are deleted.

CPF3992

No PTFs exist on save/restore media for licensed program &1 &2.

CPF6602

PTF &1-&2 &3 not found.

CPF8191

Product definition &4 in &9 damaged.

CPF8193

Product load object &4 in &9 damaged.

Top

Load Q/A Database (LODQSTDB)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)
- Using QCMDEXEC, QCAEXEC, or QCAPCMD API (*EXEC)

Threadsafe: No

Parameters
Examples
Error messages

The Load Question and Answer Database (LODQSTDB) command allows you to load a Question and Answer (Q & A) database from an alternative medium (such as tape) to the system. More information is available in the Basic system operations topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. A user must have authority to the command and be a Q & A coordinator for any Q & A database referred to by the command.
3. This command can only be used interactively.

Top

Parameters

Keyword	Description	Choices	Notes
QSTDB	Q/A database	Name, <u>*SELECT</u>	Optional, Positional 1
LIB	Lib containing Q/A database	Name, <u>QUSRSYS</u>	Optional, Positional 2

Top

Q/A database (QSTDB)

Specifies the Q & A database to be loaded.

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 to be used.

- If the Q & A database already exists on your system, the supplied subset of the Q & A database will be replaced.
- If the Q & A database does not exist on your system, it will be created in the specified library.

Top

Lib containing Q/A database (LIB)

Specifies the name of the library that contains the Q & A database.

The possible library values are:

QUSRSYS

The library default for this command is QUSRSYS.

library-name

Specify the library where the Q & A database is to be loaded. The library must exist on the system.

Top

Examples

LODQSTDB

This command shows the Load Database to System display.

Top

Error messages

None

Top

Load and Run (LODRUN)

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

Parameters
Examples
Error messages

The Load and Run Media Program (LODRUN) command restores a user-written program object from tape, diskette, or optical device into the library QTEMP. The system passes the device name to the restored program and transfers control to the restored program.

When the LODRUN command is run:

1. The media is searched for the user-written program, which must be named QINSTAPP and saved from library QTEMP.
Note: The program QINSTAPP must be owned by a user profile that resides on the target system. If QINSTAPP is restored to a system that does not have the owning user profile, control is not transferred and the program is not run.
2. If a QINSTAPP program already exists in the QTEMP library on the user's system, it is deleted.
3. The QINSTAPP program is restored to the QTEMP library using the RSTOBJ command. The following values are specified on the RSTOBJ command:

- OBJ(QINSTAPP)
- OBJTYPE(*PGM)
- SAVLIB(QTEMP)
- ENDOPT(*LEAVE)
- MBROPT(*ALL)
- ALWOBJDIF(*NONE)
- RSTLIB(QTEMP)
- VOL(*MOUNTED)

If the device is an optical device, the ENDOPT and the VOL parameters are not specified.

If the device is optical, then the value specified for the DIR for this command is used for the OPTFILE parameter for the Restore Object (RSTOBJ) command.

The SEQNBR parameter is specified according to the SEQNBR parameter on the LODRUN command.

The device used for the restore operation is determined by the LODRUN command. If *TAP, *DKT, or *OPT are specified on the DEV parameter, the LODRUN command examines the QDEVNAMING system value to determine if the system uses System i or System/36 naming conventions for devices:

- If QDEVNAMING is *NORMAL (System i convention)
 - The device TAP01 is used for DEV(*TAP).
 - The device DKT01 is used for DEV(*DKT).
 - The device OPT01 is used for DEV(*OPT).
- If QDEVNAMING is *S36 (System/36 convention)
 - The device TC is used for DEV(*TAP) if a tape cartridge is found. Otherwise, device T1 is used.
 - The device I1 is used for DEV(*DKT).
 - The device OPT01 is used for DEV(*OPT). System/36 naming conventions do not apply to optical devices.

Any other value specified on the DEV parameter is used as is.

4. Control of the system is passed to the QINSTAPP program. The QINSTAPP program can be used, for example, to restore other applications to the user's system and run those applications.

5. When the user signs off, the QINSTAPP program is removed from the system.
6. The settings of three system values work together to determine if the QINSTAPP program is allowed to be restored or if it is converted during the restore. The three system values are:
 - QVfyOBRST Verify object restore
 - QFRCCVNRST Force conversion on restore
 - QALWOBRST Allow object restore option

The LODRUN command does not transfer control to the QINSTAPP program if these values do not allow the restore and conversion of the QINSTAPP program.

The user supplying the QINSTAPP program is responsible for writing and supporting it. The QINSTAPP program is not supplied by IBM. The program can be designed to accomplish many different tasks. For example, the program could:

- Restore and run other programs or applications
- Restore a library
- Delete another program or application
- Create specific environments
- Apply fixes to existing applications

The QINSTAPP program is run only once each time the LODRUN command is entered. The LODRUN command passes only one parameter (DEV), which specifies the device from which the QINSTAPP program is restored. The QINSTAPP program should not attempt to use the LODRUN command again. This will have unpredictable results.

In addition to writing the QINSTAPP program, the user supplying the program is responsible for providing the user with the media containing the program. To distribute the program on a tape, diskette, or optical device, do the following:

1. Prepare the tape or diskette. For tape, use the Initialize Tape (INZTAP) command. For diskette, use the Initialize Diskette (INZDKT) command with FMT(*SAVRST) specified.
2. Use the Create Duplicate Object (CRTDUPOBJ) command to create the QINSTAPP program into the QTEMP library.
3. Use the Save Object (SAVOBJ) command to save the QINSTAPP program from QTEMP to the desired tape device or diskette unit. The program must be the only object in the media file that contains it. Specify the following:
 - LIB(QTEMP)
 - LABEL(*LIB)
 - CLEAR(*ALL)

Specifying LABEL(*LIB) ensures that the label will be QTEMP. If the QINSTAPP program is being saved to a tape device, and if additional applications, programs, or libraries will be saved to tape, ENDOPT(*LEAVE) also must be specified. The correct value for the TGTRLS parameter must also be entered if the target release is not the default, *CURRENT.

4. Use the Save Object (SAVOBJ), Save Library (SAVLIB), or Save License Program (SAVLICPGM) command to save any other necessary applications, programs, or libraries to the tape or diskette. This step is optional and is used to save applications that the QINSTAPP program restores to the user's system when the LODRUN command is run.

If the QINSTAPP program is saved to tape, the tape is not rewound after the QINSTAPP program is restored; the application or series of applications that the QINSTAPP program restores to the user's system should be next on the tape.

Restriction: The QINSTAPP program that is loaded from the media and called may require specific authority in order to run correctly. The user supplying the QINSTAPP program should inform you if any specific authorities are required.

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Name, <u>*TAP</u> , *DKT, *OPT	Optional, Positional 1
SEQNBR	Sequence number	Decimal number, <u>*FIRST</u> , *SEARCH	Optional
VOL	Volume identifier	Character value, <u>*MOUNTED</u> , *SAVVOL	Optional
DIR	Directory	Character value, <u>'/'</u>	Optional

Device (DEV)

Specifies the I/O device from which the program is loaded.

This is a required parameter.

The possible values are:

*TAP The program is loaded from the default tape device connected to the system.

*DKT The program is loaded from the default diskette device connected to the system.

*OPT The program is loaded from the default optical device connected to the system.

tape-device

Specify the name of the tape device from which the program is loaded onto the system.

diskette-device

Specify the name of the diskette device from which the program is loaded onto the system.

Optical-device

Specify the name of the optical device from which the program is loaded onto the system.

Sequence number (SEQNBR)

Specifies, only when tape is used, the sequence number used for the restore operation.

The possible values are:

*FIRST

The volume on a tape device is searched starting from the first data file for a data file with an identifier that is a match for the QTEMP label. When the first match is found, the object is restored.

*SEARCH

The volume on a tape device is searched starting from the first data file beyond the current tape position for a data file with an identifier that is a match for the QTEMP label. When a match is found, the object is restored.

sequence-number

Specify the sequence number of the file. Valid values range from 1 through 16777215.

Volume identifier (VOL)

Specifies, only when tape is used, the volume identifier for the tape devices.

The possible values are:

*MOUNTED

The volume currently mounted on the tape device is used.

volume-identifier

Specify the volume identifier mounted on the tape device.

Top

Directory (DIR)

Specifies, only when an optical device is used, the directory used for the restore operation. If the file named QTEMP;1 is found in the specified directory, the object is restored.

The possible values are:

'/' The root directory (/) is used.

directory-name

Specify the directory to search for a file named QTEMP.

Top

Examples

Example 1: Restoring a Program from Tape

```
LODRUN  DEV(TAP01)
```

This command restores the program object from the tape on device TAP01 to the library QTEMP. Control is then transferred to the restored program.

Example 2: Restoring the Program QINSTAPP from Tape

```
LODRUN  DEV(TAP01)  SEQNBR(5)
```

This command restores the program object QINSTAPP from the tape at sequence number 5 on device TAP01 to the library QTEMP. Control is then transferred to the restored program. If the sequence number is not found, an escape message is sent. If the file label at that sequence number is not QTEMP, an escape message is sent.

Example 3: Restoring the Program QINSTAPP from CD-ROM

```
LODRUN  DEV(*OPT)  DIR('/APP1/INST')
```

This command restores the program object QINSTAPP from the CD-ROM on device OPT01 to the library QTEMP. The filename for the QTEMP library on the CD-ROM is /APP1/INST/QTEMP. Control is then transferred to the restored program. If the file is not found, an escape message is sent.

[Top](#)

Error messages

None

[Top](#)

Create Directory (MD)

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

Parameters
Examples
Error messages

The Create Directory (MD) command adds a new directory to the system.

A directory is an object that contains the names of other objects. Libraries and folders are types of directories. When a directory is created, a link is added to the directory prefix. The directory must have been created before any objects can be placed into it.

This command is an alias for the Create Directory (CRTDIR) command and can also be issued using the following alternative command names:

- CRTDIR
- MKDIR

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- The following restriction applies when the directory to be created is a library in the QSYS.LIB or independent ASP QSYS.LIB file system, or a directory within the "root" (/), QOpenSys, or user-defined file systems:
 - The audit (*AUDIT) special authority is required when specifying a value other than *SYSVAL on the **Auditing value for objects (CRTOBJAUD)** parameter.
- The following restriction applies when the directory to be created is a folder in an existing folder in QDLS:
 - The change (*CHANGE) authority is required for the existing folder.
- The user must have execute (*X) authority to each directory in the path.
- When creating a directory in the "root" (/), QOpenSys or user_defined file system, the user must have write and execute (*WX) authority to the directory that contains the new directory.
- When creating a directory, the owner ID (UID) is the user creating the directory.

If the directory is to be created in the "root" (/), QOpenSys, and user-defined file systems, the following applies. If the S_ISGID bit of the parent directory is off, the group ID (GID) is set to the effective GID of the thread creating the directory. If the S_ISGID bit of the parent directory is on, the group ID (GID) of the new directory is set to the GID of the parent directory.

If the directory is to be created in the QSYS.LIB or independent ASP QSYS.LIB file system, the GID is obtained from the primary user profile. For all other file systems, the GID is obtained from the parent directory.

- The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authorities to specify a value for the **Scanning option for objects (CRTOBJSCAN)** parameter other than *PARENT.

Top

Parameters

Keyword	Description	Choices	Notes
DIR	Directory	<i>Path name</i>	Required, Positional 1
DTAAUT	Public authority for data	<i>Name</i> , *INDIR, *RWX, *RW, *RX, *WX, *R, *W, *X, *EXCLUDE, *NONE	Optional
OBJAUT	Public authority for object	Single values: *INDIR, *NONE, *ALL Other values (up to 4 repetitions): *OBJEXIST, *OBJMGT, *OBJALTER, *OBJREF	Optional
CRTOBJAUD	Auditing value for objects	*SYSVAL, *NONE, *USRPRE, *CHANGE, *ALL	Optional
CRTOBJSCAN	Scanning option for objects	*PARENT, *YES, *NO, *CHGONLY	Optional
RSTDRNMUNL	Restricted rename and unlink	*NO, *YES	Optional

Top

Directory (DIR)

Specifies the path name of the directory to be created.

Note: Do not use a name that begins with the character Q. The system assumes that libraries or directories with those names are system libraries or directories.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

Public authority for data (DTAAUT)

Specifies the public data authority given to the user for the directory, or specifies that all authorities are inherited from the directory it is to be created in.

*INDIR

The authority for the directory to be created is determined by the directory it is to be created in. The directory immediately preceding the new directory determines the authority. A directory created in the "root" (/), QOpenSys, or user-defined file system is assigned the same public, private and primary group authority, authorization list, and primary group as the directory it is to be created in. A directory created in QDLS for a folder defaults to *EXCLUDE for a first level folder. If created in the second level or greater, the authority of the previous level is used. The QOpenSys and "root" (/) file systems use the parent directory's Data Authority value. If the value *INDIR is specified for either the **Public authority for object (OBJAUT)** parameter or the DTAAUT parameter, then *INDIR must be specified for both parameters.

***RWX** The user can change the object and perform basic functions on the object except those limited to the owner or controlled by object existence (*OBJEXIST), object management (*OBJMGT), object alter (*OBJALTER) and object reference (*OBJREF) authorities. Read, write, and execute (*RWX) authority provides object operational (*OBJOPR) and all data authorities.

- *RW** The user can view and change the contents of an object. Read and write (*RW) authority provides *OBJOPR and data read (*READ), add (*ADD), update (*UPD) and delete (*DLT) authorities.
- *RX** The user can perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. Read and execute (*RX) authority provides *OBJOPR and data *READ and execute (*EXECUTE) authorities.
- *WX** The user can change the contents of an object and run a program or search a library or directory. Write and execute (*WX) authority provides *OBJOPR and data *ADD, *UPD, *DLT, and *EXECUTE authorities.
- *R** The user can view the contents of an object. Read (*R) authority provides *OBJOPR and data *READ authorities.
- *W** The user can change the contents of an object. Write (*W) authority provides *OBJOPR and data *ADD, *UPD, and *DLT authorities.
- *X** The user can run a program or search a library or directory. Execute (*X) authority provides *OBJOPR and data *EXECUTE authorities.
- *EXCLUDE**
The user cannot access the object. The OBJAUT value must be *NONE, if this special value is used.
- *NONE**
The user is given no data authorities to the objects. This value cannot be used with the OBJAUT value of *NONE.
- name* Specify the name of the authorization list used. The format of the authorization list name remains the current ten-character format. The OBJAUT value must be *NONE, if this special value is used.

Top

Public authority for object (OBJAUT)

Specifies the public object authority given to users for the directory, or specifies that all authorities are inherited from the directory it is to be created in.

*INDIR

The object authority is based on the authority for the directory where this directory is to be created. A directory created in the "root" (/), QOpenSys, or user-defined file system is assigned the same public, private and primary group authority, authorization list, and primary group as the directory it is to be created in. If the value *INDIR is specified for either the OBJAUT parameter or the **Public authority for data (DTAAUT)** parameter, then *INDIR must be specified for both parameters.

*NONE

None of the other object authorities (*OBJEXIST, *OBJMGT, *OBJALTER or *OBJREF) are given to the users. If *EXCLUDE or an authorization list is specified for the DTAAUT parameter, *NONE must be specified. This value cannot be used with the DTAAUT value of *NONE.

***ALL** All of the other object authorities (*OBJEXIST, *OBJMGT, *OBJALTER or *OBJREF) are given to the users.

The user can specify up to four of the following values:

*OBJEXIST

The user is given object existence (*OBJEXIST) authority to the object. The user can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object.

***OBJMGT**

The user is given object management (*OBJMGT) authority to the object. With this authority the user can specify security for the object, move or rename the object and add members to database files.

***OBJALTER**

The user is given object alter (*OBJALTER) authority to the object. The user is able to alter the attributes of the objects. On a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. With this authority on an SQL package, the user can change the attributes of the SQL package. Currently, this authority is used only for database files and SQL packages.

***OBJREF**

The user is given object reference (*OBJREF) authority to objects. Used only for database files, the user can reference an object from another object such that operations on that object may be restricted by the other object. On a physical file, the user can add a referential constraint in which the physical file is the parent.

Top

Auditing value for objects (CRTOBJAUD)

Specifies the auditing value of objects created in this directory.

Values for this parameter other than *SYSVAL may not be supported by some file systems.

***SYSVAL**

The object auditing value for the objects in the directory is determined by the Create object auditing (QCRTOBJAUD) system value.

***NONE**

Using or changing this object does not cause an audit entry to be sent to the security journal.

***USRPRF**

The user profile of the user accessing this object is used to determine if an audit record is sent for this access. The OBJAUD parameter of the Change User Auditing (CHGUSRAUD) command is used to change the auditing for a specific user.

***CHANGE**

All change accesses to this object by all users are logged.

***ALL** All change or read accesses to this object by all users are logged.

Top

Scanning option for objects (CRTOBJSCAN)

Specifies whether the objects created in a directory will be scanned when exit programs are registered with any of the integrated file system scan-related exit points.

The integrated file system scan-related exit points are:

- QIBM_QP0L_SCAN_OPEN - Integrated File System Scan on Open Exit Program
- QIBM_QP0L_SCAN_CLOSE - Integrated File System Scan on Close Exit Program

For details on these exit points, see the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This attribute can only be specified for directories created in the "root" (/), QOpenSys and user-defined file systems. For all other file systems, *PARENT should be specified and it will be ignored. Even though this attribute can be set for *TYPE1 and *TYPE2 directories, only objects which are in *TYPE2 directories will actually be scanned, no matter what value is set for this attribute.

***PARENT**

The create object scanning attribute value for this directory is copied from the create object scanning attribute value of the parent directory.

***YES** After an object is created in the directory, the object will be scanned according to the rules described in the scan-related exit programs if the object has been modified or if the scanning software has been updated since the last time the object was scanned.

***NO** After an object is created in the directory, the object will not be scanned by the scan-related exit programs.

Note: If the Scan file systems control (QSCANFCTL) value *NOPOSTRST is not specified when an object with this attribute is restored, the object will be scanned at least once after the restore.

***CHGONLY**

After an object is created in the directory, the object will be scanned according to the rules described in the scan-related exit programs only if the object has been modified since the last time the object was scanned. It will not be scanned if the scanning software has been updated. This attribute only takes effect if the Scan file systems control (QSCANFCTL) system value has *USEOCOATR specified. Otherwise, it will be treated as if the attribute is *YES.

Note: If the Scan file systems control (QSCANFCTL) value *NOPOSTRST is not specified when an object with this attribute is restored, the object will be scanned at least once after the restore.

Top

Restricted rename and unlink (RSTDRNMUNL)

Specifies whether special restrictions apply for rename and unlink operations performed on objects within a directory. This attribute is equivalent to the S_ISVTX mode bit and can only be set for a directory in the Network File System (NFS), QFileSvr.400, "root" (/), QOpenSys, or user-defined file systems. Both the NFS and QFileSvr.400 file systems support this attribute by passing it to the server and surfacing it to the caller.

***NO** No additional restrictions for renaming or unlinking objects from this directory.

***YES** Objects within this directory may be renamed or unlinked only if one or more of the following are true for the user performing the operation:

1. The user is the owner of the object.
2. The user is the owner of the directory.
3. The user has all object (*ALLOBJ) special authority.

Top

Examples

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

Example 1: Creating a Directory

```
CRTDIR DIR('MYDIR')
```

This command creates the directory MYDIR and adds it to the current directory. The defaults are used for the remaining parameters.

Top

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA089

Pattern not allowed in path name.

CPFA09C

Not authorized to object. Object is &1.

CPFA09D

Error occurred in program &1.

CPFA0A0

Object already exists. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A6

Number of links exceeds maximum allowed for the file system.

CPFA0A7

Path name too long.

CPFA0A9

Object not found. Object is &1.

CPFA0AA

Error occurred while attempting to obtain space.

CPFA0AB

Operation failed for object. Object is &1.

CPFA0AD

Function not supported by file system.

CPFA0B1

Requested operation not allowed. Access problem.

Top

Create Directory (MKDIR)

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

Parameters
Examples
Error messages

The Create Directory (MKDIR) command adds a new directory to the system.

A directory is an object that contains the names of other objects. Libraries and folders are types of directories. When a directory is created, a link is added to the directory prefix. The directory must have been created before any objects can be placed into it.

This command is an alias command for the Create Directory (CRTDIR) command and can also be issued using the following alternative command names:

- CRTDIR
- MD

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- The following restriction applies when the directory to be created is a library in the QSYS.LIB or independent ASP QSYS.LIB file system, or a directory within the "root" (/), QOpenSys, or user-defined file systems:
 - The audit (*AUDIT) special authority is required when specifying a value other than *SYSVAL on the **Auditing value for objects (CRTOBJAUD)** parameter.
- The following restriction applies when the directory to be created is a folder in an existing folder in QDLS:
 - The change (*CHANGE) authority is required for the existing folder.
- The user must have execute (*X) authority to each directory in the path.
- When creating a directory in the "root" (/), QOpenSys or user_defined file system, the user must have write and execute (*WX) authority to the directory that contains the new directory.
- When creating a directory, the owner ID (UID) is the user creating the directory.

If the directory is to be created in the "root" (/), QOpenSys, and user-defined file systems, the following applies. If the S_ISGID bit of the parent directory is off, the group ID (GID) is set to the effective GID of the thread creating the directory. If the S_ISGID bit of the parent directory is on, the group ID (GID) of the new directory is set to the GID of the parent directory.

If the directory is to be created in the QSYS.LIB or independent ASP QSYS.LIB file system, the GID is obtained from the primary user profile. For all other file systems, the GID is obtained from the parent directory.

- The user must have all object (*ALLOBJ) and security administrator (*SECADM) special authorities to specify a value for the **Scanning option for objects (CRTOBJSCAN)** parameter other than *PARENT.

Top

Parameters

Keyword	Description	Choices	Notes
DIR	Directory	<i>Path name</i>	Required, Positional 1
DTAAUT	Public authority for data	<i>Name</i> , *INDIR, *RWX, *RW, *RX, *WX, *R, *W, *X, *EXCLUDE, *NONE	Optional
OBJAUT	Public authority for object	Single values: *INDIR, *NONE, *ALL Other values (up to 4 repetitions): *OBJEXIST, *OBJMGT, *OBJALTER, *OBJREF	Optional
CRTOBJAUD	Auditing value for objects	*SYSVAL, *NONE, *USRPRE, *CHANGE, *ALL	Optional
CRTOBJSCAN	Scanning option for objects	*PARENT, *YES, *NO, *CHGONLY	Optional
RSTDRNMUNL	Restricted rename and unlink	*NO, *YES	Optional

Top

Directory (DIR)

Specifies the path name of the directory to be created.

Note: Do not use a name that begins with the character Q. The system assumes that libraries or directories with those names are system libraries or directories.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

Public authority for data (DTAAUT)

Specifies the public data authority given to the user for the directory, or specifies that all authorities are inherited from the directory it is to be created in.

*INDIR

The authority for the directory to be created is determined by the directory it is to be created in. The directory immediately preceding the new directory determines the authority. A directory created in the "root" (/), QOpenSys, or user-defined file system is assigned the same public, private and primary group authority, authorization list, and primary group as the directory it is to be created in. A directory created in QDLS for a folder defaults to *EXCLUDE for a first level folder. If created in the second level or greater, the authority of the previous level is used. The QOpenSys and "root" (/) file systems use the parent directory's Data Authority value. If the value *INDIR is specified for either the **Public authority for object (OBJAUT)** parameter or the DTAAUT parameter, then *INDIR must be specified for both parameters.

***RWX** The user can change the object and perform basic functions on the object except those limited to the owner or controlled by object existence (*OBJEXIST), object management (*OBJMGT), object alter (*OBJALTER) and object reference (*OBJREF) authorities. Read, write, and execute (*RWX) authority provides object operational (*OBJOPR) and all data authorities.

- *RW** The user can view and change the contents of an object. Read and write (*RW) authority provides *OBJOPR and data read (*READ), add (*ADD), update (*UPD) and delete (*DLT) authorities.
- *RX** The user can perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. Read and execute (*RX) authority provides *OBJOPR and data *READ and execute (*EXECUTE) authorities.
- *WX** The user can change the contents of an object and run a program or search a library or directory. Write and execute (*WX) authority provides *OBJOPR and data *ADD, *UPD, *DLT, and *EXECUTE authorities.
- *R** The user can view the contents of an object. Read (*R) authority provides *OBJOPR and data *READ authorities.
- *W** The user can change the contents of an object. Write (*W) authority provides *OBJOPR and data *ADD, *UPD, and *DLT authorities.
- *X** The user can run a program or search a library or directory. Execute (*X) authority provides *OBJOPR and data *EXECUTE authorities.
- *EXCLUDE**
The user cannot access the object. The OBJAUT value must be *NONE, if this special value is used.
- *NONE**
The user is given no data authorities to the objects. This value cannot be used with the OBJAUT value of *NONE.
- name* Specify the name of the authorization list used. The format of the authorization list name remains the current ten-character format. The OBJAUT value must be *NONE, if this special value is used.

Top

Public authority for object (OBJAUT)

Specifies the public object authority given to users for the directory, or specifies that all authorities are inherited from the directory it is to be created in.

*INDIR

The object authority is based on the authority for the directory where this directory is to be created. A directory created in the "root" (/), QOpenSys, or user-defined file system is assigned the same public, private and primary group authority, authorization list, and primary group as the directory it is to be created in. If the value *INDIR is specified for either the OBJAUT parameter or the **Public authority for data (DTAAUT)** parameter, then *INDIR must be specified for both parameters.

*NONE

None of the other object authorities (*OBJEXIST, *OBJMGT, *OBJALTER or *OBJREF) are given to the users. If *EXCLUDE or an authorization list is specified for the DTAAUT parameter, *NONE must be specified. This value cannot be used with the DTAAUT value of *NONE.

***ALL** All of the other object authorities (*OBJEXIST, *OBJMGT, *OBJALTER or *OBJREF) are given to the users.

The user can specify up to four of the following values:

*OBJEXIST

The user is given object existence (*OBJEXIST) authority to the object. The user can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object.

***OBJMGT**

The user is given object management (*OBJMGT) authority to the object. With this authority the user can specify security for the object, move or rename the object and add members to database files.

***OBJALTER**

The user is given object alter (*OBJALTER) authority to the object. The user is able to alter the attributes of the objects. On a database file, the user can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. With this authority on an SQL package, the user can change the attributes of the SQL package. Currently, this authority is used only for database files and SQL packages.

***OBJREF**

The user is given object reference (*OBJREF) authority to objects. Used only for database files, the user can reference an object from another object such that operations on that object may be restricted by the other object. On a physical file, the user can add a referential constraint in which the physical file is the parent.

Top

Auditing value for objects (CRTOBJAUD)

Specifies the auditing value of objects created in this directory.

Values for this parameter other than *SYSVAL may not be supported by some file systems.

***SYSVAL**

The object auditing value for the objects in the directory is determined by the Create object auditing (QCRTOBJAUD) system value.

***NONE**

Using or changing this object does not cause an audit entry to be sent to the security journal.

***USRPRF**

The user profile of the user accessing this object is used to determine if an audit record is sent for this access. The OBJAUD parameter of the Change User Auditing (CHGUSRAUD) command is used to change the auditing for a specific user.

***CHANGE**

All change accesses to this object by all users are logged.

***ALL** All change or read accesses to this object by all users are logged.

Top

Scanning option for objects (CRTOBJSCAN)

Specifies whether the objects created in a directory will be scanned when exit programs are registered with any of the integrated file system scan-related exit points.

The integrated file system scan-related exit points are:

- QIBM_QP0L_SCAN_OPEN - Integrated File System Scan on Open Exit Program
- QIBM_QP0L_SCAN_CLOSE - Integrated File System Scan on Close Exit Program

For details on these exit points, see the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This attribute can only be specified for directories created in the "root" (/), QOpenSys and user-defined file systems. For all other file systems, *PARENT should be specified and it will be ignored. Even though this attribute can be set for *TYPE1 and *TYPE2 directories, only objects which are in *TYPE2 directories will actually be scanned, no matter what value is set for this attribute.

***PARENT**

The create object scanning attribute value for this directory is copied from the create object scanning attribute value of the parent directory.

***YES** After an object is created in the directory, the object will be scanned according to the rules described in the scan-related exit programs if the object has been modified or if the scanning software has been updated since the last time the object was scanned.

***NO** After an object is created in the directory, the object will not be scanned by the scan-related exit programs.

Note: If the Scan file systems control (QSCANFCTL) value *NOPOSTRST is not specified when an object with this attribute is restored, the object will be scanned at least once after the restore.

***CHGONLY**

After an object is created in the directory, the object will be scanned according to the rules described in the scan-related exit programs only if the object has been modified since the last time the object was scanned. It will not be scanned if the scanning software has been updated. This attribute only takes effect if the Scan file systems control (QSCANFCTL) system value has *USEOCOATR specified. Otherwise, it will be treated as if the attribute is *YES.

Note: If the Scan file systems control (QSCANFCTL) value *NOPOSTRST is not specified when an object with this attribute is restored, the object will be scanned at least once after the restore.

Top

Restricted rename and unlink (RSTDRNMUNL)

Specifies whether special restrictions apply for rename and unlink operations performed on objects within a directory. This attribute is equivalent to the S_ISVTX mode bit and can only be set for a directory in the Network File System (NFS), QFileSvr.400, "root" (/), QOpenSys, or user-defined file systems. Both the NFS and QFileSvr.400 file systems support this attribute by passing it to the server and surfacing it to the caller.

***NO** No additional restrictions for renaming or unlinking objects from this directory.

***YES** Objects within this directory may be renamed or unlinked only if one or more of the following are true for the user performing the operation:

1. The user is the owner of the object.
2. The user is the owner of the directory.
3. The user has all object (*ALLOBJ) special authority.

Top

Examples

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

Example 1: Creating a Directory

```
CRTDIR DIR('MYDIR')
```

This command creates the directory MYDIR and adds it to the current directory. The defaults are used for the remaining parameters.

Top

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA089

Pattern not allowed in path name.

CPFA09C

Not authorized to object. Object is &1.

CPFA09D

Error occurred in program &1.

CPFA0A0

Object already exists. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A3

Path name resolution causes looping.

CPFA0A6

Number of links exceeds maximum allowed for the file system.

CPFA0A7

Path name too long.

CPFA0A9

Object not found. Object is &1.

CPFA0AA

Error occurred while attempting to obtain space.

CPFA0AB

Operation failed for object. Object is &1.

CPFA0AD

Function not supported by file system.

CPFA0B1

Requested operation not allowed. Access problem.

Top

Monitor Message (MONMSG)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The Monitor Message (MONMSG) command is used to monitor escape, notify, and status messages sent to the program message queue of the program in which the command is used. Completion and diagnostic messages cannot be monitored.

When the MONMSG command is compiled in a control language (CL) program, it establishes a monitor for the arrival of the specified messages. The command monitors the messages for the condition specified by the comparison data given in the command. If a message meeting the conditions arrives on the message queue, the CL command specified on the MONMSG command is processed.

Up to 1000 MONMSG commands can be specified in a program to monitor the arrival of messages for specific conditions or for a group of conditions. Specific message identifiers or generic message identifiers can be monitored.

The MONMSG command can be coded following most commands in a CL program or ILE CL procedure. A MONMSG command that is not placed at the beginning of the program applies only to the immediately preceding command; this is called a command-level MONMSG command. The command-level MONMSG command monitors only messages sent by the previous command. If the message sent by that command meets the conditions specified in the MONMSG command, the action specified in the same MONMSG command is taken. As many as 100 MONMSG commands, coded immediately after a command, can monitor the messages sent by that command.

When the action specified in the MONMSG command has been performed, and that action does not end with a GOTO or RETURN command, control returns to the command in the program that follows the command that sent the message. If the action ends with a GOTO command, control branches to the command in the program specified in the GOTO command. If the action ends with a RETURN command, control returns to the program that called the program that contains the MONMSG command.

If one or more MONMSG commands are placed at the beginning of the program, immediately following the declare commands or the PGM command if there are no declare commands, they monitor messages sent by all of the commands in the program (maximum of 100). This is called a program-level MONMSG command. If any message sent by any command in the program meets the conditions specified in any one of the program-level MONMSG commands, the corresponding action specified in the same command is taken.

The action taken by a command-level MONMSG command overrides a program-level MONMSG command.

If a command is coded for the EXEC parameter on a MONMSG command that is placed at the beginning of a program, *only* the GOTO command can be used, and it must specify the label for the command to which control is to be passed if a monitored message occurs. The label specified on a program-level MONMSG command cannot be a label associated with a subroutine. If a GOTO command is performed for a program-level MONMSG, the subroutine stack will be reset by the next Call Subroutine (CALLSUBR) command.

If a command is not coded for the EXEC parameter, monitored messages are ignored.

Restrictions:

- This command is valid only in a CL program or ILE CL procedure.
- It can be coded after the last declare command (if declare commands are used), following the PGM command that begins the program, or it can be coded following any command allowed in a CL program or ILE CL procedure, except for the following: DO, DOWHILE, DOUNTIL, DOFOR, ELSE, ENDDO, SELECT, WHEN, OTHERWISE, ENDSELECT, ENDPGM, CALLSUBR, SUBR, RTNSUBR, ENDSUBR, GOTO, IF, or RETURN. Note that if another program sends a message that is monitored by this command, a return cannot be made to that program.

Top

Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	Values (up to 50 repetitions): <i>Name</i>	Required, Positional 1
CMPDTA	Comparison data	<i>Character value</i> , *NONE	Optional, Positional 2
EXEC	Command to execute	<i>Command string</i>	Optional, Positional 3

Top

Message identifier (MSGID)

Specifies the message identifiers of one or more escape, notify, or status messages that are to be monitored by this command. As many as 50 specific or generic message identifiers can be specified on one command.

Note: Many CL commands issue one escape message for many different error conditions. Details about the error or failure are given in diagnostic messages that precede the escape message. Although diagnostic messages cannot be monitored, they can be received from the job's external message queue after the escape message has activated the user's message monitor.

The first 3 characters of a message identifier must be a code consisting of an alphabetic character followed by two alphanumeric (alphabetic or decimal) characters; the last 4 characters may consist of the decimal numbers 0 through 9 and the characters A through F.

Note: Message identifiers using the MCH code (MCHnnnn) use only the numbers 0 through 9 in the last four characters.

If zeros are specified in either two or all four of the rightmost positions, such as USRmm00, a *generic message identifier* is specified. For example, if CPF0000 is specified, all messages with the prefix 'CPF' are monitored. Generic message identifiers can be used for both command-level MONMSG and procedure-level MONMSG statements.

Specify the message identifiers of 1 to 50 messages that are monitored when they arrive at this program's message queue. The message identifiers and message text of the escape, notify, and status messages which may be sent by a command can be found in the command's documentation in the Information Center as well as the command's online help. CL variables cannot be used to specify message identifiers.

This is a required parameter.

Comparison data (CMPDTA)

Specifies the comparison data that is used to determine whether the monitored message (having one of the specified message identifiers) received on the program's message queue is acted on by this command. The message data specified in the MSGDTA parameter of the Send Program Message (SNDPGMMSG) command is compared with this comparison data. If the first part (up through the first 28 characters, or less) of the message's substitution values matches the comparison data specified, the action specified in the EXEC parameter of this command is taken. The action is also taken if no comparison data is specified.

*NONE

No comparison data is specified. If the message in the program's message queue is from a command that this command is monitoring, and if it has the specified identifier, the action specified for the **Command to execute (EXEC)** parameter is taken.

comparison-data

Specify a character string of no more than 28 characters, enclosed in apostrophes if necessary, that is compared with the same number of characters in the message data of the received message, starting with the first character in the message data. If the comparison data matches the first part of the received message data, this command performs the function specified in the EXEC parameter. A CL variable cannot be specified for the comparison data.

The comparison data can be displayed by the Display Program Variable (DSPPGMVAR) command.

Command to execute (EXEC)

Specifies the CL command to be processed when a monitored message sent to the program's message queue meets the conditions specified in this command. If no command is specified and a monitored message arrives on the queue, the message is ignored, and control passes to the next command in the program.

If the MONMSG command is placed at the beginning of the program, the EXEC parameter must specify the GOTO command and the label identifying the command that receives control.

Specify the CL command, including its parameters to be used, that is run when a message meeting the conditions specified in this command is received. The command specified is not run if the received message does not meet the specified conditions. A CL variable cannot be specified in place of the CL command.

Note: If a DO, DOWHILE, DOUNTIL, DOFOR, or SELECT command is specified on EXEC, the entire group associated with the command is processed if the condition is met.

Examples

Example 1: Monitoring Messages Sent by Any Command

```
PGM
MONMSG MSGID(CPF0001 CPF1999) EXEC(GOTO EXIT2)
```

This example shows a MONMSG command at the beginning of a CL procedure that monitors for messages CPF0001 and CPF1999; these messages might be sent by any command processed later in the procedure. When either message is received from any of the commands running in the procedure, control branches to the command identified by the label EXIT2.

CPF0001 states that an error was found in the command that is identified in the message itself. CPF1999, which can be sent by many of the debugging commands (like CHGPGMVAR), states that errors occurred on the command, but it does not identify the command in the message.

Example 2: Monitoring Messages Sent by a Single Command

```
CHGVAR  VAR(&A)  VALUE(&A / &B)
MONMSG  MSGID(MCH1211) EXEC(CHGVAR  VAR(&A)  VALUE(1))
```

In this example, the MONMSG command follows a Change Variable (CHGVAR) command and, therefore, is only monitoring messages sent by the CHGVAR command. Escape message MCH1211 is sent to this program's message queue when a division by zero is attempted. Because MSGID(MCH1211) is specified, the MONMSG command is monitoring for this condition; when it receives the message, the second CHGVAR command is processed. In this command, the variable &A is set to a value of 1.

[Top](#)

Error messages

None

[Top](#)

Add Mounted FS (MOUNT)

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

Parameters
Examples
Error messages

The Add Mounted File System (MOUNT) command makes the objects in a file system accessible to the integrated file system name space. The file system to be made accessible can be either a user-defined file system (*UDFS) on the local system or a remote file system accessed through local Network File System client (*NFS). The directory that is the destination for the mount, the **Directory to mount over (MNTOVRDIR)**, must exist.

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

- ADDMFS

For more information about Network File System commands, see i5/OS Network File System Support book, SC41-5714

Restrictions:

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
- The user must have write (*W) authority to the directory to be mounted over.
- The user must have execute (*X) authority to each directory in the path.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS	Required, Key, Positional 1
MFS	File system to mount	<i>Path name</i>	Required, Key, Positional 2
MNTOVRDIR	Directory to mount over	<i>Path name</i>	Required, Key, Positional 3
OPTIONS	Mount options	<i>Character value</i> , *DFT	Optional
CCSID	Coded character set ID	<i>Element list</i>	Optional
	Element 1: Data file CCSID	1-65533, *ASCII, *JOBCCSID, *BINARY	
	Element 2: Path name CCSID	1-65533, *ASCII, *JOBCCSID	
CODEPAGE	Code page	<i>Element list</i>	Optional
	Element 1: Data file code page	1-32767, *ASCII, *JOBCCSID, *BINARY	
	Element 2: Path name code page	1-32767, *ASCII, *JOBCCSID	

Top

Type of file system (TYPE)

Specifies the type of file system to be mounted. The type of mount determines the correct form for the **File system to mount (MFS)** parameter.

***NFS** The file system specified for the MFS parameter is a Network File System. The MFS parameter must be of the form *hostname:pathname* where *hostname* can either be the name of a system or an IP address, and *pathname* must be an absolute path name.

*UDFS

The file system specified for the MFS parameter is a user-defined file system. The MFS parameter must be in one of the two following forms:

- */dev/qaspXX/udfsname.udfs* where *XX* is one of the valid system or basic user auxiliary storage pool (ASP) numbers on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.
- */dev/aspname/udfsname.udfs*, where *aspname* is one of the valid independent ASP names on the system, and *udfsname* is the name of the user-defined file system. All other parts of the name must appear as in the example above.

The name part of the path must be unique within the specified *qaspXX* or *aspname* directory.

This is a required parameter.

Top

File system to mount (MFS)

Specifies the path name of the file system to be mounted. It can be the path to a local Block Special File (*BLKSF) or a remote NFS path name. See the **Type of file system (TYPE)** parameter to determine the correct format for the MFS parameter.

This is a required parameter.

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

Top

Directory to mount over (MNTOVRDIR)

Specifies the path name of the existing directory that the file system will be mounted over. This directory gets 'covered' by the mounted file system. This directory must exist.

Multiple file systems can be mounted over the same directory, one on top of the other. However, only the topmost mounted file system is accessible, and the file systems must later be unmounted in the opposite order from which they were mounted (last-in first-out order).

This is a required parameter.

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

Top

Mount options (OPTIONS)

The options list contains a character string of mount options. The options are separated by commas. For some options, an equal '=' and a value follow the option. If an option is not specified, the default value for that option will be used. The options list may contain spaces.

***DFT** The default value for the options string for the mount of a Network File System (*NFS) is:

```
'rw,suid,retry=5,rsize=32768,wsize=32768,timeo=20,retrans=5,acregmin=30,acregmax=60,acdirmin=30,acdirmax=60,hard,async,sec=sys,vers=3:2,cachetimeo=60,nocache'
```

The default value for the options string for the mount of a user-defined file system (*UDFS) is:

```
'rw,suid'
```

For the mount of a Network File System, all of the following options are valid. For the mount of a user-defined file system, only the *ro*, *rw*, *suid* and *nosuid* options are valid. If options are specified that are not valid for the file system type to be mounted, they are ignored.

options-list

The following are the available options and their descriptions:

rw | ro This option specifies the protection for the mounted file system. Either *ro* (read-only) or *rw* (read-write) may be specified. If neither is specified, *rw* is assumed.

suid | nosuid

For the mount of a user-defined file system or a Network File System, if *suid* is specified, setuid execution is allowed. This means that bits other than the permission bits may be set. If *nosuid* is specified, setuid execution is not allowed.

hard | soft

For the mount of a Network File System, specifies whether NFS file systems are hard or soft mounted. Hard mounted means that operations on them are retried until they are acknowledged by the server. Soft mounted means that a timeout error is returned if a remote operation fails the number of times specified on the *retrans* option. If neither is specified, *hard* is assumed.

rsize=n

For the mount of a Network File System, specifies the size of the read buffer in bytes. The read buffer is used for data transfer between the NFS client and the remote NFS server on an NFS read request. The allowed range is 512 to 32768. If *rsize* is not specified, the default value of 32768 is assumed. For better performance, the read buffer should be a multiple of the the application buffer size.

wsize=n

For the mount of a Network File System, specifies the size of the write buffer in bytes. The write buffer is used for data transfer between the NFS client and the remote NFS server on an NFS write request. The allowed range is 512 to 32768. If *wsize* is not specified, the default value of 32768 is assumed. For better performance, the write buffer should be a multiple of the application buffer size.

timeo=n

For the mount of a Network File System, specifies the amount of time, in tenths of seconds, to wait for the client to respond on each try. The allowed range is 0 to 10000. If *timeo* is not specified, the default value of 20 tenths of a second (2 seconds) is assumed.

retry=n

For the mount of a Network File System, specifies the number of times to retry the mount operation. The allowed range is 0 to 10000. If *retry* is not specified, the default value of 5 retransmission attempts is assumed.

retrans=n

For the mount of a Network File System, specifies the number of times to retry the transmission to the server. The allowed range is 0 to 10. If *retrans* is not specified, the default value of 5 retransmission attempts is assumed.

acregmin=n

For the mount of a Network File System, specifies the minimum number of seconds to hold locally stored file attributes after file updates. The allowed range is 1 to 3600. If *acregmin* is not specified, the default value of 30 seconds is assumed.

acregmax=n

For the mount of a Network File System, specifies the maximum number of seconds to hold locally stored file attributes after file updates. The allowed range is 1 to 2,000,000,000. If *acregmax* is not specified, the default value of 60 seconds is assumed.

acdirmin=n

For the mount of a Network File System, specifies the minimum number of seconds to hold locally stored directory attributes after a directory update. The allowed range is 1 to 3600. If *acdirmin* is not specified, the default value of 30 seconds is assumed.

acdirmax=n

For the mount of a Network File System, specifies the maximum number of seconds to hold locally stored directory attributes after a directory update. The allowed range is 1 to 2,000,000,000. If *acdirmax* is not specified the default value of 60 seconds is assumed.

nocto For the mount of a Network File System, specifies whether to force the refresh of remote attributes when opening a file. If this option is specified, attributes are not refreshed from the server when opening a file, and changes are not sent to the server on the last close. If *nocto* is not specified, the default value of no suppression is assumed.

noac For the mount of a Network File System, specifies whether to suppress local storage of attributes and names. If this option is specified, local storage of attributes and names is suppressed. If *noac* is not specified, the default value of no suppression is assumed. If *noac* is specified, values specified for *acregmin*, *acregmax*, *acdirmin*, and *acdirmax* may be specified but are not used.

async For the mount of a Network File System, specifies whether to allow asynchronous write requests. If *async* is not specified, only synchronous write requests will be used.

cachetimeo=n

For the mount of a Network File System, specifies the amount of time in seconds, for the client to revalidate data cache consistency. The allowed range is 1 to 10000. If *cachetimeo* is not specified, the default value of 60 seconds is assumed.

nocache

For the mount of a Network File System, specifies whether to disable the data cache. If *nocache* is not specified, data cache will be enabled by default.

sec=flavor[:flavor...]

For the mount of a Network File System, specifies an ordered list of security flavors that may be used to access the mount point. Allowable flavor values are:

sys UNIX-like (user ids, group ids).

krb5 Kerberos 5, no integrity or privacy. Only valid when NFS version 4 specified.

krb5i Kerberos 5, with integrity. Only valid when NFS version 4 specified.

krb5p Kerberos 5, with privacy. Only valid when NFS version 4 specified.

vers=version[:version...]

For the mount of a Network File System, specifies NFS versions allowed for the mount. If this option is specified, only the specified NFS versions will be used to attempt the mount, in the order they are provided. The default used if this option is not specified is vers=3:2. Allowable version numbers are: 2:3:4.

Top

Coded character set ID (CCSID)

Specifies, for Network File Systems, a pair of coded character set identifiers (CCSIDs) to identify a specific character representation to be used. The first CCSID specifies what encoding scheme should be assumed for data files on the remote system. The second CCSID specifies what encoding scheme should be assumed for path names on the remote system.

This parameter is only valid if mounting a Network File System.

Element 1: Data file CCSID

***BINARY**

No conversion is used.

***ASCII**

The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**

The CCSID from the default job CCSID is used.

1-65533

Specify a CCSID to be assumed for data files on the remote system.

Element 2: Path name CCSID

***ASCII**

The ASCII equivalent of the default job CCSID associated with the current job is used.

***JOBCCSID**

The CCSID from the default job CCSID is used.

1-65533

Specify a CCSID to be assumed for path names on the remote system. Only CCSIDs that can be converted into UCS-2 level 1 (1200) are supported. See Globalization information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for a list of supported conversions.

Top

Code page (CODEPAGE)

Specifies, for Network File Systems, a pair of code pages. The first code page specifies what code page should be assumed for data files on the remote system. The second code page specifies what code page should be assumed for path names on the remote system.

This parameter is only valid if mounting a Network File System.

Note: This parameter is replaced by **Coded character set ID (CCSID)** but the CODEPAGE parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the CCSID parameter.

Element 1: Data file code page

Note: A code page that has the same number of bytes per character as the original data should be specified.

*BINARY

No conversion is used.

*ASCII

The ASCII equivalent of the default job coded character set identifier (CCSID) associated with the current job is used.

*JOBCCSID

The default job coded character set identifier (CCSID) associated with the current job is used.

1-32767

Specify a code page to be assumed for data files on the remote system. Only code pages that correspond to single-byte or double-byte encoding schemes are supported. Code pages that correspond to mixed-byte encoding schemes are not supported.

Element 2: Path name code page

*ASCII

The ASCII equivalent of the default job coded character set identifier (CCSID) associated with the current job is used.

*JOBCCSID

The default job coded character set identifier (CCSID) associated with the current job is used.

1-32767

Specify a code page to be assumed for path names on the remote system. Only code pages whose CCSIDs can be converted into UCS-2 level 1 (1200) are supported. See Globalization information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for a list of supported conversions.

Top

Examples

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

Example 1: Mounting a User-Defined File System

```
ADDMFS TYPE(*UDFS) MFS('/DEV/QASP03/PROD1')
      MNTOVRDIR('DIRB')
```

This command mounts a user-defined file system PROD1 over the directory, DIRB. It uses the defaults for the other parameters.

Example 2: Mounting a Network File System

```
ADDMFS TYPE(*NFS) MFS('RAINFALL:/QSYS.LIB/RAY.LIB')
      MNTOVRDIR('/mystuff')
```

This command mounts the **/qsys.lib/ray.lib** file system from the remote system RAINFALL into the directory **/mystuff**.

Example 3: Mounting a Network File System with OPTIONS

```
ADDMFS TYPE(*NFS) MFS('RAINFALL:/QSYS.LIB/RAY.LIB')
MNTOVRDIR('/mystuff')
OPTIONS('ro,nosuid,rsize=256, retrans=10')
CODEPAGE(*ASCII *JOBCCSID) CCSID(*ASCII *JOBCCSID)
```

This command mounts the **/qsys.lib/ray.lib** file system from the remote system RAINFALL into the directory **/mystuff**. In addition it specifies to mount as read-only, not allow setuid execution, set the read buffer to 256 bytes, and the retransmission attempts to 10. The job CCSID is used to determine the coded character set identifier to use for remote path names.

Top

Error messages

*ESCAPE Messages

CPDBCC2

A non-recoverable error occurred when attempting to resolve the name.

CPF3BCB

Encoding scheme &1 of CCSID &2 not supported.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A2

Information passed to this operation was not valid.

CPFA0A9

Object not found. Object is &1.

CPFA0D0

CCSID conversion error occurred.

CPFA0D9

Character string not converted.

CPFA1B0

Unrecognized option found on options list.

CPFA1B8

*IOSYSCFG authority required to use &1.

CPFA1C6

The value supplied for the File system to mount (MFS) parameter does not have the correct form.

CPFA1CE

Cannot find an address for the specified system name.

Top

Move Object (MOV)

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

Parameters
Examples
Error messages

The Move Object (MOV) command moves an object from the directory it is in to a different directory.

If the **To directory (TODIR)** parameter is used, the object is moved to another directory and the object keeps the same name. If the **To object (TOOBJ)** parameter is used the object is also renamed.

If the original object is a read-only file (a file that has the PC read-only attribute flag turned on), the move command operates as follows:

1. If the original file can be deleted (that is, the read-only bit can be turned off for the file), the move will succeed, retaining the read-only attribute of the file.
2. If the original file cannot be deleted, (for example, a CD-ROM file), the move operation will fail and a message will be issued indicating that the source is read-only.

When moving a file within a file system, the Last access date/time, the Data change date/time and the Attribute change date/time are preserved in the new file. If the file is moved outside of the original file system to the "root" (/), QOpenSys, QDLS, or UDFS file systems, the Attribute change date/time is changed to the current time. In the case of moving to a database file member (*MBR) in the QSYS.LIB or independent ASP QSYS.LIB file system, the Data change date/time is updated as well.

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

- MOVE

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- The directory to which the object is to be moved must not already contain the name supplied in the TOOBJ parameter (or in the case where TODIR is used, the name supplied in OBJ cannot exist in TODIR).
- Only objects that are a stream file type move between file systems.
- A directory cannot be moved to a subordinate directory.
- Database file members cannot be moved.
- Objects in QDLS can not be moved between auxiliary storage pools (ASPs).
- Libraries in independent ASP QSYS.LIB can not be moved to basic auxiliary storage pools (ASPs). However libraries in independent ASP QSYS.LIB can be moved to the system ASP or other independent ASPs.
- The move command does not copy the private authorities for objects when moving from one file system to another file system.

Note: The authority requirements for this command are complex with respect to file systems, object types, requested operations etc.. Therefore, see the System i Security Reference, SC41-5302 book for information about the required authorities for this command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
TODIR	To directory	<i>Path name, '.'</i>	Optional, Positional 2
TOOBJ	To object	<i>Path name</i>	Optional
FROMCCSID	From CCSID	1-65533, * OBJ , *PCASCII, *JOBCCSID	Optional
TOCCSID	To CCSID	1-65533, * OBJ , *CALC, *STDASCII, *PCASCII, *JOBCCSID	Optional
DTAFMT	Data Format	* BINARY , *TEXT	Optional
FROMCODPAG	From code page	1-32767, * OBJ , *PCASCII	Optional
TOCODEPAGE	To code page	1-32767, * OBJ , *CALC, *STDASCII, *PCASCII	Optional

Top

Object (OBJ)

Specifies the path name of the object or objects to be moved.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

Note: An object name pattern can only be used when the **To directory (TODIR)** parameter is used.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

To directory (TODIR)

Specifies the path name of the directory to which the object is to be moved. The moved object uses the name supplied on the **Object (OBJ)** parameter.

: The path object moves to the current directory.

directory-name

Specify the name of the directory to which the object is to be moved.

Note: The TODIR and **To object (TOOBJ)** parameters are mutually exclusive.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

To object (TOOBJ)

Specifies the path name of the directory the object is to be moved to and the new name of the object.

Note: The **To directory (TODIR)** and **TOOBJ** parameters are mutually exclusive.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

From CCSID (FROMCCSID)

Specifies the method for obtaining the coded character set identifier (CCSID) for the source of the move operation. This CCSID will be used for data conversion, if requested. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

***OBJ** Use the data CCSID of the object to be moved.

***PCASCII**

Use the data CCSID of the object to be moved to compute a CCSID in the Microsoft Windows encoding scheme (x4105) (Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation). Use this as the CCSID from which the data will be converted when **DTAFMT(*TEXT)** is specified. This option allows data from PCs to be converted properly if the data was created using Microsoft Windows.

***JOBCCSID**

The coded character set identifier (CCSID) from the default job CCSID is used.

1-65533

Specify a CCSID value.

Top

To CCSID (TOCCSID)

Specifies the data coded character set identifier (CCSID) for the target of the move operation. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

***OBJ** Use the data CCSID of the object to be moved. If this CCSID cannot be used by the file system that the object is to be moved into, the move operation will fail.

***CALC**

Use the data CCSID of the object to be moved. If this CCSID cannot be used by the file system that the object is to be moved into, allow the file system to determine a different CCSID and continue with the move.

***STDASCII**

Compute a CCSID in the IBM PC Data encoding scheme (x2100), based on the source file's CCSID. Associate this CCSID for the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. If this CCSID cannot be used by the file system that the object is to be copied into, the move operation will fail.

***PCASCII**

Compute a CCSID in the Microsoft Windows encoding scheme (x4105), based on the source file's CCSID. Associate this CCSID with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this CCSID cannot be used by the file system that the object is to be moved into, the move operation will fail.

***JOBCCSID**

The coded character set identifier (CCSID) from the default job CCSID is used.

1-65533

Specify a CCSID value. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

Top

Data Format (DTAFMT)

Specifies the format of the data in the file to be moved.

***BINARY**

The file contains data in binary form (such as an executable file). Do not convert data on the move. However, if the object to be moved to has a different CCSID than the source object, all extended attributes will be converted into the CCSID of the new object before being set.

***TEXT**

The file contains data in textual form. Convert data to the CCSID of the new object during the move. The data is processed as text during the move.

If a database member is to be moved to a stream file, any line-formatting characters (such as carriage return, tab, and end-of-file) are just converted from one CCSID to another.

If a stream file is to be moved to a database member, the stream file must contain end-of-line characters or the move will fail. If the stream file does contain end-of-line characters, the following actions are performed during the move to a database file.

- End-of-line characters are removed.
- Records are padded with blanks (for a source physical file member) or nulls (for a data physical file member).
- Tab characters are replaced by the appropriate number of blanks to the next tab position.

Top

From code page (FROMCODPAG)

Specifies the method for obtaining the code page for source of the move operation. This code page will be used for data conversion, if requested. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

Note: This parameter is replaced by the **From CCSID (FROMCCSID)** parameter, but the FROMCODPAG parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the FROMCCSID parameter.

***OBJ** Use the data code page of the object to be moved.

***PCASCII**

Use the data code page of the object to be moved to compute a code page in the Microsoft Windows encoding scheme (x4105) (Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation). Use this as the code page from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly if the data was created using Microsoft Windows.

1-32767

Specify a code page value.

Top

To code page (TOCODEPAGE)

Specifies the data code page for the target of the move operation. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

Note: This parameter is replaced by **To CCSID (TOCCSID)**, but the TOCODEPAGE parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the TOCCSID parameter.

***OBJ** Use the data code page of the object to be moved. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

***CALC**

Use the data code page of the object to be moved. If this code page cannot be used by the file system that the object is to be moved into, allow the file system to determine a different code page and continue with the move.

***STDASCII**

Compute a code page in the IBM PC Data encoding scheme (x2100), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

***PCASCII**

Compute a code page in the Microsoft Windows encoding scheme (x4105), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

1-32767

Specify a code page value. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

Examples

Example 1: Moving an Object

```
MOV  OBJ('/CURRENT/DECEMBER-1994-MONTHLY-PAYROLL-FILE')
     TODIR('/ARCHIVE')
```

This command moves a file named DECEMBER-1994-MONTHLY-PAYROLL-FILE from a directory named CURRENT to a directory named ARCHIVE.

Example 2: Moving with Conversion

```
MOV  OBJ('/DATAFB')
     TOOBJ('/QSYS.LIB/APP1.LIB/DATA.FILE/DATAFB.MBR')
     TOCODEPAGE(*CALC) DTAFMT(*TEXT) TOCCSID(*CALC)
```

The stream file 'DATAFB' is to be moved to the database file 'DATAFB.MBR'. By specifying TOCCSID(*CALC), the file system being moved to (the QSYS.LIB file system in this case) will try to create the new member in the same CCSID as '/DATAFB'. If this fails (in this case, if 'DATA.FILE' is not in the same CCSID as 'DATAFB'), the file system will be allowed to choose an appropriate CCSID and complete the move. By specifying DTAFMT(*TEXT), the data in 'DATAFB' is handled as text and is converted into the CCSID chosen for the new file 'DATAFB.MBR'.

Top

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA08E

More than one name matches pattern.

CPFA093

Name matching pattern not found.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0B0

Request not allowed to operate from one file system to another.

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0B2

No objects satisfy request.

CPFA0B8

&3 objects moved. &4 objects failed.

CPFA0C4

Object not a file. Object is &1.

[Top](#)

Move Document (MOVDOC)

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

Parameters
Examples
Error messages

The Move Document (MOVDOC) command changes the path the system uses to search for a document. The document is **not** physically moved to another location of auxiliary storage, and a new object is **not** created.

Restrictions:

- You must be enrolled in the System Directory and have all (*ALL) authority to the document being moved and have change (*CHANGE) authority to both the FROM and TO folders (if applicable).
- To move a document to or from a folder, the user must have *CHANGE authority to the folder.
- Documents cannot be moved between folders that reside in different auxiliary storage pools (ASPs).

Top

Parameters

Keyword	Description	Choices	Notes
FROMDOC	From document	Character value, *SYSOBJNAM	Required, Positional 1
FROMFLR	From folder	Character value, *NONE	Optional, Positional 2
TOFLR	To folder	Character value, *NONE	Optional, Positional 3
RENAME	Rename	Character value, *SAME	Optional, Positional 4
SYSOBJNAM	System object name	Name	Optional

Top

From document (FROMDOC)

Specifies the document to be moved. If a document name is specified on the FROMDOC parameter, then a folder name must be specified on the **From folder (FROMFLR)** parameter. If *SYSOBJNAM is specified on the FROMDOC parameter, then a system object name must be specified on the **System object name (SYSOBJNAM)** parameter.

Note: If FROMDOC(name) is specified, then FROMFLR(name) is required. If FROMDOC(*SYSOBJNAM) is specified, then SYSOBJNAM(name) and RENAME(name) are required.

This is a required parameter.

*SYSOBJNAM

A system object name is used to identify the document to be moved. This parameter must be used to move a folderless document and may be used instead of a folder/document name any time the system object name is known.

name Specify the name of the document to be moved.

From folder (FROMFLR)

Specifies the name of the folder from which the document is being moved. A folder name must be entered on this parameter if a document name is entered on the **From document (FROMDOC)** parameter. FROMFLR(*NONE) cannot be specified if FROMDOC(name) is specified.

*NONE

The document to be moved is specified by its system object name.

name Specify the name of the folder containing the document to be moved.

Top

To folder (TOFLR)

Specifies the name of the folder into which the document is being moved. A folder name must be entered in this parameter if a document name is entered in the **Rename (RENAME)** parameter.

*NONE

The document is to become a folderless object. If you specify TOFLR(*NONE), the document becomes folderless and can only be referred to by its system object name.

name Specify the name of the folder that is to contain the document.

Top

Rename (RENAME)

Specifies the name by which the moved document is to be known in the TOFLR folder. This parameter allows the user to name a document when moving a folderless document to a folder. It also allows the user to rename the document when moving it from one folder to another.

If the user wants to move a document into a folder, the name of the document in the TOFLR folder must be unique.

If the new name is already assigned to a folder or a document in a folder specified on the TOFLR parameter, the user must either choose a new name for the target document or rename the folder or document that has the same name.

Note: If FROMDOC(*SYSOBJNAM) is specified, then RENAME(name) must be specified.

*SAME

The document name does not change when moving it from one folder to another, or the document will no longer have a name when making it folderless.

name Specify the name of the moved document in the TOFLR folder.

Top

System object name (SYSOBJNAM)

Specifies the system object name of the document to be moved. This parameter can only be specified if FROMDOC(*SYSOBJNAM) is specified.

name Specify the system object name.

Examples

Example 1: Adding a Folderless Document

```
MOVDOC FROMDOC(*SYSOBJNAM) FROMFLR(*NONE) TOFLR(FLR1)
        RENAME(DOC1) SYSOBJNAM(CNTR192366)
```

This command, whose system object name is CNTR192366, adds a folderless document to FLR1 and names it DOC1.

Example 2: Moving a Document and Keeping its Name

```
MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(FLR2)
        RENAME(*SAME)
```

This command moves DOC1 from FLR1 to FLR2 and keeps the name DOC1.

Example 3: Moving and Renaming a Document

```
MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(FLR2)
        RENAME(DOC2)
```

This command moves DOC1 from FLR1 to FLR2 and renames it DOC2.

Example 4: Moving a Document and Making It Folderless

```
MOVDOC FROMDOC(DOC1) FROMFLR(FLR1) TOFLR(*NONE)
```

This command moves DOC1 from FLR1 and changes it to a folderless document.

Error messages

*ESCAPE Messages

CPF8A13

Document &2 in folder &1 not moved.

Move Object (MOVE)

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

Parameters
Examples
Error messages

The Move Object (MOVE) command moves an object from the directory it is in to a different directory.

If the **To directory (TODIR)** parameter is used, the object is moved to another directory and the object keeps the same name. If the **To object (TOOBJ)** parameter is used the object is also renamed.

If the original object is a read-only file (a file that has the PC read-only attribute flag turned on), the move command operates as follows:

1. If the original file can be deleted (that is, the read-only bit can be turned off for the file), the move will succeed, retaining the read-only attribute of the file.
2. If the original file cannot be deleted, (for example, a CD-ROM file), the move operation will fail and a message will be issued indicating that the source is read-only.

When moving a file within a file system, the Last access date/time, the Data change date/time and the Attribute change date/time are preserved in the new file. If the file is moved outside of the original file system to the "root" (/), QOpenSys, QDLS, or UDFS file systems, the Attribute change date/time is changed to the current time. In the case of moving to a database file member (*MBR) in the QSYS.LIB or independent ASP QSYS.LIB file system, the Data change date/time is updated as well.

This command is an alias for the Move Object (MOV) command and can also be issued using the following alternative command name:

- MOV

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- The directory to which the object is to be moved must not already contain the name supplied in the TOOBJ parameter (or in the case where TODIR is used, the name supplied in OBJ cannot exist in TODIR).
- Only objects that are a stream file type move between file systems.
- A directory cannot be moved to a subordinate directory.
- Database file members cannot be moved.
- Objects in QDLS can not be moved between auxiliary storage pools (ASPs).
- Libraries in independent ASP QSYS.LIB can not be moved to basic auxiliary storage pools (ASPs). However libraries in independent ASP QSYS.LIB can be moved to the system ASP or other independent ASPs.
- The move command does not copy the private authorities for objects when moving from one file system to another file system.

Note: The authority requirements for this command are complex with respect to file systems, object types, requested operations etc.. Therefore, see the System i Security Reference, SC41-5302 book for information about the required authorities for this command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
TODIR	To directory	<i>Path name, '.'</i>	Optional, Positional 2
TOOBJ	To object	<i>Path name</i>	Optional
FROMCCSID	From CCSID	1-65533, *OBJ , *PCASCII, *JOBCCSID	Optional
TOCCSID	To CCSID	1-65533, *OBJ , *CALC, *STDASCII, *PCASCII, *JOBCCSID	Optional
DTAFMT	Data Format	*BINARY , *TEXT	Optional
FROMCODPAG	From code page	1-32767, *OBJ , *PCASCII	Optional
TOCODEPAGE	To code page	1-32767, *OBJ , *CALC, *STDASCII, *PCASCII	Optional

Top

Object (OBJ)

Specifies the path name of the object or objects to be moved.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

Note: An object name pattern can only be used when the **To directory (TODIR)** parameter is used.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

To directory (TODIR)

Specifies the path name of the directory to which the object is to be moved. The moved object uses the name supplied on the **Object (OBJ)** parameter.

: The path object moves to the current directory.

directory-name

Specify the name of the directory to which the object is to be moved.

Note: The TODIR and **To object (TOOBJ)** parameters are mutually exclusive.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

To object (TOOBJ)

Specifies the path name of the directory the object is to be moved to and the new name of the object.

Note: The **To directory (TODIR)** and **TOOBJ** parameters are mutually exclusive.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

From CCSID (FROMCCSID)

Specifies the method for obtaining the coded character set identifier (CCSID) for the source of the move operation. This CCSID will be used for data conversion, if requested. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

***OBJ** Use the data CCSID of the object to be moved.

***PCASCII**

Use the data CCSID of the object to be moved to compute a CCSID in the Microsoft Windows encoding scheme (x4105) (Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation). Use this as the CCSID from which the data will be converted when **DTAFMT(*TEXT)** is specified. This option allows data from PCs to be converted properly if the data was created using Microsoft Windows.

***JOBCCSID**

The coded character set identifier (CCSID) from the default job CCSID is used.

1-65533

Specify a CCSID value.

Top

To CCSID (TOCCSID)

Specifies the data coded character set identifier (CCSID) for the target of the move operation. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

***OBJ** Use the data CCSID of the object to be moved. If this CCSID cannot be used by the file system that the object is to be moved into, the move operation will fail.

***CALC**

Use the data CCSID of the object to be moved. If this CCSID cannot be used by the file system that the object is to be moved into, allow the file system to determine a different CCSID and continue with the move.

***STDASCII**

Compute a CCSID in the IBM PC Data encoding scheme (x2100), based on the source file's CCSID. Associate this CCSID for the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. If this CCSID cannot be used by the file system that the object is to be copied into, the move operation will fail.

***PCASCII**

Compute a CCSID in the Microsoft Windows encoding scheme (x4105), based on the source file's CCSID. Associate this CCSID with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this CCSID for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this CCSID cannot be used by the file system that the object is to be moved into, the move operation will fail.

***JOBCCSID**

The coded character set identifier (CCSID) from the default job CCSID is used.

1-65533

Specify a CCSID value. If this CCSID cannot be used by the file system that the object is being moved into, the move operation will fail.

Top

Data Format (DTAFMT)

Specifies the format of the data in the file to be moved.

***BINARY**

The file contains data in binary form (such as an executable file). Do not convert data on the move. However, if the object to be moved to has a different CCSID than the source object, all extended attributes will be converted into the CCSID of the new object before being set.

***TEXT**

The file contains data in textual form. Convert data to the CCSID of the new object during the move. The data is processed as text during the move.

If a database member is to be moved to a stream file, any line-formatting characters (such as carriage return, tab, and end-of-file) are just converted from one CCSID to another.

If a stream file is to be moved to a database member, the stream file must contain end-of-line characters or the move will fail. If the stream file does contain end-of-line characters, the following actions are performed during the move to a database file.

- End-of-line characters are removed.
- Records are padded with blanks (for a source physical file member) or nulls (for a data physical file member).
- Tab characters are replaced by the appropriate number of blanks to the next tab position.

Top

From code page (FROMCODPAG)

Specifies the method for obtaining the code page for source of the move operation. This code page will be used for data conversion, if requested. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

Note: This parameter is replaced by the **From CCSID (FROMCCSID)** parameter, but the FROMCODPAG parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the FROMCCSID parameter.

***OBJ** Use the data code page of the object to be moved.

***PCASCII**

Use the data code page of the object to be moved to compute a code page in the Microsoft Windows encoding scheme (x4105) (Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation). Use this as the code page from which the data will be converted when DTAFMT(*TEXT) is specified. This option allows data from PCs to be converted properly if the data was created using Microsoft Windows.

1-32767

Specify a code page value.

Top

To code page (TOCODEPAGE)

Specifies the data code page for the target of the move operation. This parameter is ignored if the object specified on the **Object (OBJ)** parameter is not a regular file. A regular file is a file that supports the integrated file system input/output (I/O) operations open, read, and write.

Note: This parameter is replaced by **To CCSID (TOCCSID)**, but the TOCODEPAGE parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the TOCCSID parameter.

***OBJ** Use the data code page of the object to be moved. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

***CALC**

Use the data code page of the object to be moved. If this code page cannot be used by the file system that the object is to be moved into, allow the file system to determine a different code page and continue with the move.

***STDASCII**

Compute a code page in the IBM PC Data encoding scheme (x2100), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

***PCASCII**

Compute a code page in the Microsoft Windows encoding scheme (x4105), based on the source file's code page. Associate this code page with the target of the move operation and, if DTAFMT(*TEXT) is specified, also use this code page for the data conversion. This option allows the resulting data to be used by Microsoft Windows applications. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

1-32767

Specify a code page value. If this code page cannot be used by the file system that the object is to be moved into, the move operation will fail.

Examples

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

Example 1: Moving an Object

```
MOV  OBJ('/CURRENT/DECEMBER-1994-MONTHLY-PAYROLL-FILE')
     TODIR('/ARCHIVE')
```

This command moves a file named DECEMBER-1994-MONTHLY-PAYROLL-FILE from a directory named CURRENT to a directory named ARCHIVE.

Example 2: Moving with Conversion

```
MOV  OBJ('/DATAFB')
     TOOBJ('/QSYS.LIB/APP1.LIB/DATA.FILE/DATAFB.MBR')
     TOCODEPAGE(*CALC) DTAFMT(*TEXT) TOCCSID(*CALC)
```

The stream file 'DATAFB' is to be moved to the database file 'DATAFB.MBR'. By specifying TOCCSID(*CALC), the file system being moved to (the QSYS.LIB file system in this case) will try to create the new member in the same CCSID as '/DATAFB'. If this fails (in this case, if 'DATA.FILE' is not in the same CCSID as 'DATAFB'), the file system will be allowed to choose an appropriate CCSID and complete the move. By specifying DTAFMT(*TEXT), the data in 'DATAFB' is handled as text and is converted into the CCSID chosen for the new file 'DATAFB.MBR'.

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA08E

More than one name matches pattern.

CPFA093

Name matching pattern not found.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0B0

Request not allowed to operate from one file system to another.

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0B2

No objects satisfy request.

CPFA0B8

&3 objects moved. &4 objects failed.

CPFA0C4

Object not a file. Object is &1.

[Top](#)

Move Object (MOV OBJ)

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

Parameters
Examples
Error messages

The Move Object (MOV OBJ) command removes an object from its currently assigned library and places it in a different library. The save and restore information is removed from the object description.

Note: The value of the **Create authority (CRTAUT)** parameter specified on the Create Library (CRTLIB) command for the to-library is not used. The ownership and public and private authorities of the object remain the same.

If the object being moved is currently journaled, an entry is written to the journal recording the change. Use the Display Object Description (DSPOBJD) command to display journal information for the object.

When the object being moved is an object type which is eligible to be journaled, there are two methods for the object to automatically start journaling after the object has been moved.

1. If the target library is journaled, the journal inherit rules for the library will determine whether or not journaling is started for the object.
2. If the target library contains a data area named QDFTJRN, the object will automatically start journaling based on the contents of the QDFTJRN data area.

Note: The QDFTJRN data area overrides the journaling state and journal inherit rules of the target library.

Note: Support of the QDFTJRN data area will be discontinued in a future release.

Use the Display Library Description (DSPLIBD) command to display journal information for the library. Refer to the Start Journal Library (STRJRNLIB) command for more information about journaling a library.

With the exception noted below in the considerations for moving an object from QTEMP to a primary or secondary ASP, if the object was journaled in the from-library, it will continue to be journaled in the to-library.

Note: For additional information regarding journaling, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

1. For an object other than a *FILE object, you must have:
 - object management (*OBJMGT) authority for the object which is to be moved,
 - delete (*DELETE) and read (*READ) authorities for the library from which the object is to be moved, and
 - add (*ADD) and read (*READ) authorities for the library to which the object is to be moved.
2. For a *FILE object, you must have:
 - object management (*OBJMGT) and object operational (*OBJOPR) authority for the object which is to be moved,
 - delete (*DELETE), add (*ADD), and execute (*EXECUTE) authorities for the library from which the object is to be moved, and
 - add (*ADD) and read (*READ) authorities for the library to which the object is to be moved.
3. The following object types cannot be moved: libraries, user profiles, edit descriptions, line descriptions, controller descriptions, device descriptions, journals, and journal receivers.

4. The following objects cannot be moved: the system operator message queue QSYSOPR, all workstation user message queues, and the system log QHST.
5. The library to which the object is to be moved must not already contain an object of the same name and type as the object to be moved.
6. The library to which the object is to be moved cannot be QTEMP.
7. The user space (*USRSPC), user index (*USRIDX), and user queue (*USRQ) user domain objects can only be moved into libraries that are permitted in the system value QALWUSRDMN (allow user domain objects in library). However, if the user object was created as a system domain object, it is not restricted.
8. As a general rule, objects cannot be moved to the to-library if the object and the to-library are in different auxiliary storage pools (ASPs). An error message is sent when the object cannot be moved. There are some specific exceptions to the general rule:
 - You can move save files that are in a basic user ASP to libraries that are in the system ASP (ASP 1) if the save file's library is also in the system ASP.
 - You can move objects in a secondary ASP to the primary ASP in the same ASP group if the to-library is QRPLxxxx (where 'xxxx' is the number of the primary ASP of the ASP group.)
 - You can move an object from QTEMP to a primary or secondary ASP with the following considerations:
 - The 'move' is accomplished through a save and restore operation.
 - The size of the object must be less than 1 terabyte. (The Move Library to ASP (QHSMMOV) API does not have this size limitation.)
 - If the object cannot be renamed, it cannot be moved.
 - For data queues, message queues, and logical files, only the object descriptions are moved. The contents of the objects are not moved.
 - The private authorities for the objects will be preserved.
 - After the object has been moved, the following attributes will differ from the original object:
 - The date last used will be set to blank.
 - The change date and time will be set to the current date and time.
 - The days used count will be set to zero.
 - The date use count reset will be set to blank.
 - The restore date and time will be set to the current date and time.
 - When the object being moved is an object type which is eligible to be journaled, there are two methods for the object to automatically start journaling.
 - a. If the target library is journaled, the journal inherit rules for the library will determine whether or not journaling is started for the object.
 - b. If the target library contains a data area named QDFTJRN, the object will automatically start journaling based on the contents of the QDFTJRN data area.

Note: The QDFTJRN data area overrides the journaling state and journal inherit rules of the target library.

Note: Support of the QDFTJRN data area will be discontinued in a future release.

If the object will not be automatically journaled for a restore operation for the object type, the object will not be journaled even if the original object was journaled.

Note: For additional information regarding journaling, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OBJTYPE	Object type	*ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *JOBQ, *JOBQ, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SRVPGM, *SSND, *SVRSTG, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
TOLIB	To library	<i>Name, *CURLIB</i>	Required, Positional 3
ASPDEV	From ASP device	<i>Name, *, *CURASPGRP, *SYSBAS</i>	Optional
TOASPDEV	To ASP device	<i>Name, *ASPDEV, *, *CURASPGRP, *SYSBAS</i>	Optional

Top

Object (OBJ)

Specifies the object to be moved to another library.

This is a required parameter.

Qualifier 1: Object

Specify the name of the object to be moved.

Qualifier 2: Library

***LIBL** All libraries in the library list for the current thread are searched until the first match is found. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

name Specify the name of the library to be searched.

Top

Object type (OBJTYPE)

Specifies the object type of the object to be moved.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

object-type

Specify the object type of the object to be moved.

Top

To library (TOLIB)

Specifies the library where the object is to be moved. The library QTEMP cannot be specified.

This is a required parameter.

***CURLIB**

The object is to be moved to the current library. If no current library exists in the library list for the current thread, the QGPL library is used.

name Specify the name of the library where the object is to be moved.

Top

From ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device where storage is allocated for the library containing the object to be moved. If the library resides in an ASP that is not part of the library name space associated with the thread, this parameter must be specified to ensure the correct object is moved. If this parameter is used when *LIBL or *CURLIB is specified for the **Library (OBJ)** parameter, ASPDEV(*) is the only valid value.

*
- The ASPs that are currently part of the thread's library name space will be searched to find the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

***CURASGRP**

If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. If no ASP group is associated with the thread an error will be issued.

***SYSBAS**

The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to find the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

name Specify the name of the primary or secondary ASP device to be searched to find the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Note: To specify a specific auxiliary storage pool (ASP) device name, you must have use (*USE) authority for each ASP device in the ASP group.

Top

To ASP device (TOASPDEV)

Specifies the auxiliary storage pool (ASP) device where storage is allocated for the to-library specified for the **To library (TOLIB)** parameter. If the to-library is in an ASP that is not part of the library name space associated with the thread, this parameter must be specified to ensure the correct object is moved. If this parameter is used when *CURLIB is specified for the TOLIB parameter, either TOASPDEV(*) must be specified or TOASPDEV(*ASPDEV) must be specified and the **From ASP device (ASPDEV)** parameter must be *.

*ASPDEV

The ASP device specified for the ASPDEV parameter will be searched to find the library.

- * The ASPs that are currently part of the thread's library name space will be searched to find the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

*CURASPGRP

If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. If no ASP group is associated with the thread an error will be issued.

*SYSBAS

The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to find the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

name Specify the name of the primary or secondary ASP device to be searched to find the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Note: To specify a specific auxiliary storage pool (ASP) device name, you must have use (*USE) authority for each ASP device in the ASP group.

Top

Examples

Example 1: Moving an Object from the General Purpose Library

```
MOV OBJ(QGPL/X) OBJTYPE(*PGM) TOLIB(MY)
```

The general purpose library (QGPL) is searched for the X program (*PGM) object. The X program object is moved to the MY library. After this command is run, the X program object is no longer in the QGPL library.

Example 2: Moving an Object from a Library in the Library List

```
MOV OBJ(*LIBL/Y) OBJTYPE(*FILE) TOLIB(Z)
-or-
MOV OBJ Y *FILE Z
```

The library list (*LIBL) is searched for the Y file object. If more than one file object with the same name exists in the libraries making up the library list, the first Y file object found in the library list is moved to the Z library. After this command is run, the Y file object is no longer in the library where it was found.

Example 3: Moving an Object from a Library in an Independent Auxiliary Storage Pool (ASP) to a Library in a different ASP.

```
MOV OBJ(INVENTORY/MONTHLY) OBJTYPE(*PGM)  
      TOLIB(WINVENTORY) ASPDEV(SALES) TOASPDEV(WSALES)
```

The INVENTORY library in the SALES independent auxiliary storage pool (ASP) is searched for the MONTHLY program object. The MONTHLY program object is moved to the WINVENTORY library in the WSALES ASP. After this command is run, the MONTHLY program object is no longer in the INVENTORY library in the SALES ASP. The SALES ASP and the WSALES ASP must have been activated (by varying on the ASP device) and have a status of 'Available'.

Top

Error messages

*ESCAPE Messages

CPFA030

Object already in use.

CPFB8ED

Device description &1 not correct for operation.

CPF0601

Not allowed to do operation to file &1 in &2.

CPF0602

File &1 already in library &2.

CPF0605

Device file &1 in &2 saved with storage freed.

CPF0610

File &1 in &2 not available.

CPF0678

Operation not performed for file name &1 in &2.

CPF1763

Cannot allocate one or more libraries.

CPF2105

Object &1 in &2 type *&3 not found.

CPF2110

Library &1 not found.

CPF2112

Object &1 in &2 type *&3 already exists.

CPF2113

Cannot allocate library &1.

CPF2114

Cannot allocate object &1 in &2 type *&3.

CPF2135

Object &1 type *&3 already exists in library.

CPF2150

Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF2160
Object type *&1 not eligible for requested function.

CPF216C
TOASPDEV value not allowed with TOLIB(*CURLIB).

CPF2173
Value for ASPDEV not valid with special value for library.

CPF218C
&1 not a primary or secondary ASP.

CPF2182
Not authorized to library &1.

CPF2183
Object &1 cannot be moved into library &3.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2193
Object &1 cannot be moved into library &4.

CPF22BC
Object &1 type &3 is not program defined.

CPF2451
Message queue &1 is allocated to another job.

CPF2512
Operation not allowed for message queue &1.

CPF32CF
Distributed file error, reason code &3.

CPF32C3
Distributed file error, level ID mismatch

CPF320B
Operation was not valid for database file &1.

CPF320C
File &1 not allowed in SQL collection &2.

CPF3201
File &1 in library &2 already exists.

CPF3202
File &1 in library &2 in use.

CPF3203
Cannot allocate object for file &1 in &2.

CPF322D
Operation not done for data base file &1.

CPF3220
Cannot do operation on file &1 in &2.

CPF3224
Not authorized to perform operation on file &1.

CPF323C
QRECOVERY library could not be allocated.

CPF323D
User does not have correct authority.

CPF323F
Move or rename of file &1 in library &2 not complete.

CPF3231
Cannot move file &1 from library &2.

CPF324B
Cannot allocate dictionary for file &1.

CPF324C
Concurrent authority holder operation prevents move, rename or restore.

CPF3245
Damage to file &1 member &6 prevents operation on file &3.

CPF325D
Field CCSID values not compatible.

CPF327C
File &1 cannot be moved into library &4.

CPF327E
Alternative name for file &1 not allowed.

CPF329D
Operation not successful for file &1 in library &2.

CPF3323
Job queue &1 in &2 already exists.

CPF3330
Necessary resource not available.

CPF3353
Output queue &1 in &2 already exists.

CPF3373
Job queue &1 in &2 not moved. Job queue in use.

CPF3374
Output queue &1 in &2 not moved. Output queue in use.

CPF3467
Output queue &1 deleted and then created again.

CPF3469
Operation not allowed for output queue.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7010
Object &1 in &2 type *&3 already exists.

CPF7014
Object &1 cannot be moved to library &4.

CPF9807
One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

CPF9827

Object &1 cannot be created or moved into &2.

CPF9833

*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF9876

Protected library &2 cannot be modified.

[Top](#)

Merge Message Catalog (MRGMSGCLG)

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

Parameters
Examples
Error messages

The Merge Message Catalog (MRGMSGCLG) command merges message text from one or more source files (SRCFILE parameter) with message text in the specified message catalog (CLGFILE parameter). If the catalog specified does not already exist, it will be created using values specified for the CLGCCSID, DTAAUT, and OBJAUT parameters. If the catalog already exists, the CCSID, DTAAUT, and OBJAUT attributes of the existing message catalog will be used.

You can specify up to 300 message text source files. Message text source files are processed in the sequence specified. Each successive source file modifies the catalog. If a message number in the source file already exists in the message catalog, the new message text defined in the source file replaces the old message text in the message catalog file. If a message number in the source file does not already exist in the message catalog, the message information is added to the message catalog.

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

- GENCAT

Top

Parameters

Keyword	Description	Choices	Notes
CLGFILE	Message catalog name	<i>Path name</i>	Required, Positional 1
SRCFILE	Source file path name	Values (up to 300 repetitions): <i>Path name</i>	Required, Positional 2
CLGCCSID	Message catalog CCSID	1-65533, *SRCCCSID, *JOB	Optional
TEXT	Text 'description'	<i>Character value</i> , *BLANK	Optional
SRCCCSID	Source file CCSID	1-65533, *SRCFILE, *JOB	Optional
DTAAUT	Public authority for data	<i>Name</i> , *INDIR, *NONE, *RWX, *RX, *RW, *WX, *R, *W, *X, *EXCLUDE	Optional
OBJAUT	Public authority for object	Single values: *INDIR, *NONE, *ALL Other values (up to 4 repetitions): *OBJEXIST, *OBJMGT, *OBJALTER, *OBJREF	Optional

Top

Message catalog name (CLGFILE)

Specifies the path name of the message catalog to be changed or created. All directories in a stream file path name must exist. If no stream file exists with the specified path name, a message catalog with the specified file name is created. If the path name is in the QSYS file system, the file must exist. If a file member in the QSYS file system does not exist, it is created. Source physical files with multiple data fields are not supported.

Top

Source file path name (SRCFILE)

Specifies the path name of the source file that contains the message text to be merged into the message catalog. If the file is from the QSYS file system, then it must be a database source physical file.

Note: If the source file is not a record file, then each line in the source file must have been terminated with a newline or linefeed character when the source file was created.

Top

Message catalog CCSID (CLGCCSID)

Specifies the coded character set ID (CCSID) in which to store the message text in the message catalog. If the message catalog is a stream file, the CCSID value entered is used to set the stream file's attributes. Use the Work with Object Links (WRKLNK) command to display the CCSID of a message catalog. Use the Display File Description (DSPFD) command to determine the CCSID of a message catalog in the QSYS file system.

The possible values are:

*SRCCSID

Special value indicating that the CCSID will be determined from the value specified for the source file CCSID (SRCCSID parameter).

***JOB** Special value indicating the job CCSID is used for the catalog information. If the job CCSID is 65535, the job default CCSID is used.

coded-character-set-ID

Specify the CCSID used for the catalog information. The values 0, 65534, and 65535 are not valid.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the message catalog.

Note: Assigning text to objects is dependent on the support provided by the file system or object type used for the message catalog.

The possible values are:

*BLANK

The mode name consisting of 8 blank characters is used.

'description'

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

Top

Source file CCSID (SRCCSID)

Specifies the coded character set ID (CCSID) of the source file.

The possible values are:

***SRCFILE**

Special value indicating that the CCSID will be determined from the CCSID of the first source file (SRCFILE parameter).

***JOB** Special value indicating the job CCSID is used for the CCSID of the source file. If the job CCSID is 65535, the job default CCSID is used.

coded-character-set-ID

Specify the CCSID of the source file. The values 0, 65534, and 65535 are not valid.

Top

Public authority for data (DTAAUT)

Specifies the public authority given users for the data in the object created.

The possible values are:

***INDIR**

The authority for the object being created is determined by the directory it is being created in. If *INDIR is used for DTAAUT, it is also required for OBJAUT.

***RWX** The users are given *RWX authority to the objects. *RWX authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence, object management, object alter, and object reference authority. The user can change the object and perform basic functions on the object. *RWX authority provides object operational authority and all the data authorities.

***RX** *RX authority allows the user to perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. *RX authority provides object operational authority and read and execute authorities.

***RW** *RW authority allows the user to view the contents of an object and modify the contents of an object. *RW authority provides object operational authority and data read, add, update, and delete authorities.

***WX** *WX authority allows the user to modify the contents of an object and run a program or search a library or directory. *WX authority provides object operational authority and data add, update, delete, and execute authorities.

***R** *R authority allows the user to view the contents of an object. *R authority provides object operational authority and data read authority.

***W** *W authority allows the user to modify the contents of an object. *W authority provides object operational authority and data add, update, and delete authorities.

***X** *X authority allows the user to run a program or search a library or directory. *X authority provides object operational authority and data execute authority.

***EXCLUDE**

Exclude authority prevents the user from accessing the object. The OBJAUT value must be *NONE if this special value is used.

***NONE**

The users will not be given any of the data authorities to the objects. This value cannot be used with OBJAUT value of *NONE.

authorization-list-name

Specify the name of the authorization list used.

Top

Public authority for object (OBJAUT)

Specifies the authorities given users to the object.

The possible values are:

*INDIR

The object authority is based on the authority for the directory where this object is being created. If *INDIR is used for DTAAUT, it is also required for OBJAUT.

*NONE

None of the other object authorities (existence, management, alter, or reference) will be given to the users. If *EXCLUDE or an authorization list name is specified for the DTAAUT parameter, this value must be specified.

***ALL** All of the other object authorities (existence, management, alter, and reference) will be given to the users.

Or specify up to four (4) of the following values:

*OBJEXIST

The users will be given object existence authority to the object.

*OBJMGT

The users will be given object management authority to the object.

*OBJALTER

The users will be given object alter authority to the object.

*OBJREF

The users will be given object reference authority to the object.

Top

Examples

```
MRGMSGCLG  CLGFILE('/USDIR/USMSG.CAT')  CLGCCSID(*SRCCSID)
           SRCFILE('/QSYS.LIB/MYLIB.LIB/MSGSRC.FILE/USMSG.MBR')
           DTAAUT(*R)  TEXT('Message catalog for USA')
```

This command merges the message text from member USMSG of source physical file MSGSRC in library MYLIB in the QSYS file system with message catalog USMSG.CAT in directory USDIR. If the message catalog does not already exist, it will be created with the CCSID of the source file and data authority of *R. The text parameter describes this as a message catalog for the USA.

Top

Error messages

*ESCAPE Messages

CPF3BE3

Message catalog &1 not created or updated.

Top

Merge Message File (MRGMSGF)

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

Parameters
Examples
Error messages

The Merge Message File (MRGMSGF) command allows you to merge messages from one message file with those in another message file. Another message file may be specified to hold the messages that are replaced during the merging process. None of the message files specified are deleted by this command.

Before the command is processed, messages can be in the from-message file (FROMMSGF), in the to-message file (TOMSGF), or in both files, but not in the replaced-message file (RPLMSGF). The three possibilities result in the following when the MRGMSGF command is processed:

- When the messages are only in the FROMMSGF, they are added to the TOMSGF
- When the messages are only in the TOMSGF, they remain in the TOMSGF
- When the messages are in both the FROMMSGF and the TOMSGF, the messages in the TOMSGF are first saved into the RPLMSGF (if a replace-message file is specified); then the messages in the TOMSGF are replaced by the messages in the FROMMSGF

Restrictions: You must have use (*USE) authority for the from-message file (FROMMSGF parameter); use (*USE), add (*ADD), and delete (*DLT) authorities for the to-message file (TOMSGF parameter); and *USE and *ADD authorities for the replace-message file (RPLMSGF parameter).

Top

Parameters

Keyword	Description	Choices	Notes
FROMMSGF	From message file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: From message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
TOMSGF	To message file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: To message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
RPLMSGF	Replaced message file	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Replaced message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SELECT	Message IDs to select	Single values: *ALL Other values (up to 50 repetitions): <i>Name</i>	Optional
OMIT	Message IDs to omit	Single values: *NONE Other values (up to 50 repetitions): <i>Name</i>	Optional

Top

From message file (FROMMSGF)

Specifies the message file from which the messages are to be merged.

This is a required parameter.

Qualifier 1: From message file

name Specify the name of the message file from which the messages are to be merged.

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 from-message file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the from-message file is located.

Top

To message file (TOMSGF)

Specifies the message file into which the messages are to be merged.

This is a required parameter.

Qualifier 1: To message file

name Specify the name of the message file into which the messages are to be merged.

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 to-message file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the to-message file is located.

Top

Replaced message file (RPLMSGF)

Specifies the message file that will receive overlaid messages from the message file specified for the **To message file (TOMSGF)** parameter.

Single values

***NONE**

Overlaid messages from the TOMSGF message file are not copied to a replaced-message file.

Qualifier 1: Replaced message file

name Specify the name of the message file that will receive overlaid 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 replaced-message file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the replaced-message file is located.

Top

Message IDs to select (SELECT)

Specifies selective message IDs to merge from the message file specified for the **From message file (FROMMSGF)** parameter into the message file specified for the **To message file (TOMSGF)** parameter. Only the selected messages will be merged.

Single values

***ALL** All message IDs in the from-message file are merged with those in the to-message file.

Other values

message-identifier

Specify a list of up to 50 message IDs to be merged.

Top

Message IDs to omit (OMIT)

Specifies selective message IDs to not be merged from the message file specified for the **From message file (FROMMSGF)** parameter into the message file specified for the **To message file (TOMSGF)** parameter. All message IDs in the from-message file not included in this list are merged.

Single values

***NONE**

No message IDs are omitted from the merging process.

Other values

message-identifier

Specify a list of up to 50 message IDs not to be merged. All messages whose message IDs are not listed will be merged.

Top

Examples

Example 1: Merging Two Files

MRGMSGF FROMMSGF(A) TOMSGF(B)

Table 1. Contents of the Files Before the Merge

Message File A	Message File B
-----	-----

```

ABC1234 'text A4'   ABC1233 'text B3'
ABC1236 'text A6'   ABC1234 'text B4'
ABC1237 'text A7'   ABC1235 'text B5'
ABC1238 'text A8'   ABC1236 'text B6'

```

Below are the two message files after the MRGMSGF command is processed. Notice that messages ABC1234 and ABC1236 are in both files. When the merge occurs, the message text from file A (text A4 and A6 respectively) replaces the message text in file B (text B4 and B6 respectively).

Table 2. Contents of the Files After the Merge

Message File A	Message File B
-----	-----
ABC1234 'text A4'	ABC1233 'text B3'
ABC1236 'text A6'	ABC1234 'text A4'
ABC1237 'text A7'	ABC1235 'text B5'
ABC1238 'text A8'	ABC1236 'text A6'
	ABC1237 'text A7'
	ABC1238 'text A8'

Example 2: Merging Two Files with Replace File Option

In the example below, messages that are replaced in the to-file are saved to a separate file before being replaced.

```
MRGMSGF FROMMSGF(A) TOMSGF(B) RPLMSGF(C)
```

Table 3. Contents of the Files Before the Merge

Message File A	Message File B
-----	-----
ABC1234 'text A4'	ABC1233 'text B3'
ABC1236 'text A6'	ABC1234 'text B4'
ABC1237 'text A7'	ABC1235 'text B5'
ABC1238 'text A8'	ABC1236 'text B6'

Below are the two message files after the MRGMSGF command is processed. Notice that messages ABC1234 and ABC1236 are in both files. When the merge occurs, the text from these two messages is first moved to file C (text B4 and B6 respectively). Then, message text from file A (text A4 and A6 respectively) replaces the message text in file B (text B4 and B6 respectively).

Table 4. Contents of the Files After the Merge

Message File A	Message File B	Message File C
-----	-----	-----
ABC1234 'text A4'	ABC1233 'text B3'	ABC1234 'text B4'
ABC1236 'text A6'	ABC1234 'text A4'	ABC1236 'text B6'
ABC1237 'text A7'	ABC1235 'text B5'	
ABC1238 'text A8'	ABC1236 'text A6'	
	ABC1237 'text A7'	
	ABC1238 'text A8'	

Top

Error messages

*ESCAPE Messages

- CPF2401**
Not authorized to library &1.
- CPF2407**
Message file &1 in &2 not found.
- CPF2411**
Not authorized to message file &1 in &2.
- CPF2452**
Replaced message file must contain no messages.
- CPF2461**
Message file &1 could not be extended.
- CPF2483**
Message file currently in use.
- CPF2510**
Message file &1 in &2 logically damaged.
- CPF2519**
Error occurred while processing message ID list.
- CPF2561**
Messages were not merged.
- CPF2562**
Cannot specify the same message file more than once.
- CPF9830**
Cannot assign library &1.
- CPF9838**
User profile storage limit exceeded.

Top

Merge TCP/IP Host Table (MRGTCPHT)

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

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

The Merge TCP/IP Host Table (MRGTCPHT) command is used to merge host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the host table entries in a user specified physical file member.

A file format option is provided that allows files in either *AS400, *OPSYS, *AIX, or *NIC file formats to be merged with the local host table.

A maximum of 65 host names per IP address is allowed when host tables are merged. For example: If the local host table already has 64 host names and the physical file member to be merged has 2 additional host names, only the first host name in the physical file is merged into the final host table. Host names that exist both in the local host table and the physical file member being merged are not duplicated.

Attention: The current copy of the local host table is not saved by the MRGTCPHT command. To save the current host table, use the Copy TCP/IP Host Table (CPYTCPHT) command. Do this before issuing the MRGTCPHT command.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
FROMFILE	From file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: From file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
FROMMBR	From member	<i>Name, *FIRST, *LAST</i>	Optional, Positional 2
FILEFMT	File format	<i>*AS400, *OPSYS, *AIX, *NIC</i>	Optional
REPLACE	Replace host table	<i>*NO, *YES</i>	Optional

[Top](#)

From file (FROMFILE)

Specifies the physical file that contains the member being used for the merge operation.

This is a required parameter.

Qualifier 1: From file

name Specify the name of the physical file.

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 where the physical file is located.

Top

From member (FROMMBR)

Specifies the physical file member to be used in the merge operation.

***FIRST**

The first member of the physical file is used to merge with the host table.

***LAST**

The last member of the physical file is used to merge with the host table.

name Specify the name of the physical file member to be used.

Top

File format (FILEFMT)

Specifies the format of the physical file member to be merged with the local host table.

***AS400**

The physical file member to be merged with the local host table is *AS400 format.

Note: *AS400 can only be used if the physical file member specified is a host table from a system running Version 3 Release 1 Modification 0 (V3R1M0) through Version 5 Release 4 Modification 0 (V5R4M0) of the operating system. If you import a host table from a system running any operating system version prior to V3R1M0, specify *AIX.

If the physical file member is in *AS400 format, only host table entries containing IPv4 addresses will be merged or replaced, and any entries currently in the host table that contain IPv6 addresses will be unchanged.

***OPSYS**

The physical file member to be merged with the local host table is *OPSYS format. *OPSYS format is used for a physical file member produced by the Copy TCP/IP Host Table (CPYTCPHT) command. This value must be used if the physical file member is from a system running Version 6 Release 1 Modification 0 (V6R1M0) or later of the operating system.

***AIX** The physical file member to be merged with the local host table is *AIX format.

Note: If the physical file member is in *AIX format, only host table entries containing IPv4 addresses will be merged or replaced, and any entries currently in the host table that contain IPv6 addresses will be unchanged.

***NIC** The physical file member to be merged with the local host table is *NIC format.

Note: If the physical file member is in *NIC format, only host table entries containing IPv4 addresses will be merged or replaced, and any entries currently in the host table that contain IPv6 addresses will be unchanged.

Replace host table (REPLACE)

Specifies whether the physical file member is to be merged with or replaces the local host table.

***NO** The physical file member is merged with the local host table.

***YES** The physical file member replaces the local host table.

Examples

Example 1: Replacing Local Host Table

```
MRGTCPTH FROMFILE(HOSTFILE) REPLACE(*YES) FILEFMT(*OPSYS)
```

This command replaces the contents of the host table with the contents of the first member of physical file HOSTFILE. The first member of physical file HOSTFILE is in *OPSYS host table format.

Example 2: Merging Local Host Table

```
MRGTCPTH FROMFILE(HOSTLIB/NICFILE) FROMMBR(NEWHOSTS) FILEFMT(*NIC)
```

This command merges the current contents of the local host table with the contents of the NEWHOSTS member of physical file NICFILE in library HOSTLIB. The physical file is in *NIC format. The records are converted from *NIC format to *AS400 format by this command.

Error messages

*ESCAPE Messages

TCP1927

Records of file &1, member &2 not valid.

TCP1929

Host table not available.

TCP1934

Merge file &1, member &3, in library &2 not found.

TCP8050

*IOSYSCFG authority required to use &1.

Work with TCP/IP Network Sts (NETSTAT)

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

Parameters
Examples
Error messages

The Work with TCP/IP Network Status (WRKTC PSTS) command, also known as NETSTAT, is used to get information about the status of TCP/IP network routes, interfaces, TCP connections and UDP ports on your local system. You can also use NETSTAT to end TCP/IP connections and to start or end TCP/IP interfaces.

If IP over SNA (IPS) is enabled, NETSTAT displays information about the IP over SNA interfaces, routes, and connections. You can also use NETSTAT to end IP over SNA connections and to start or end IP over SNA interfaces.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	*SELECT, *IFC, *RTE, *CNN, *IFC6, *RTE6, *CNN6, *STATUS	Optional, Positional 1

Top

Option (OPTION)

Specifies which TCP/IP status information you want to work with.

*SELECT

Display the Work with TCP/IP Network Status menu.

*IFC Display the Work with IPv4 Interface Status list.

*RTE Display the Display IPv4 Route Information list.

*CNN Display the Work with IPv4 Connection Status list.

*IFC6 Display the Work with IPv6 Interface Status list.

*RTE6 Display the Display IPv6 Route Information list.

*CNN6

Display the Work with IPv6 Connection Status list.

*STATUS

Display the Display TCP/IP Stack Status information.

Top

Examples

Example 1: Displaying the Work with TCP/IP Network Status Menu

```
NETSTAT
-or-
NETSTAT OPTION(*SELECT)
```

Either of these commands will display the Work with TCP/IP Network Status menu.

Example 2: Using the OPTION Parameter

```
NETSTAT OPTION(*IFC)
```

This command displays the Work with IPv4 Interface Status panel.

Example 3: Using a Positional Parameter

```
NETSTAT *STATUS
```

The OPTION parameter is a positional parameter. The OPTION keyword is not required. This command starts NETSTAT, and the Display TCP/IP Stack Status panel is shown.

Top

Error messages

*ESCAPE Messages

TCP2670

Not able to complete request. TCP/IP services are not available.

TCP3844

Data for interface &3 not available.

TCP3881

Data for list not available.

TCP3882

Data not available.

TCP9999

Internal system error in program &1.

Top

Start NSLOOKUP Query (NSLOOKUP)

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

Parameters
Examples
Error messages

The Start NSLOOKUP Query (STRDNSQRY) command, or its alias NSLOOKUP, starts the Name Server Lookup tool. This tool is deprecated and is only provided for backwards compatibility. The recommended tool is now DIG which can be run using the Start DIG Query (STRDIGQRY or DIG) command. In addition, another simpler tool, HOST, is also recommended which can be run using the Start HOST Query (STRHOSTQRY or HOST) command.

NSLOOKUP is an interactive query tool that allows you to retrieve information from or test the response of a Domain Name System (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 other DNS servers.

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

NSLOOKUP retrieves information from DNS servers. It needs an active DNS server to which it can send its queries. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its current DNS server for the session:

1. The DNS server your system is configured to use
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.

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

host Look up information for host and query the current DNS server.

host dns-server

Look up information for host, but query dns-server instead of the current DNS server.

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

server dns-server

Change the current DNS server to dns-server, using the current DNS server to get the IP address of dns-server.

lserver dns-server

Change the current DNS server to dns-server, using the initial DNS server to get the IP address of 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 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 DMNNSVR parameter to specify the new DNS server as the current server for the query session.

root This interactive command is not implemented.

finger This interactive command is not implemented.

ls This interactive command is not implemented.

view This interactive command is not implemented.

help This interactive command is not implemented.

? This interactive command is not implemented.

exit Exits interactive mode.

set keyword

set keyword=value

This command is used to change state information that affects the lookups.

set all Displays the current values of the frequently used options to set. Information about the current DNS server and host is also displayed.

set class=value

Change the default query class to one of the following values:

- IN (the Internet class)
- CH (the CHAOS class)
- HS (the Hesiod class)
- ANY (wildcard - any class)

The class specifies the protocol group of the information. The initial default query class is IN.

set debug

Turn debugging mode on. A lot more information is displayed about the packet sent to the current DNS server and the resulting answer.

set nodebug

Turn debugging mode off.

set d2 Turn exhaustive (verbose) debugging mode on. A lot more information is displayed about the packet sent to the current DNS server and the resulting answer.

set nod2

Turn exhaustive (verbose) debugging mode off.

set domain=name

Set default domain to name.

set search

Append the name in the domain search list to the request.

set nosearch

Do not append the name in the domain search list to the request.

set port=X

Use port 'X' to query the current DNS server, where 'X' is a numerical port number. The default value is 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.

set querytype=X

set type=X

Sets the record type that used in the query. Replace 'X' with one of the DNS record types (A, CNAME, MX, PTR, AAAA, etc.).

set recurse

Tell the current DNS server to query other DNS servers if the current server does not have the information.

set norecurse

Tell the current DNS server to not query other DNS servers if the current server does not have the information.

set retry=X

Set the number of retries to X (where X is a numerical value). The default value for number of retries is 1.

Note: 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 the timeout interval to X seconds (where X is a numerical value). The default value is 5 seconds.

Note: Timeout 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.

set vc Use TCP for queries instead of UDP.

set novc

Use UDP for queries instead of TCP.

Restrictions:

- This command is not threadsafe when run in interactive mode.

Top

Parameters

Keyword	Description	Choices	Notes
HOSTNAME	Query name	Character value, * NONE	Optional, Positional 1
DMNNSVR	Domain name server	Character value, * CFG	Optional, Positional 2

Top

Query name (HOSTNAME)

Specifies the name that you want the Domain Name System (DNS) server to supply information about. You can use either a name or a numerical IP address. For example, a name could be "mycomputer.mycompany.com" or "mycompany.com". A numerical IP address could be "64.236.1.107". Use either the name to obtain the address, or use the address to obtain the name.

*NONE

No host name is provided. If *NONE is specified or passed as the default for this parameter, an interactive NSLOOKUP session will be started.

Note: This value is only valid if the STRDNSQRY command is run in an interactive job.

host-name

Specify the name to use for the DNS server query.

host-internet-address

Specify the IPv4 address in dotted-decimal notation or a colon-delimited IPv6 address of a host to use for the DNS server query.

Top

Domain name server (DMNNAMSVR)

Specifies the name or the IP address of the DNS server that NSLOOKUP will use as its current server for the query session. You can specify any DNS server to which your TCP/IP network has access. Or, if you want to test the response of a DNS server prior to designating it for use by your system, specify that server.

NSLOOKUP retrieves information from DNS servers. It needs an active DNS server to send its queries to. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to set one of the following as its current DNS server for the session:

1. DNS server your system is configured to use (*CFG), 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 current server for the session.

*CFG Use the DNS server that is currently designated for use by this system. These server internet addresses can be seen by prompting the Change TCP/IP Domain (CHGTCPDMN) command and looking at the values shown for the INTNETADR parameter.

server-domain-name

Specify the name of a DNS server.

server-internet-address

Specify the IP address of a DNS server.

Top

Examples

Example 1: Simple DNS Lookup

```
STRDNSQRY  HOSTNAME('64.236.1.107')
           DMNNAMSVR(*CFG)
```

This command starts a DNS query using the host located at IP address 64.236.1.107 and the DNS server that is currently designated for use by this system.

The output from this type of query might look like this:


```
Server:          10.0.1.100
Address:         10.0.1.100#53
Non-authoritative answer:
107.1.236.64.in-addr.arpa      name = any-ext.ns.aol.com.
Authoritative answers can be found from:
1.236.64.in-addr.arpa  nameserver = dns-02.atdn.net.
1.236.64.in-addr.arpa  nameserver = dns-01.atdn.net.
dns-02.atdn.net internet address = 205.188.157.236
dns-01.atdn.net internet address = 152.163.159.236
```

Example 2: DNS Lookup of an Internet Address

```
NSLOOKUP HOSTNAME(aol.com)
          DMNNAMSVR('64.236.1.107')
```

This command starts a DNS query to the local machine to find the the internet address for **aol.com**.

The output from this type of query might look like this:

```
Server:          64.236.1.107
Address:         64.236.1.107#53
aol.com
  origin = dns-01.ns.aol.com
  mail addr = hostmaster.aol.net
  serial = 2006080400
  refresh = 1800
  retry = 300
  expire = 604800
  minimum = 600
```

[Top](#)

Error messages

*ESCAPE Messages

DNS0013

Error processing command parameters.

DNS0065

Option 33 of i5/OS is required, but is not installed.

TCP7124

Program &1 in library &2 type *PGM ended abnormally.

[Top](#)

Run DNS Update (NSUPDATE)

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

Parameters
Examples
Error messages

The Run DNS Update (RUNDNSUPD) command, or its alias NSUPDATE, is used to submit Dynamic Updates requests to a Domain Name System (DNS) server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via RUNDNSUPD or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with RUNDNSUPD have to be in the same zone. Requests are sent to the zones master server. This is identified by the MNAME field of the zones SOA record.

This utility will be run in interactive mode if *NONE is specified for the **Batch input file (BCHFILE)** parameter. To run this utility in non-interactive mode, specify a file name for the BCHFILE parameter.

Restrictions:

- You must have execute (*X) authority to the directories in the path of the batch input file.
- You must have read (*R) authority to the batch input file.
- You must have execute (*X) authority to the directories in the path of the key file.
- You must have read (*R) authority to the key file.
- You must have execute (*X) authority to the directories in the path of the output file.
- You must have write (*W) authority to the output file if it already exists.
- You must have read, write and execute (*RWX) authority to the output file's parent directory if the output file does not already exist.

Top

Parameters

Keyword	Description	Choices	Notes
BCHFILE	Batch input file	<i>Path name</i> , *NONE	Optional, Positional 1
DEBUG	Show debug information	*NO, *YES	Optional, Positional 2
KEYNAME	Key name	<i>Character value</i> , *NONE	Optional
KEYFILE	Key file	<i>Path name</i> , *NONE	Optional
TIMEOUT	Update timeout	1-300, <u>300</u> , *NOMAX	Optional
UDPRTYITV	UDP timeout	1-65535, <u>3</u> , *CALC	Optional
UDPNBRRTY	UDP retry	0-100, <u>3</u>	Optional
PROTOCOL	Network protocol	*UDP, *TCP	Optional
TOSTMF	Output file	<i>Path name</i> , *STDOUT	Optional

Batch input file (BCHFILE)

Specifies a stream file with all update commands to be used as batch input.

Each command in the file is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates will be rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line (or the send command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats are as follows. Lines in the file beginning with a semicolon are considered comments and are ignored:

```
server {servername} [port]
local {address} [port]
zone {zonename}
class {classname}
key {name} {secret}
prereq nxdomain {domain-name}
prereq yxdomain {domain-name}
prereq nxrrset {domain-name} [class] {type}
prereq yxrrset {domain-name} [class] {type}
prereq yxrrset {domain-name} [class] {type} {data...}
update delete {domain-name} [ttl] [class] [type [data...]]
update add {domain-name} [ttl] [class] {type} {data...}
show
send
answer
quit
```

See the examples for descriptions of these commands.

*NONE

Do not use a batch file. This will enter interactive mode. To return from interactive mode, you must enter the command 'quit'.

Note: This value is only valid if the command is run in an interactive job.

path-name

Specify the path for a stream file from which input is read. You must have read (*R) authority to this file.

Show debug information (DEBUG)

Specifies whether or not to turn debugging mode on. More information is displayed about the packet sent to the server and the resulting answer when debugging mode is on.

*NO Turn off debugging messages.

***YES** Turn on debugging messages.

Top

Key name (KEYNAME)

Specifies a Transaction Signature (TSIG) key to sign the DNS queries. The only message digest algorithm currently used for TSIG is HMAC-MD5, although others may be added later. The TSIG key is a base-64 encoded string, typically generated by the Generate DNS Key (GENDNSKEY) command. The DNS server that is queried needs to be configured with the TSIG key and algorithm that is being used or the transaction will fail. See RFC 2845 for TSIG.

*NONE

Do not specify a key name.

character-value

Specify the Transaction Signature key. The format is 'name:key' where name is the key name, and key is the actual key as a base-64 encoded string. Example:

```
my-tsig-key:JNvcpxysbJ2hsd0qQ5qrQ==
```

The key name in this case is 'my-tsig-key' and the base-64 encoded key is 'JNvcpxysbJ2hsd0qQ5qrQ=='.

The DNS server being queried needs to include this key and algorithm in its configuration in order to accept this TSIG key from clients.

Top

Key file (KEYFILE)

Specifies a file containing a SIG(0) key used to authenticate Dynamic DNS update requests. In this case, the key specified is not an HMAC-MD5 key. SIG(0) uses public key cryptography. To use a SIG(0) key, the public key must be stored in a KEY record in a zone served by the name server. See RFC 3535 and RFC 2931 for SIG(0).

Note: If a value other than *NONE is specified for this parameter, do not specify a value for the KEYNAME parameter.

*NONE

Do not specify a key file.

path-name

Specify the path for a stream file containing key information. For example, '/home/myprofile/my-key-file' could be specified.

Top

Update timeout (TIMEOUT)

Specifies the maximum time an update request can take before it is aborted.

300 The default timeout is 300 seconds.

***NOMAX**

The update request timeout is disabled.

1-300 Specify a valid timeout value in seconds.

Top

UDP timeout (UDPRTYITV)

Specifies the interval, in seconds, between UDP retries.

3 A retry interval of 3 seconds is used.

*CALC

The retry interval is calculated by the utility based on the values specified for the **Update timeout (TIMEOUT)** parameter and the **UDP retry (UDPNBRRTY)** parameter.

1-65535

Specify the number of seconds to use between UDP retries.

Top

UDP retry (UDPNBRRTY)

Specifies the number of UDP retries.

3 Three UDP retries will be used.

0-100 Specify the number of UDP retries. If zero is specified, only one update request will be made.

Top

Network protocol (PROTOCOL)

Specified whether to use TCP or UDP when sending requests to the server.

*UDP Use UDP to send the query. However, TCP will be automatically selected for queries that require it, such as zone transfer (AXFR) requests.

*TCP Use TCP to send the query.

Top

Output file (TOSTMF)

Specifies the name of a stream file where all command output is written.

*STDOUT

All command output goes to the standard output device (normally the display).

path-name

Specify the path for a stream file where output should be written.

Top

Examples

The command formats and their meaning are as follows:

server {servername} [port]

Sends all dynamic update requests to the DNS server `servername`. When no server statement is provided, RUNDNSUPD will send updates to the master server of the correct zone. The MNAME field of that zone's SOA record will identify the master server for that zone. `port` is the port number on `servername` where the dynamic update requests get sent. If no port number is specified, the default DNS port number of 53 is used.

local {address} [port]

Sends all dynamic update requests using the local address. When no local statement is provided, RUNDNSUPD will send updates using an address and port chosen by the system. `port` can additionally be used to make requests come from a specific port. If no port number is specified, the system will assign one.

zone {zonename}

Specifies that all updates are to be made to the zone `zonename`. If no zone statement is provided, RUNDNSUPD will attempt to determine the correct zone to update based on the rest of the input.

class {classname}

Specify the default class. If no class is specified the default class is IN.

key {name} {secret}

Specifies that all updates are to be TSIG signed using the keyname/keysecret pair. The `key` command overrides any key specified on the command line via `-y` or `-k`.

prereq nxdomain {domain-name}

Requires that no resource record of any type exists with name `domain-name`.

prereq yxdomain {domain-name}

Requires that `domain-name` exists (has at least one resource record, of any type).

prereq nxrrset {domain-name} [class] {type}

Requires that no resource record exists of the specified type, class and `domain-name`. If class is omitted, IN (internet) is assumed.

prereq yxrrset {domain-name} [class] {type}

This requires that a resource record of the specified type, class and `domain-name` must exist. If class is omitted, IN (internet) is assumed.

prereq yxrrset {domain-name} [class] {type} {data...}

The data from each set of prerequisites of this form sharing a common type, class, and `domain-name` are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone at the given type, class, and `domain-name`. The data are written in the standard text representation of the resource records RDATA.

update delete {domain-name} [ttl] [class] {type} [data...]

Deletes any resource records named `domain-name`. If type and data is provided, only matching resource records will be removed. The internet class is assumed if class is not supplied. The `ttl` is ignored, and is only allowed for compatibility.

update add {domain-name} [ttl] [class] {type} {data...}

Adds a new resource record with the specified `ttl`, class and data.

show Displays the current message, containing all of the prerequisites and updates specified since the last send.

send Sends the current message. This is equivalent to entering a blank line.

answer

Displays the answer.

quit Exit interactive mode.

Example 1: Update Example

```

RUNDNSUPD  BCHFILE(*NONE)
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
> quit

```

This command will start the interactive mode for the update utility. The lines that have the '>' prefix are interactive commands.

Insert and delete resource records from the 'example.com' zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master DNS server for example.com.

Any A records for oldhost.example.com are deleted and an A record for newhost.example.com with IP address 172.16.1.1 is added. The newly-added record has a 1 day TTL (86400 seconds)

Example 2: Pre-requisite Example

```

RUNDNSUPD
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
> quit

```

The prerequisite condition gets the DNS server to check that there are no resource records of any type for nickname.example.com. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in RFC1034 that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC2535 to allow CNAMEs to have RRSIG, DNSKEY and NSEC records.)

Example 3: Batch File Example

```

RUNDNSUPD  BCHFILE('/home/ibmuser/my-updates')

```

This command sends the updates in the file '/home/ibmuser/my-updates' to the server defined in the files **server** command. For example, the contents of the file might look like this:

```

server myserver.i5os.ibm.com 53
zone i5os.ibm.com
class in
prereq yxdomain box1.i5os.ibm.com.
update delete box1.i5os.ibm.com. A
update add box1.i5os.ibm.com. 3600 A 10.9.9.9
prereq yxdomain box2.i5os.ibm.com.
update delete box2.i5os.ibm.com. A
update add box2.i5os.ibm.com. 3600 A 10.9.9.10

```

And the results displayed on the screen might look like this:

```

; TSIG error with server: tsig indicates error
update failed: NOTAUTH(BADKEY)

```


Or the results displayed on the screen might look like this:

```
; TSIG error with server: tsig indicates error  
update failed: REFUSED
```

The first example represent the case when the key in the file is not valid. The second example represents the case when the server is not allowing dynamic updates from you. No output or blank output only means the update was successful.

[Top](#)

Error messages

*ESCAPE Messages

DNS0013

Error processing command parameters.

DNS0065

Option 33 of i5/OS is required, but is not installed.

TCP7124

Program &1 in library &2 type *PGM ended abnormally.

[Top](#)

Open Data Base File (OPNDBF)

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

Parameters
Examples
Error messages

The Open Database File (OPNDBF) command opens a database file member. Processing of records is done later by application programs that do shared open operations.

Restrictions:

- This command is conditionally threadsafe. In multithreaded jobs, this command:
 - Is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.
 - Is not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.
 - Is not threadsafe for logical files that require a format selector program.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OPTION	Open option	<i>*INP, *OUT, *ALL</i>	Required, Positional 2
MBR	Member to be opened	<i>Name, *FIRST, *LAST</i>	Optional, Positional 3
OPNID	Open file identifier	<i>Name, *FILE</i>	Optional, Positional 4
ACCPATH	Access path to use	<i>*FILE, *ARRIVAL</i>	Optional
SEQONLY	Limit to sequential only	<i>Element list</i>	Optional
	Element 1: Sequential only	<i>*NO, *YES</i>	
	Element 2: Number of records	<i>Integer</i>	
COMMIT	Commitment control active	<i>*NO, *YES</i>	Optional
OPNSCOPE	Open scope	<i>*ACTGRPDFN, *ACTGRP, *JOB</i>	Optional
DUPKEYCHK	Duplicate key check	<i>*NO, *YES</i>	Optional
TYPE	Type of open	<i>*NORMAL, *PERM</i>	Optional

Top

File (FILE)

Specifies the file that contains the member to be opened. Overrides currently in effect are processed.

This is a required parameter.

Qualifier 1: File

name Specify the name of 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 used to locate the file. If no library is specified as the current library, QGPL is used.

name Specify the name of the library to be searched.

Top

Open option (OPTION)

Specifies the options to use to open a file. The options chosen on the first full open operation of a file are not changed on subsequent shared options.

This is a required parameter.

***INP** The file is opened only for input operations.

***OUT** The file is opened only for output operations.

***ALL** The file is opened for all operations (input, output, update, and delete).

Top

Member to be opened (MBR)

Specifies the member to open in the database file.

***FIRST**

The first member of the specified file is used.

***LAST**

The last member created in the file is opened.

member-name

Specify the name of the member to be opened.

Top

Open file identifier (OPNID)

Specifies the identifier used for naming this open operation so it can be referred to when the member is closed or positioned. This identifier must be specified on the Close File (CLOF) command, and on the Position Database File (POSDBF) command. It is not used on another Open Database File (OPNDBF) command until the file is closed, or an escape message is sent and the open operation fails.

***FILE** The file name is used for the open operation identifier.

name Specify the name used to identify this open operation.

Top

Access path to use (ACCPATH)

Specifies which access path to use for this open operation.

***FILE** The file access path is used. If the file is keyed, the keyed access path is used; otherwise, the arrival sequence path is used.

***ARRIVAL**

The arrival sequence access path is used. If the file is keyed, the keyed access path is ignored.

Top

Limit to sequential only (SEQONLY)

Specifies, for database files whose records are normally processed in sequential order, whether sequential-only processing is used on the file. This parameter also specifies the number of records transferred as a group to or from the database file if sequential-only processing is used. If an override specifying sequential only processing is in effect, it takes precedence over what is specified on this parameter.

Note: If *ALL is specified for the **Open option (OPTION)** parameter or *YES is specified for the **Commitment control active (COMMIT)** parameter, the *NO value is used for this parameter.

Element 1: Sequential only

***NO** The database file does not use sequential-only processing.

***YES** The database file uses sequential-only processing.

Element 2: Number of records

integer

The file uses sequential-only processing. This parameter value indicates the number of records the database blocks up in its internal buffer before actually accessing the data in the member. Specifying this number is not required. If this value is not specified, the database chooses a default value.

Top

Commitment control active (COMMIT)

Specifies whether this file is placed under commitment control.

Before a database file is opened under commitment control, the user must ensure that all files in the 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.

***NO** This file is not placed under commitment control.

***YES** This file is placed under commitment control.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN

The scope of the open operation is determined by the activation group of the program that called the OPNDBF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

*ACTGRP

The scope of the open data path (ODP) is the activation group. Only those shared opens from the same activation group can share this ODP. This ODP is not reclaimed until the activation group is deactivated, or until the Close File (CLOF) command closes the activation group.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Duplicate key check (DUPKEYCHK)

Specifies whether duplicate key checking is done on input and output operations opened by this command.

***NO** No duplicate key feedback is provided on input and output commands.

***YES** Duplicate key feedback is provided on input and output commands.

Top

Type of open (TYPE)

Specifies the recursion level at which the reclaim resources function (RCLRSC) is effective.

Note: This parameter is not valid when the **Open scope (OPNSCOPE)** parameter is specified.

*NORMAL

Allow the reclaim resources function to close the file if the program exits without doing a close operation.

*PERM

The file remains open until a close operation is done using the Close File (CLOF) command, or until the routing step ends. The open data path (ODP) remains in existence even if the Reclaim Resources (RCLRSC) command is used.

Top

Examples

```
OPNDBF FILE(MASTER/PAYROLL) OPTIONS(*INP)
```

This command opens the first member in the file PAYROLL for input processing. The open identifier associated with this open operation has the file name as its identifier. If the file is specified as SHARE(*YES), subsequent open operations of the file PAYROLL (such as in an application program) perform more efficiently and use the same ODP.

Error messages

*ESCAPE Messages

CPF4125

Open of member &3 file &1 in &2 failed.

CPF4174

OPNID(&4) for file &1 already exists.

CPF4175

Output only and MBR(*ALL) cannot be used together.

CPF4176

File &1 in &2 not a data base file.

CPF432A

Open not allowed under commitment control; reason code &8.

CPF4327

Commitment control resource limit exceeded.

CPF4328

Member &4 not journaled to journal &6.

CPF4329

Cannot associate journal &6 with commitment definition &9.

CPF8361

Cannot place resource under commitment control. Reason code &1.

CPF8367

Cannot perform commitment control operation.

Open Query File (OPNQRYF)

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

Parameters
Examples
Error messages

The Open Query File (OPNQRYF) command opens a file to a set of database records that satisfies a database query request. Once opened, the file looks like a database file opened using the Open Database File (OPNDBF) command, and the records in the file are accessed by high-level language programs that share the open data path (ODP). The path is closed, and all query resources are deallocated, using the Close File (CLOF) command. Additional help can be found in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This command is used to do any combination of the following database functions:

- Join records from more than one file, member, and record format. The join may be either equal or non-equal in nature.
- Calculate new field values using numeric and character operations on field values and constants.
- Group records by like values of one or more fields, and calculate aggregate functions, such as minimum field value and average field value, for each group.
- Select a subset of the available records, with selection both before and after grouping the records.
- Arrange result records by the value of one or more key fields.

Restrictions:

1. The user can use overrides to change the file, library, and member names specified for the FILE parameter. Overrides are ignored for the file and library specified for the FORMAT parameter, unless FORMAT(*FILE) is specified. Parameter values specified on an override command, other than TOFILE, MBR, LVLCHK, WAITRCD, SEQONLY, or INHWRT and SHARE, are ignored by the OPNQRYF command.
2. The OPNQRYF command does not share an existing open data path (ODP) in the job or activation group. If an existing SHARE(*YES) ODP in the job or activation group has the same file, library, and member name as the open query file open data path (ODP), the query file does not open and an escape message is sent.
3. Each subsequent shared open operation must use the same open options (such as SEQONLY) that are in effect when the OPNQRYF command is run.
4. Some system functions (such as the Display Physical File Member (DSPPFM) and Copy File (CPYF) commands) do not share an existing open data path. The OPNQRYF command cannot be used with those functions.
5. The file opened with the OPNQRYF command cannot be used in programs written in BASIC because BASIC does not share an existing open data path.
6. This command is conditionally threadsafe. In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA.
7. Users of this command must have the following authorities:
 - Execute (*EXECUTE) authority for any library that is needed to locate the files specified for the FILE and FORMAT parameters
 - Object operational (*OBJOPR) authority for any physical or logical file specified for the FILE parameter, and one or more of the following data authorities for the physical file or based-on physical file members of a logical file member:
 - Read (*READ) authority if the file is opened for input (using option *INP)

- Add (*ADD) authority if the file is opened for output (using option *OUT)
- Update (*UPD) authority if the file is opened for updates (using option *UPD)
- Delete (*DLT) authority if the file is opened for deletions (using option *DLT)
- *READ, *ADD, *UPD, and *DLT authority if the file is opened for all I/O operations (using option *ALL)
- *OBJOPR authority for any file specified for the FORMAT parameter
- Use (*USE) authority for any translate tables specified for the MAPFLD parameter (using option *USE)

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File specifications	Values (up to 32 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Member	<i>Name, *FIRST, *LAST, *ALL</i>	
	Element 3: Record format	<i>Name, *ONLY</i>	
OPTION	Open options	Single values: *ALL Other values (up to 4 repetitions): *INP , *OUT, *UPD, *DLT	Optional, Positional 2
FORMAT	Format specifications	Single values: *FILE Other values: <i>Element list</i>	Optional
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Record format	<i>Name, *ONLY</i>	
QRYSLT	Query selection expression	<i>Character value, *ALL</i>	Optional
KEYFLD	Key field specifications	Single values: *NONE , *FILE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Key field	<i>Qualified object name</i>	
	Qualifier 1: Key field	<i>Name</i>	
	Qualifier 2: File or element	<i>Name, *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32</i>	
	Element 2: Key field order	*ASCEND , *DESCEND	
	Element 3: Order by absolute value	*ABSVAL	
UNIQUEKEY	Unique key fields	1-120, *NONE , *ALL	Optional

Keyword	Description	Choices	Notes
JFLD	Join field specifications	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: From field	<i>Qualified object name</i>	
	Qualifier 1: From field	<i>Name</i>	
	Qualifier 2: File or element	<i>Name</i> , *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	
	Element 2: To field	<i>Qualified object name</i>	
	Qualifier 1: To field	<i>Name</i>	
	Qualifier 2: File or element	<i>Name</i> , *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	
	Element 3: Join operator	*EQ, *NE, *LT, *GT, *LE, *GE	
JDFTVAL	Join with default values	*NO, *YES, *ONLYDFT	Optional
JORDER	Join file order	*ANY, *FILE	Optional
GRPFLD	Grouping field names	Single values: *NONE Other values (up to 50 repetitions): <i>Qualified object name</i>	Optional
	Qualifier 1: Grouping field names	<i>Name</i>	
	Qualifier 2: File or element	<i>Name</i> , *MAPFLD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	
GRPSLT	Group selection expression	<i>Character value</i> , *ALL	Optional
MAPFLD	Mapped field specifications	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Mapped field	<i>Name</i>	
	Element 2: Field definition expression	<i>Character value</i>	
	Element 3: Mapped field type	*CALC, *BIN2, *BIN4, *FLT4, *FLT8, *DEC, *ZONED, *CHAR, *VCHAR, *HEX, *VHEX, *DATE, *TIME, *TIMESTP, *ONLY, *VONLY, *OPEN, *VOPEN, *EITHER, *VEITHER, *GRAPHIC, *VGRAPHIC	
	Element 4: Length	0-32766	
	Element 5: Decimal positions	0-63	
	Element 6: Mapped field CCSID	1-65535, *CALC, *HEX	
IGNDECERR	Ignore decimal data errors	*NO, *YES	Optional
OPNID	Open file identifier	<i>Name</i> , *FILE	Optional
SEQONLY	Limit to sequential only	Single values: *NO Other values: <i>Element list</i>	Optional
	Element 1: Sequential only	*YES	
	Element 2: Number of records	1-32767	
COMMIT	Commitment control active	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *ACTGRP, *JOB	Optional
DUPKEYCHK	Duplicate key check	*NO, *YES	Optional
ALWCPYDTA	Allow copy of data	*YES, *OPTIMIZE, *NO	Optional

Keyword	Description	Choices	Notes
OPTIMIZE	Performance optimization	Single values: *ALLIO, *MINWAIT Other values: <i>Element list</i>	Optional
	Element 1: Performance optimization	*FIRSTIO	
	Element 2: Number of records	1-2147483647	
OPTALLAP	Optimize all access paths	*NO, *YES	Optional
SRTSEQ	Sort sequence	Single values: *HEX, *JOB, *LANGIDSHR, *LANGIDUNQ Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Sort sequence	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
LANGID	Language ID	<i>Name</i> , *JOB	Optional
CCSID	Final output CCSID	1-65535, *JOB, *HEX	Optional
TYPE	Type of open	*NORMAL, *PERM	Optional

Top

File specifications (FILE)

Specifies one or more files, members, and record formats processed by the open query file. All files specified must be physical or logical database files, or Distributed Data Management (DDM) files. An SQL derived key index may not be specified. If Distributed Data Management files are used, all files they refer to must be on the same target system.

When more than one file, member, and record format is specified, the query joins field values to create a single set of records. Any file specified in the list may be a join logical file or view logical file member. More information on view logical files is in DB2 for i5/OS SQL reference topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

You can specify 32 values for this parameter.

This is a required parameter.

Element 1: File

Qualifier 1: File

name Specify the name of the file to be processed.

Qualifier 2: Library

*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.

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

Element 2: Member

*FIRST

The oldest member created is to be used.

*LAST

The newest member created is to be used.

***ALL** All members of a partition file are to be used.

name Specify the name of the database file member to be used.

Element 3: Record format

*ONLY

The only record format in the file is to be used. If the file has more than one record format, a record format name must be specified.

name Specify the name of the record format to be used. The record format must exist in the database file specified in the first element of this parameter.

Top

Open options (OPTION)

Specifies the open option used for the query file. The options chosen on the first full open of a file are not changed on subsequent shared opens. You can either specify *ALL or a value that combines *INP, *OUT, *UPD, and *DLT in a list of up to four values in any order.

Single values

***ALL** Open the file for all operations (*INP, *OUT, *UPD, *DLT).

Other values (up to 4 repetitions)

***INP** Open the file for input. *INP is the only value allowed if join processing or group processing is requested, if UNIQUEKEY processing is specified, if all the fields in the open query file record format specified for the **Format specifications (FORMAT)** parameter are for input-only use, or if a temporary file is required to run the query.

***OUT** Open the file for output. In some high-level languages, output to certain files (such as files defined as 'direct access' in the high-level language program) is done by using a combination of input and update operations. *UPD and *INP are specified, or *ALL is specified, in order to use an open query file with such a program.

***UPD** Open the file for update operations. If an input operation comes before an update, you must specify *INP when *UPD is specified.

***DLT** Open the file for delete operations. If a delete operation is preceded by an input operation, you must specify *INP when *DLT is specified.

Top

Format specifications (FORMAT)

Specifies the record format used for records available through the open query file. The simple field names in the open query file record format must represent fields that are either defined on the **Mapped field specifications (MAPFLD)** parameter or are unique across all files, members, and record formats specified on the **File specifications (FILE)** parameter. The value for any field that has the same name as a field specified on the MAPFLD parameter is determined by the mapped-field-definition on the MAPFLD parameter. The value for any field not defined on the MAPFLD parameter is determined by a mapping of the field with the same name in one of the based-on files, members, and record formats specified for the

FILE parameter. Only the name, type, length, decimal positions, keyboard shift, and usage attributes of each field specified in the record format that is identified on the **Format specifications (FORMAT)** parameter are used for the open query file. All other attributes are ignored. The attributes do not have to be the same. If they differ, the fields are mapped in a way similar to the Change Variable (CHGVAR) command.

Single values

***FILE** The record format of the first or only entry on the **File specifications (FILE)** parameter is used. *FILE is not allowed when more than one file, member, and record format are specified on the FILE parameter (requiring a join query).

Element 1: File

Qualifier 1: File

name Specify the name of a physical or logical database file, or a Distributed Data Management (DDM) file that contains the record format to be used.

Qualifier 2: Library

***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.

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

Element 2: Record format

***ONLY**

The only record format in the file is used. If no record format name is specified, *ONLY is the default. If the file has more than one record format, a record format name must be specified.

name Specify the name of the record format to be used. The record format must exist in the database file specified for the first element of this parameter.

Top

Query selection expression (QRYSLT)

Specifies the selection values used (before grouping) to determine the records that are available through the open query file.

***ALL** All records in the physical or logical files, members, and record formats specified for the **File specifications (FILE)** parameter (after join processing, if required) are selected.

'query-selection'

Specify an expression (contained in apostrophes) that describes the values used to determine which records are selected. You can specify any logical expression formed from relationships (such as *EQ and *NE) of field and constant values or functions of field and constant values. At least one field name is specified in each relationship. However, you cannot specify a field that depends on an aggregate function (either directly in its definition or indirectly by referring to a mapped field).

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified for the FILE parameter contains the field. The specified value *MAPFLD may be used to qualify the field name if the field is defined on the **Mapped field specifications (MAPFLD)** parameter.

For more information on data type compatibility, see Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Key field specifications (KEYFLD)

Specifies the name of one or more key fields that are used to arrange the query records, or specifies that the access path sequence of the first or only file, member, and record format specified for the **File specifications (FILE)** parameter is used to arrange the query records. If key field names are specified, you also indicate whether the part of the key associated with each key field is ascending or descending, and whether the records are arranged by the absolute value of a numeric key field. If the key field specified is a double-byte (DBCS) field, the data is arranged in a single-byte sequence.

Single values

*NONE

No key fields are used to arrange the query records; therefore, any arrangement is acceptable. It is even possible for the system to give query records in a different arrangement if the same query is run twice, based on such factors as the current number of records in the file members queried. *NONE allows the system more flexibility to improve the performance of processing records through the open query file.

***FILE** The query records have the same arrangement as the first file, member, and record format specified for the **File specifications (FILE)** parameter. *FILE can be specified even if the first file in the list has only an arrival sequence access path, in which case the query record arrangement matches the arrival sequence of the first file, member, and record format specified for the FILE parameter.

When KEYFLD(*FILE) is specified, and a sort sequence other than *HEX has been specified for the SRTSEQ parameter, you may receive your records in an order that does not reflect the true file order. If the file is keyed, the sort sequence is applied to the key fields of the file. If the file has a sort sequence table or an alternative collating sequence table, ordering is ignored. This allows users to indicate which fields to apply a sort sequence to without having to list all the field names. If a sort sequence is not specified for the query, the query is ordered as in releases previous to V2R3M0.

Element 1: Key field

Specify one or more field names (a maximum of 50 field names can be specified) to be used to define a keyed access path to arrange the query records. Each field name may be qualified with either a file name or number that indicates which element value in the list of files, members, and record formats specified for the **File specifications (FILE)** parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is defined on the **Mapped field specifications (MAPFLD)** parameter.

The sum of the lengths of all key fields cannot be more than 32766 bytes.

Note: The limits noted above are reduced by 2 bytes for each variable-length key field used. For instance, if three key fields are variable-length, the sum of the lengths of all key fields cannot exceed 32760 bytes, since 32766 bytes - (3 variable-length fields * 2 bytes per field) = 32760 bytes.

Qualifier 1: Key field

name Specify the name of the field to be used as a key field.

Qualifier 2: File or element

*MAPFLD

The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Element 2: Key field order

*ASCEND

The part of the key defined by the specified key field is ordered by ascending key values.

*DESCEND

The part of the key defined by the specified key field is ordered by descending key values.

Element 3: Order by absolute value

*ABSVAL

The part of the key defined by the specified key field is arranged by the absolute value of the key field. *ABSVAL is specified together with either *ASCEND or *DESCEND, but it is ignored if the key field is not numeric. If *ABSVAL is not specified, the records are arranged by the signed value of a numeric key field.

Top

Unique key fields (UNIQUEKEY)

Specifies whether the query is restricted to records with unique key values, and specifies how many of the key fields must be unique. If *ALL or a number is specified for this parameter, null values are considered equal.

*NONE

The key fields specified for the **Key field specifications (KEYFLD)** parameter are not required to be unique. All query records are available through the open query file, regardless of key value.

***ALL** All key fields specified for the KEYFLD parameter must be unique. If there are multiple query records with the same values for all of the key fields, only the first such record is available through the open query file.

1-120 Specify the number of key fields, ranging from 1 through 120, that is unique. This value must be no larger than the number of key fields determined by the KEYFLD parameter. If there are multiple query records with the same value for the specified number of consecutive key fields, only the first such record is available through the open query file.

Top

Join field specifications (JFLD)

Specifies whether the query joins records from multiple file members, and specifies how to join field values from the files, members, and record formats specified for the **File specifications (FILE)** parameter in constructing the query records.

The first file, member, and record format specified for the FILE parameter is called the join primary, and all other element list values specified for the FILE parameter are called join secondaries. This parameter specifies a list of pairs of field names, in which the first field in each pair provides a value that is used to select records in a join secondary that have the same value in the second field name of the pair.

The join from-field and to-field may be mapped fields (specified for the **Mapped field specifications (MAPFLD)** parameter), but you cannot use a field that depends on an aggregate function either directly in its definition or indirectly by referring to a mapped field.

The join from-field and to-field are not required to have identical field attributes. For more information on data type compatibility, see Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If more than one file is specified for the FILE parameter, *NO is specified for the **Join with default values (JDFTVAL)** parameter and *ANY is specified for the **Join file order (JORDER)** parameter, then the system takes information from the **Join field specifications (JFLD)** parameter and the **Query selection expression (QRYSLT)** parameter and derives the final join specifications. If you specify a file on the FILE parameter that is not referred to on the QRYSLT parameter or the JFLD parameter, all records for that file are logically joined to all other records created from the other files specified for the FILE parameter.

If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, or *FILE is specified for the JORDER parameter, the join fields must be specified for the JFLD parameter.

Up to 50 join field pairs can be specified.

Single values

*NONE

No join operation is specified. If more than one file is specified for the FILE parameter, *NO is specified for the JDFTVAL parameter, and *ANY is specified for the JORDER parameter, the system automatically finds the join fields from the QRYSLT parameter.

Element 1: From field

Specify a field name to provide the value used to select records in a join secondary file, member, and record format. The field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats, specified for the FILE parameter contains the field. The special value *MAPFLD can also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A join from-field is a simple field or a mapped field, defined on the MAPFLD parameter. If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, a join from-field depends only on fields that are contained in the join primary or in join secondaries specified for the FILE parameter ahead of the join secondary associated with the to-field of the pair.

Qualifier 1: From field

name Specify the name of the from-field.

Qualifier 2: File or element

*MAPFLD

The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Element 2: To field

Specify a field name used to select records from a join secondary file, member, and record format in constructing the query records. The field name is qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified in the FILE parameter contains the field. The special value *MAPFLD can also be used to qualify the field name if the field is defined on the MAPFLD parameter.

A join to-field is a simple field or a mapped field, defined on the MAPFLD parameter. If either *YES or *ONLYDFT is specified for the JDFTVAL parameter, a join to-field depends only on fields that are contained all in a single join secondary. If the join secondary is a join logical file, only fields contained in the primary physical file member for the join logical file are used as components of the join to-field. The sum of the lengths of all to-fields for each join secondary (after change, if the from-field and to-field attributes are not identical) cannot be more than 2000 bytes unless JDFTVAL(*NO) is specified, where there is no 2000-byte limit.

Qualifier 1: To field

name Specify the name of the to-field.

Qualifier 2: File or element

*MAPFLD

The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Element 3: Join operator

Specifies the type of join operation that is performed for the specified from-field and to-field. If *NO is specified for the JDFTVAL parameter and *ANY is specified for the JORDER parameter, or if more than one join field pair is specified, a different join operator may be specified for each pair. If *YES or *ONLYDFT is specified for the JDFTVAL parameter, or *FILE is specified for the JORDER parameter, then only one join operator may be specified regardless of the join pairs.

***EQ** An equal join operation is performed.

***GT** A greater than join operation is performed.

***LT** A less than join operation is performed.

***NE** A not equal join operation is performed.

***GE** A greater than or equal join operation is performed.

***LE** A less than or equal join operation is performed.

Top

Join with default values (JDFTVAL)

Specifies whether the query file should include join records that use default values for any of the fields from a join secondary file that does not contain a record with correct field values that satisfy the join connections specified on the **Join field specifications (JFLD)** parameter.

Join processing attempts to collect field values from the join primary and join secondaries. It does so by matching join from-field values to records in a join secondary that produce the appropriate values in the join to-field. If there are no records in a join secondary to produce the to-field values required for the pairs of join fields associated with the join secondary, this parameter specifies whether query records should be constructed using default values for all fields obtained from the join secondary.

If the **File specifications (FILE)** parameter includes any join logic files, all join logical files must be compatible with this parameter's value. If the data description specification (DDS) used to create a queried join logical file does not contain the JDFTVAL keyword, this parameter may not be used for any of the join logical files specified for the FILE parameter, and JDFTVAL(*NO) is required. If any join logical file has the JDFTVAL keyword specified for the FILE parameter, then join logical files for this open query file must be created using the JDFTVAL keyword, and *YES is required. If any files on the FILE parameter are view logical files, then *NO must be specified this parameter.

If the JDFTVAL attribute is not compatible with the attributes of the join logical files processed, you can replace the join or view logical files specified for the FILE parameter with their based-on physical file members. You can provide the correct, additional from-field and to-field pairs on the JFLD parameter in order to join records from the physical file members in any way.

If more than one file is specified for the FILE parameter, and either *YES or *ONLYDFT is specified, the system uses the join fields as specified for the JFLD parameter as the final join specification.

***NO** No default values are used to construct join query records.

***YES** Create all records for the join, including those produced both with and without using default values. No view logical files are allowed on the FILE parameter.

***ONLYDFT**

Create only the records produced by using default values in constructing the join. This option is used to include only exception records in the records available through the open query file. If *ONLYDFT is specified, no join or view logical files may be specified for the FILE parameter.

Top

Join file order (JORDER)

Specifies, for a join query, whether the join order must match the order specified for the **File specifications (FILE)** parameter. If the join order is varied, the query records are generated in a different arrangement. If the value specified for the **Join with default values (JDFTVAL)** parameter is *YES or *ONLYDFT, this parameter is ignored. The order specified for the FILE parameter is always preserved, because changing the join order can change which records are returned when join default value processing is required.

If more than one file is specified in the FILE parameter and *FILE is specified, the system uses the join fields as specified for the **Join field specifications (JFLD)** parameter as the final join specifications.

***ANY** Any join file order is allowed, and any such arrangement may be used by the system to create

the query records. It is possible for a query to return result records in a different arrangement if the same query is run twice consecutively (based on factors such as the current number of records in the files that are asked). *ANY allows the system more flexibility to improve the performance of processing records through the open query file than any other **Join file order (JORDER)** parameter value.

***FILE** The order of the file, member, and record format elements specified for the FILE parameter are preserved in the join operation.

Top

Grouping field names (GRPFLD)

Specifies the field names that are used to group query results. One query record is created for each group of records (after join processing, if required) selected by the **Query selection expression (QRYSLT)** parameter. The group is defined by the collection of records that has the same set of values for the fields specified in the record format identified on the **Format specifications (FORMAT)** parameter. All null values within a group are considered equal. If no field names are specified and group processing is required, the whole file is considered to be one group. Each query record that is created is either made available through the open query file or is discarded, depending on the selection values specified for the **Group selection expression (GRPSLT)** parameter. To ensure a sequence, you must specify the **Key field specifications (KEYFLD)** parameter.

Single values

*NONE

No fields are used to form groups. If the grouping function is required (because selection values are specified for the GRPSLT parameter, or an aggregate function is used by a field specified for the **Mapped field specifications (MAPFLD)** parameter), all records selected by the values specified for the QRYSLT parameter are handled as a single group.

Other values

Specify one or more field names (up to 50) to be used to group the query results. Each field name may be qualified with either a file name or number to indicate which element in the list of files, members, and record formats specified for the FILE parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the MAPFLD parameter.

A grouping field defined on the MAPFLD parameter cannot refer to an aggregate function in its definition (either directly, or indirectly through the use of another field specified for the MAPFLD parameter). The sum of the lengths of all grouping fields cannot exceed 2000 bytes.

Qualifier 1: Grouping field names

name Specify the name of a field to be used to group query results.

Qualifier 2: File or element

*MAPFLD

The field is defined on the MAPFLD parameter.

1-32 Specify the position of the element list value for the FILE parameter to be used. The element list value identifies the database file, file member, and record format to be used.

name Specify the name of a database file specified for the FILE parameter.

Top

Group selection expression (GRPSLT)

Specifies the selection values used after grouping to determine which records are available through the open query file.

***ALL** All records defined by the grouping function described by the **Grouping field names (GRPFLD)** parameter are selected.

'group-selection'

Specify an expression (contained in apostrophes) that describes the values used to determine which records are to be selected. Any logical expression formed from relationships (such as *EQ and *NE) of field and constant values, or functions of field and constant values, are specified. Only grouping fields (specified for the GRPFLD parameter), literals, aggregate functions (such as %AVG and %STDDEV), and mapped fields (specified for the **Mapped field specifications (MAPFLD)** parameter) that are composed of grouping fields, aggregate functions, and literals are referred to in any relationship. At least one field must be specified in each relationship.

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified for the **File specifications (FILE)** parameter contains the field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the MAPFLD parameter.

For more information on data type compatibility, see Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Mapped field specifications (MAPFLD)

Specifies the definition of query fields that are mapped or derived from other fields. MAPFLD is generally not needed if the field names specified on other parameters are simple field names that exist in only one of the file, member, and record format elements specified for the **File specifications (FILE)** parameter.

Up to 50 mapped field definitions can be specified.

Single values

***NONE**

No mapped fields are needed. All field names specified on other parameters exist in some record format specified for the FILE parameter.

Element 1: Mapped field

name Specify the simple field name used on any other parameter that must refer to this mapped field. A qualified name is **not** allowed for the first part of the parameter list element. All specified mapped-field-name values must be unique.

Element 2: Field definition expression

character-value

Specify an expression of up to 256 characters (contained in apostrophes) which defines the mapped field in terms of other fields that either exist in one of the file, member, and record format elements specified for the FILE parameter, or are defined by some other mapped field definition appearing earlier in the list. Either numeric operations or string operations are allowed, depending on the data type of the fields used in the definition.

Each field name may be qualified with either a file name or number that indicates which element in the list of files, members, and record formats specified for the FILE parameter contains the

field. The special value *MAPFLD may also be used to qualify the field name if the field is specified for the **Mapped field specifications (MAPFLD)** parameter.

Element 3: Mapped field type

Specify the field type for this mapped field, or specify *CALC to allow the system to calculate appropriate attributes (including field type) for the mapped field. *CALC is the default if no field-type value is specified.

When *CALC is used, the field attributes are determined in one of two ways. The attributes either match the field definition in the record format identified on the **Format specifications (FORMAT)** parameter, or (if the field is not in the record format on the FORMAT parameter) the attributes are calculated based on the expression specified in the mapped-field-definition for this field. If the mapped field is used in the record format identified on the FORMAT parameter, you must either use *CALC or specify attributes (field-type, field-length, and field-decimals) identical to those of the field in the record format specified for the FORMAT parameter.

The field type must be valid for the final result of the expression specified for the mapped-field-definition.

The following are the mappings that are not supported between character, DBCS-open, DBCS-either, DBCS-only, graphic, binary string, and numeric types:

- From character or numeric to DBCS-only
- From DBCS-open to DBCS-either or DBCS-only
- From DBCS-either to character, numeric, or DBCS-only
- From DBCS-only or DBCS-graphic to character or numeric
- From UCS-2 or UTF-16 to DBCS-either or DBCS-only
- From binary string to any non-binary string
- From numeric to binary string

Note: Binary string refers to both BLOB and BINCHAR data types.

For more information on mappings see Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

*CALC

Calculate appropriate field type attributes.

***BIN2** Two-byte binary field.

***BIN4** Four-byte binary field.

***FLT4** Four-byte floating-point field.

***FLT8** Eight-byte floating-point field.

***DEC** Packed decimal field.

***ZONED**

Zoned decimal field.

***CHAR**

Character field.

***VCHAR**

Variable length character field.

***HEX** Hexadecimal field.

***VHEX**
Variable length hexadecimal field.

***DATE**
Date field.

***TIME**
Time field.

***TIMESTP**
Timestamp field.

***ONLY**
DBCS-only field.

***VONLY**
Variable length DBCS-only field.

***OPEN**
DBCS-open field.

***VOPEN**
Variable length DBCS-open field.

***EITHER**
DBCS-either field.

***VEITHER**
Variable length DBCS-either field.

***GRAPHIC**
DBCS-graphic field.

***VGRAPHIC**
Variable length DBCS-graphic field.

Element 4: Length

0-32766

Specify the field length in number of digits for a numeric field, number of bytes for a character or DBCS field, or number of characters for a graphic field. A field length must be an even value for DBCS-only and DBCS-either field types. The range of valid lengths for each field type is shown Table 1. A value must not be specified if *CALC is used for the element 3 (Mapped field type).

Table 1. Figure: Table 1. Query Field Structure

Field Type	External Field Length	Default Length and Decimals
*BIN2	1-5	5 0
*BIN4	1-10	10 0
*FLT4	1-9	7 6
*FLT8	1-17	15 14
*DEC	1-63	15 5
*ZONED	1-63	15 5
*CHAR	1-32766	32
*VCHAR	0-32740	32
*HEX	1-32766	32
*VHEX	0-32740	32
*DATE	5-10	8
*TIME	4-8	7
*TIMESTP	14; 16-26	26
*ONLY	4-32766	32
*VONLY	0-32740	32
*OPEN	4-32766	32
*VOPEN	0-32740	32
*EITHER	4-32766	32
*VEITHER	0-32740	32
*GRAPHIC	1-16383	32
*VGRAPHIC	0-16370	32

Element 5: Decimal positions

0-63 Specify the number of decimal positions for a numeric field, expressed as a number of decimal digits, that is no larger than the total number of digits specified for the field length. If no value is given, the value is assumed to be zero. A value must not be specified for a binary or character field, or if *CALC is specified for element 3 (Mapped field type).

Element 6: Mapped field CCSID

***CALC**

The coded character set identifier (CCSID) value is determined by the CCSIDs of the fields or literal values that make up the MAPFLD field definition.

***HEX** A pre-defined value is used such that no translation of the field data takes place.

1-65535

Specify the CCSID to be used. To see a complete list of identifiers when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt).

Literal values in the MAPFLD definition are tagged with the job default CCSID. However, if the MAPFLD consists of only a literal value and the user specifies a field-CCSID value, the literal value will be tagged with that CCSID. This allows you to tag a literal with a CCSID other than the job's default CCSID.

Note: Normally, *HEX and *VHEX fields do not have an associated CCSID. Because of this, the data in the field is treated the same regardless of the default CCSID of the system on which the data is being used. However, if you specify a CCSID for a *HEX or *VHEX field, the CCSID overrides the hexadecimal attribute of the field (causing the field to be treated as *CHAR or *VCHAR), and the data in the field may be treated differently if it is moved to a system that has a different default CCSID.

Top

Ignore decimal data errors (IGNDECERR)

Specifies whether the system ignores decimal data errors during query processing.

***NO** The system does not ignore decimal data errors.

***YES** The system ignores decimal data errors. When errors in decimal data are encountered, the not valid sign or digits are automatically changed to valid values.

Top

Open file identifier (OPNID)

Specifies the identifier used to name the open query file so that it is referred to on the Close File (CLOF) or Position Database File (POSDBF) command when it is closed. The identifier must differ from the identifier associated with any other file previously opened with the Open Database File (OPNDBF) command or OPNQRYF command, and which is not yet closed.

***FILE** The name of the first or only file specified for the **File specifications (FILE)** parameter is used for the open identifier.

name Specify the name you want to associate with this open query file.

Top

Limit to sequential only (SEQONLY)

Specifies whether sequential-only processing is used for the file, and specifies the number of records processed as a group when read or write operations are performed to the open query file. The open query file ODP uses a different SEQONLY value than the one specified on this parameter, depending on other parameter values specified on this command. A message is sent if the SEQONLY value is changed.

Single values

***NO** The file does not use sequential-only processing.

Element 1: Sequential only

***YES** The open query file uses sequential-only processing.

Element 2: Number of records

1-32767

Specify the number of records that are processed as a group when read or write operations are performed to the open query file. If no value is specified, the system calculates the number of records to be processed as a group.

Top

Commitment control active (COMMIT)

Specifies whether this file is placed under commitment control.

Before a database file is opened under commitment control, the user must ensure that all files in the 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.

***NO** The open query file is not placed under commitment control.

***YES** The open query file is placed under commitment control.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

Note: This parameter is not valid when TYPE is also specified.

***ACTGRPDFN**

The scope of the open operation is determined by the activation group of the program that called the OPNQRYF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller. In a multithreaded job, only those opens within the same thread and within the same activation group can share this ODP.

***ACTGRP**

The scope of the open data path (ODP) is the activation group. Only those shared opens from the same activation group can share this ODP. This ODP is not reclaimed until the activation group is deactivated, or until the Close File (CLOF) command closes the activation group.

***JOB** The scope of the open operation is the job in which the open operation occurs. If the job is multi-threaded, only those opens from the same thread can share this ODP.

Top

Duplicate key check (DUPKEYCHK)

Specifies whether duplicate key checking should be done on input and output operations for the file opened by this command.

***NO** No duplicate key feedback is provided on input and output commands.

***YES** Duplicate key feedback is provided on input and output commands.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether the system is allowed to copy data from the files, members, and record formats specified for the **File specifications (FILE)** parameter. If so, the system is allowed to open the query file to the copy. The system generally tries to avoid using a copy of the data because the copy does not reflect changes made to the database after the information is copied. However, certain requests require that the data be copied in order to perform the specified query functions (such as when key fields contained in multiple based-on files for a join are specified).

***YES** The system may use a copy of data from the files, members, and record formats specified for the **File specifications (FILE)** parameter. A copy of the data is used only when it is needed to perform the requested query functions.

***OPTIMIZE**

The system uses a sort routine to order the output from the files, file members, and record formats specified for the FILE parameter. A sort routine is used only if the KEYFLD parameter is specified, and if using a sort routine would improve query performance without conflicting with other OPNQRYF options.

A sort will improve the performance of a query that returns most or all of the records in the file or files specified for the FILE parameter.

Using a sort can increase the time required for the OPNQRYF command to process. This occurs because the sort is performed and all records to be returned through the query are processed while the OPNQRYF command is active. However, because the records are already processed, the reading of the records (by using either a program or the CPYFRMQRYF command) is very fast. Therefore, the overall time to process the query is reduced.

Specifying the KEYFLD parameter for the OPNQRYF command does not ensure that the query will use an index if ALWCPYDTA(*OPTIMIZE) is specified. If a sort routine is used, the file is not opened with indexed access. If the program reading the records from the OPNQRYF command requires indexed access (random processing rather than sequential processing), ALWCPYDTA(*YES) or ALWCPYDTA(*NO) should be specified.

When a sort is used, the query file's position is not changed when a ROLLBACK statement is issued. Therefore, when a query is opened that has parameters, ROLLBACK statements that follow do not reset the queried file's position to where it was at the start of the unit of recovery.

Note: Do not specify ALWCPYDTA(*OPTIMIZE) if you require that a ROLLBACK statement reposition the query file, or if you require that the queried file be opened with indexed access.

The following items are required before a sort is valid for the OPNQRYF command:

- ALWCPYDTA(*OPTIMIZE) must be specified.
- The OPTION parameter, if specified, must be *INP.
- A value other than *FILE or *NONE must be specified on the KEYFLD parameter.
- The UNIQUEKEY parameter must not be specified, or must specify *NONE.
- The SEQONLY parameter, if specified, must be *YES.
- The DUPKEYCHK parameter must not be specified, or must specify *NO.
- The total buffer length of all fields in the file specified for the FORMAT parameter (or FILE parameter, if the FORMAT parameter is not specified) must not exceed 32700 bytes.

The query optimizer determines whether a sort is used. This decision is based on the number of records expected from the query and the options specified for the OPNQRYF statement. The following items influence the optimizer's choice of a sort:

- The OPTIMIZE parameter should specify *ALLIO or *MINWAIT. If *FIRSTIO is specified, the number of records specified should be close or equal to the number of result records expected from the query.
- The number of records in a file specified for the FILE parameter should contain a minimum of 200 records.
- The query result should contain a minimum of 200 records.

***NO** The system does not use a copy of data from the files, members, and record formats specified for the **File specifications (FILE)** parameter. If it is necessary to use a copy of the data to perform the requested query functions, the query file is not opened and an error message is issued.

Top

Performance optimization (OPTIMIZE)

Specifies what optimization goal is used by the system in deciding how to perform the selection and join processing necessary to satisfy other specifications on this command.

If the **Key field specifications (KEYFLD)** parameter or **Grouping field names (GRPFLD)** parameter require that an access path be built (when no existing access path can be shared), the access path is built completely, regardless of the value specified for this parameter. Optimization primarily affects the timing of selection processing.

Single values

*ALLIO

The system attempts to improve the total time to process the whole query, assuming that all query records are read from the file.

*MINWAIT

The system attempts to improve the query to minimize delays when reading records from the file.

Element 1: Performance optimization

*FIRSTIO

The system attempts to improve the time to open the query file and retrieve the first buffer of records from the file.

Element 2: Number of records

1-2147483647

Specify the number of records expected to be retrieved. The query optimizer will use this information to determine the proper implementation for the query.

Top

Optimize all access paths (OPTALLAP)

Specifies whether the query optimizer should consider all the access paths that exist over the files being queried when determining how to accomplish the query.

*NO Allow the query optimizer to operate normally. When determining how to start a query, the optimizer considers access paths until an internal timeout value has been exceeded. If there are a large number of access paths over the files being queried, the optimizer may time out before it has considered all the available access paths.

*YES Force the query optimizer to ignore the internal timeout value and consider all the available access paths over all the files in the query. Note that if there are a large number of access paths over the files it may take a long time to optimize the query.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence to be used for sorting and grouping selections specified for the QRYSLT or GRPSLT parameters, joins specified for the JFLD parameter, ordering specified for the KEYFLD parameter, grouping specified for the GRPFLD parameter, %MIN or %MAX built in functions, or unique key values specified for the UNIQUEKEY parameter.

Single values

*JOB The SRTSEQ value for the job is retrieved for the job.

*HEX A sort sequence table is not used, and the hexadecimal values of the characters are used to determine the sort sequence.

***LANGIDSHR**

A shared weight sort table is used.

***LANGIDUNQ**

A unique weight sort table is used.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this query.

Qualifier 2: Library

***LIBL** All libraries in the user and system portions 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 to be searched.

Top

Language ID (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved for the job.

language-ID

Specify the language identifier to be used by the job.

Top

Final output CCSID (CCSID)

Specifies the coded character set identifier (CCSID) in which data from character, DBCS-open, DBCS-either and graphic fields will be returned. Data from UTF-8, UCS-2, or UTF-16 fields will not be converted.

***JOB** Data is returned in the CCSID of the job issuing the OPNQRYF command.

***HEX** No CCSID conversion is performed before the data is returned.

1-65535

Specify a CCSID value. Data will be converted to this CCSID before it is returned.

Top

Type of open (TYPE)

Specifies the level at which the Reclaim Resources (RCLRSC) command closes the file.

Note: This parameter is ignored unless the default value is specified on the OPNSCOPE parameter and the request is from the default activation group.

***NORMAL**

The Reclaim Resources (RCLRSC) command closes the file if the program call that ran this command is ended without closing the file.

*PERM

The file remains open until the Close File (CLOF) command closes it, or until the routing step or default activation group ends. The query file remains open even if the Reclaim Resources (RCLRSC) command is run.

Top

Examples

Example 1: Selecting Specific Records

Note: Additional examples of selecting records using the OPNQRYF command can be found in the Database Programming topic in the Information Center.

```
OPNQRYF FILE(ordfile) OPTION(*ALL)
        QRYSLT('orddate=%range("840101" "841231") &
              ordamt>100')
        KEYFLD((ordamt *descend))
```

This command uses the QRYSLT parameter to select only records in the first member of file ORDFILE that have an order date in 1984 and an order amount greater than 100. Because the FORMAT parameter is omitted, the open query file has the same record format as file ORDFILE. The open query file allows all file operations (input, output, update, and delete). The KEYFLD specification is used to force the records to be arranged by descending value of order amount.

Example 2: Using the %XLATE Built-In Function

```
OPNQRYF FILE(telefile)
        QRYSLT('%xlate(usrname qsystemtbl) *ct
              "GEORGE"')
```

This command uses the %XLATE built-in function to translate the field USRNAME to uppercase, and to instruct the *CT operator to select only records that contain the value GEORGE in the field USRNAME. QSYSTEMTBL is an IBM-supplied system translation table that converts lowercase alphabetic (a through z) to uppercase (A through Z). The translation is done to ensure that the search value is recognized even if its characters appear in mixed case. The records available through the open query file have the same record format as those in file TELEFILE.

Example 3: Using the %XLATE Built-In Function

```
OPNQRYF FILE(telefile) QRYSLT('usrname *ct ''GEORGE'')
        MAPFLD((usrname
              '%xlate(telefile/username qsystemtbl)'))
```

In the previous example, the value of field USRNAME, which is returned to the high-level language (HLL) program that reads records from the open query file, is not translated to uppercase.

This example shows a way to make the uppercase version of field USRNAME available to the HLL program. This is done by defining a mapped field (MAPFLD parameter) for the translated value of field USRNAME. The field has the same field name as the field name in the open query file record format being used. The translated version of the field is used for selection (QRYSLT parameter) and is used in the open query file record format.

Example 4: Using the %SST Built-In Function

```
OPNQRYF FILE((histlib/ordfile hist1))
         OPTION(*inp *upd *dlt)
         FORMAT(ordinfo orddt1s) QRYSLT('month=7')
         MAPFLD((year '%sst(orddate 1 2)' *zoned 2)
                (month '%sst(orddate 3 2)' *zoned 2)
                (day '%sst(orddate 5 2)' *zoned 2))
```

This command uses the %SST built-in function to create a substring of the year, month, and day parts of character field ORDDATE in file ORDFILE. If the file ORDINFO has a record format, ORDDTLS, containing at least the field's YEAR, MONTH, and DAY records, these fields have input-only usage in the open query file record format because they are defined by using a built-in function (%SST) and are mappings that mix character and numeric (zoned decimal format) types. The file is opened for input, update, and delete operations, but none of the field's YEAR, MONTH, and DAY records are updated using the open query file open data path (ODP). The open query file uses only records in the HIST1 member of file ORDFILE in library HISTLIB, and the records retrieved through the file have the same format as record format ORDDTLS in file ORDINFO. Only records pertaining to the month of July are processed through the open query file (QRYSLT parameter).

Example 5: Returning the First Record of Each Set

```
OPNQRYF FILE((routelf *first locusr))
         QRYSLT('%sst(toloc 1 4) *eq "ROCH"')
         KEYFLD(fromusr fromloc tousr toloc) UNIQUEKEY(*all)
```

This command uses the KEYFLD and UNIQUEKEY parameters to return only the first record of each set of records in record format LOCUSR in the first member of file ROUTELF that have the same values for the fields FROMUSR, FROMLOC, TOUSR, and TOLOC. The query result is further restricted by selecting only records that have the value ROCH in the first four characters of field TOLOC. The records available through the open query file contain all of the fields in record format LOCUSR of file ROUTELF. If the file ROUTELF contains information about messages routed by an application, this example identifies all unique sender and receiver pairs in which the receiving location name begins with ROCH.

Example 6: Joining a File to Itself

```
OPNQRYF FILE(partpf partpf) FORMAT(partjoin)
         JFLD((1/pnbr 2/pnbr *GE))
         MAPFLD((pnm1 '1/pname')
                (pnm2 '2/pname')
                (pnbr '1/pnbr'))
```

This example illustrates how a file is joined to itself, as well as how to use the MAPFLD parameter to rename fields in the based-on files. A greater than or join is performed using field PNBR as both the join from-field and the join to-field.

The format of file PARTJOIN is assumed to contain fields named PNBR, PNM1, and PNM2. The field name PNBR is valid in the query output record format because that field is defined on the MAPFLD parameter. If the record format in file PARTJOIN contains a field named PNAME, an error occurs because the field exists in both files specified on the FILE parameter, and is not the name of a field defined on the MAPFLD parameter. The mapped field definitions are field names, so the attributes of fields PNM1 and PNM2 match the attributes of field PNAME, and the attributes of field PNBR in the open query file records match field PNBR in file PARTPF. Further, when a file is joined to itself, it is always necessary to specify a file number name for any field that is defined in the based-on file.

Example 7: Renaming Fields in Based-On Files

The same query can also be specified as follows:

```
OPNQRYF FILE(partpf partpf) FORMAT(partjoin)
        QRYSLT('1/pnbr *GE 2/pnbr')
        MAPFLD((pnm1 '1/pname')
              (pnm2 '2/pname')
              (pnbr '1/pnbr'))
```

Because more than one file is specified on the FILE parameter, and the default value is specified for the JDFTVAL and JORDER parameters, the system takes the join specifications from the values specified on the QRYSLT parameter.

Example 8: Selecting Master Records With No Detail Records

```
OPNQRYF FILE(cusmas ordfil) FORMAT(cusmas)
        JFLD((cusnbr ordfil/cusnbr)) JDFTVAL(*onlydft)
        MAPFLD((cusnbr 'cusmas/cusnbr'))
```

This command uses a join query to select only master records that have no associated detail records. The master file (CUSMAS) is joined (equal join) to the detail file (ORDFIL) by the customer number field that appears in both record formats. The customer number field name is the same in both record formats (CUSNBR). Because CUSNBR is the name of a field defined on the MAPFLD parameter, everywhere the simple field name CUSNBR is used, the mapped field version of the CUSNBR field in file CUSMAS is used (including the open query file record format, which matches the customer master file record format). The JDFTVAL parameter indicates that only records that are produced by using default values are available through the open query file. Every master record that has associated detail records (with the same value of the customer number field) is excluded, and every master record that has no associated detail records creates a result record.

Example 9: Identifying Detail Records With No Associated Master Record

```
OPNQRYF FILE(ordfil cusmas) FORMAT(ordfil)
        JFLD((cusnbr cusmas/cusnbr)) JDFTVAL(*onlydft)
        MAPFLD((cusnbr 'ordfil/cusnbr'))
```

This change of the previous example (using the same files) shows how to identify all detail records with no associated master record (in this case, all orders with an unregistered customer number):

Example 10: Calculating Basic Statistics

```
OPNQRYF FILE(scores) FORMAT(clsstats) GRPFLD(clsid)
        GRPSLT('clsavg<70 & clsmax-clsmin>30')
        MAPFLD((clsct '%count')
              (clsavg '%avg(usrscore)')
              (clsmin '%min(usrscore)')
              (clsmax '%max(usrscore)'))
```

This command uses the grouping function to calculate basic statistics for each group of records in file SCORES that have the same value in the field CLSID. Assuming file CLSSTATS has a record format containing field CLSID and all fields specified on the MAPFLD parameter, each record available through the open query file contains the value of the grouping field (CLSID) as well as the number of records

included in the group and the average, minimum, and maximum values of field USRSCORE in the group. Selection occurs after grouping, so that records are created for groups only when the average value of USRSCORE in the group is less than 70 and the difference between the maximum and minimum scores in the group is greater than 30.

Example 11: Selecting Records With a Specific Value

```
OPNQRYF FILE(ITMMAST)
        QRYSLT('itmcode=%range(32 50) & itmtype="P"')
        ALWCPYDTA(*NO) OPTIMIZE(*FIRST10)
        SEQONLY(*YES 10) TYPE(*PERM)
```

This command selects from the first member of file ITMMAST only the records that have a value of field ITMCODE in the range from 32 through 50 and also have a value of field ITMTYPE equal to the letter P. The ALWCPYDTA parameter specifies that the open query file must never use a copy of the records in file ITMMAST. The OPTIMIZE and SEQONLY parameter values cause the system to attempt to improve processing for the open query file to minimize the time needed to retrieve the first buffer of ten records. This combination of parameter values is a good choice if the file is used with a high-level language interactive inquiry program that shares the open query file open data path (ODP) and shows ten records on each display screen. The open data path (ODP) for the open query file is 'permanent' (TYPE parameter), which means that it remains open either until the file is closed by using the Close File (CLOF) command or until the routing step ends.

Example 12: Tagging a Literal with a Specific CCSID

```
OPNQRYF FILE(itmmast) QRYSLT('itmtype=pfield')
        MAPFLD((pfield 'P' *CHAR 1 *N 930))
```

This command selects from the first member of file ITMMAST only the records that have a value of field ITMTYPE equal to the letter 'P' in character set 930. The mapped field is created so that the literal 'P' can be tagged with a specific CCSID.

If a literal is not tagged with a specific CCSID, it is assigned the CCSID of the job running the query. Because of this, if an OPNQRYF statement is part of a CL program that is shared among systems with differing CCSIDs (in different countries, perhaps), a query that uses a literal in the selection specifications may not return the same results on all systems, even though the data in the files is the same. This happens because the internal representation of the literal may be different when the CL program is run in a job with a different CCSID. This representation then may not match the same records in the file. Note that the internal representation of the data in the file does not change based on the CCSID of the current job.

Tagging the literal with a specific CCSID avoids this problem. A literal tagged with a specific CCSID keeps the same internal representation on all systems. The CCSID that is used to tag the literal should be the same as the CCSID assigned to the field against which the literal is being compared.

Example 13: Using a Nonjoin Query

```
OPNQRYF FILE((EMPLOYEE)) KEYFLD((NAME))
        ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records in the EMPLOYEE file.

Example 14: Using a Join Query

```
OPNQRYF FILE((EMPLOYEE) (MANAGEMENT)) FORMAT(EMPLOYEE)
KEYFLD((NAME)) JFLD((1/EMPID 2/MEMPID))
ALWCPYDTA(*OPTIMIZE)
```

This command returns all of the records required by the join criteria.

Example 15: Query Comparing Character and Numeric Data

```
OPNQRYF FILE((STAFF)) QRYSLT('SALARY > "18357.50"')
```

This command returns all of the records in the STAFF file where their salary is greater than 18357.50 even though SALARY is a numeric field and the literal value in the QRYSLT is character.

Top

Error messages

*ESCAPE Messages

CPF2115

Object &1 in &2 type *&3 damaged.

CPF2169

Job's sort sequence information not available.

CPF2619

Table &1 not found.

CPF3BCC

Language identifier &1 not valid.

CPF3BC6

Sort sequence &1 not valid.

CPF3BC7

CCSID &1 outside of valid range.

CPF3BC8

Conversion from CCSID &1 to CCSID &2 is not supported.

CPF3BC9

Conversion from CCSID &1 to CCSID &2 is not defined.

CPF3BDD

Sort sequence &1 not valid for UCS2 data.

CPF3FC0

Language identifier is not valid.

CPF4174

OPNID(&4) for file &1 already exists.

CPF8133

Table &4 in &9 damaged.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

- CPF9803**
Cannot allocate object &2 in library &3.
- CPF9807**
One or more libraries in library list deleted.
- CPF9808**
Cannot allocate one or more libraries on library list.
- CPF9810**
Library &1 not found.
- CPF9812**
File &1 in library &2 not found.
- CPF9813**
Record format &3 in file &1 not found.
- CPF9815**
Member &5 file &2 in library &3 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9826**
Cannot allocate file &2.
- CPF9830**
Cannot assign library &1.
- CPF9899**
Error occurred during processing of command.

***STATUS Messages**

- CPI4011**
Query running. &2 records selected, &1 processed.
- CPI4301**
Query running.
- CPI4302**
Query running. Building access path for &2 in &1.
- CPI4303**
Query running. Creating copy of file &1 in &2.
- CPI4304**
Query running. &1 records selected. Selection complete.
- CPI4305**
Query running. Sorting copy of file *N in *N.
- CPI4306**
Query running. Building access path from file &1 in &2.
- CPI4307**
Query running. Building hash table from &2 in &1.

Top

Otherwise (OTHERWISE)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

Specifies the command or group of commands (in an If or Do group) that are processed if none of the conditions on any of the When commands within a Select command group were evaluated to be true. After the command or Do group is processed, control is passed to the next command *after* the End Select command associated with this Otherwise command. If the command specified in this parameter is a DO, DOWHILE, DOUNTIL, or DOFOR command, all commands within the Do group are considered to be the command specified by the parameter.

Restrictions:

- This command is valid only within a CL program or ILE CL procedure.
- This command is valid only within a SELECT-ENDSELECT command group.
- Only one OTHERWISE may be specified in a SELECT-ENDSELECT command group.
- All WHEN commands in a SELECT-ENDSELECT command group must appear before the OTHERWISE command.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	<i>Command string</i>	Optional, Positional 1

Top

Command (CMD)

Specifies the command or commands (in a If or Do group) to be processed if no When commands had an expression that evaluated to true.

If the command specified in this parameter is a DO, DOWHILE, DOUNTIL, or DOFOR command, all of the commands specified within the Do group are considered to be part of the command specified by the parameter.

If no command is specified on the CMD parameter (a null OTHERWISE) control is passed to the next command *after* the ENDSELECT command associated with this WHEN command.

Any CL command can be specified on the CMD parameter, except the following commands:

- ELSE
- PGM, ENDPGM
- ENDDO
- MONMSG

- DCL, DCLF
- WHEN, OTHERWISE, ENDSELECT

[Top](#)

Examples

```
DCL  VAR(&NAME)  TYPE(*CHAR)  LEN(10)
:
SELECT
  WHEN  COND(&NAME *EQ *CMD)  THEN(DO)
    :    (group of CL commands)
  ENDDO
  WHEN  COND(&NAME *EQ *PGM)  THEN(DO)
    :    (group of CL commands)
  ENDDO
  OTHERWISE  CMD(CHGVAR &NAME *PGM)
ENDSELECT
```

The OTHERWISE specifies the command to run if none of conditions on any of the WHEN commands in a SELECT command group command group were matched. In this example the CHGVAR will be run when the value of &NAME is not *CMD and not *PGM.

[Top](#)

Error messages

None

[Top](#)

Override with Data Base File (OVRDBF)

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

Parameters
Examples
Error messages

The Override with Database File (OVRDBF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program *and* override certain parameters of the file being processed. Parameters overridden by this command are specified in the file description, in the program, or in other previously issued file override commands. This command applies to physical files, logical files, and distributed data management (DDM) files.

To override (replace) a file named in the program, specify the name of that file in the FILE parameter, and specify the name of the file that overrides it (the file to be processed by the program) in the TOFILE parameter. The other parameters of this command can be used to override parameter values contained in the file description of the overriding file.

To override only certain parameters of the file named in the program, instead of replacing the entire file, specify the name of the file in the FILE parameter and specify the *FILE value for the TOFILE parameter. Then use the other parameters of this command to override specific parameters of the file. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other previously issued file override commands.

Restrictions:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. In a multithreaded job, only Activation Group or Job scoped overrides will affect opens performed in a secondary thread.

Note: The override cannot be used for all commands. A list of the commands that cannot be overridden, along with more information on overriding files is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i>	Required, Positional 1
TOFILE	Overriding to data base file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
MBR	Overriding member	<i>Name</i> , *FIRST, *LAST, *ALL	Optional, Positional 3

Keyword	Description	Choices	Notes
POSITION	Starting position in file	Single values: *NONE, *START, *END Other values: <i>Element list</i>	Optional
	Element 1: Retrieve order	*RRN, *KEYB, *KEYBE, *KEY, *KEYAE, *KEYA	
	Element 2: *RRN-rcd nbr *KEY-nbr key flds	<i>Unsigned integer</i>	
	Element 3: *KEY-rec format having key	<i>Name</i>	
	Element 4: *KEY-key value	<i>Character value</i>	
RCDFMTLCK	Record format lock	Values (up to 32 repetitions): <i>Element list</i>	Optional
	Element 1: Record format	<i>Name</i>	
	Element 2: Lock state	*SHRRD, *SHRNUP, *SHRUPD, *EXCLRD, *EXCL	
FRCRATIO	Records to force a write	<i>Integer</i> , *NONE	Optional
FMTSLR	Rcd format selector program	<i>Qualified object name</i>	Optional
	Qualifier 1: Rcd format selector program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
WAITFILE	Maximum file wait time	<i>Integer</i> , *IMMED, *CLS	Optional
WAITRCD	Maximum record wait time	<i>Integer</i> , *IMMED, *NOMAX	Optional
NBRRCDS	Records retrieved at once	<i>Integer</i>	Optional
EOFPLY	EOF retry delay in sec	1-99999, *NONE	Optional
LVLCHK	Record format level check	*NO	Optional
EXPCHK	Check expiration date	*YES, *NO	Optional
INHVRT	Inhibit write	*YES, *NO	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSCOPE	Override scope	*ACTGRPDEFN, *CALLLVL, *JOB	Optional
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDEFN, *JOB	Optional
SEQONLY	Limit to sequential only	Single values: *NO Other values: <i>Element list</i>	Optional
	Element 1: Sequential only	*YES	
	Element 2: Number of records	<i>Integer</i>	
DSTDTA	Distributed Data	*BUFFERED, *PROTECTED, *CURRENT	Optional

Top

File being overridden (FILE)

Specifies the file in the using program to which this override command is applied. The specified file must be a database file when *FILE is specified in the **Overriding to data base file (TOFILE)** parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

name Specify the name of the file.

Top

Overriding to data base file (TOFILE)

Specifies the database file that is used instead of the file specified on the **File being overridden (FILE)** parameter, or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the same parameters specified in the database file, in the program, or in other previously issued OVRDBF commands.

Single values

***FILE** The database file named in the **File being overridden (FILE)** parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to data base file

name Specify the name of the database file that is used instead of the file specified in the FILE parameter.

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 database file. If no library is specified as the current library, QGPL is used.

name Specify the library where the database file is located.

Top

Overriding member (MBR)

Specifies the members used within the database file. This parameter is not valid for distributed data management (DDM) files that refer to remote systems other than the System/38 or the AS/400 system.

***FIRST**

The first member of a database file is used.

***LAST**

The last member of a database file is used.

***ALL** All members in your file are processed in order. All members are opened with the same override parameters as the first member. Overrides issued prior to the open of the first member are processed, but overrides or delete overrides issued following the open of the first member are not processed. EOFDLY, FMTSLR, INHWRT, or the POSITION parameter cannot be specified if MBR(*ALL) has been specified on a previously issued OVRDBF command that is still in effect for this file. An escape message is sent if any of the mutually exclusive parameters are specified.

name Specify the member name that overrides (at file open time) the member name specified in the using program, or in other called OVRDBF commands. If the member name is not specified, and a TOFILE parameter other than *FILE has been specified, the first member in the file is used.

Top

Starting position in file (POSITION)

Specifies the starting position for reading records from the database file. The first record to get can be at the beginning (*START) or at the end (*END) of the file, the nth record in the file (*RRN), or the record indicated by a key field value and one of the key-search values (*KEY, *KEYA, *KEYAE, *KEYB, or *KEYBE). This parameter overrides the value specified in the program, or in other called OVRDBF commands.

Note: This parameter cannot be specified if *ALL was specified previously on the **Overriding member (MBR)** parameter.

Single values

*NONE

No special positioning is required. The first input/output operation indicates the record that is read.

*START

The starting position is the first record in the file. If a read-previous record operation is specified in the program, an end-of-file condition occurs.

***END** The starting position is the last record in the file. When the next record is read, an end-of-file condition is reached. If a read previous record operation is requested, the last record of the file is read.

Element 1: Retrieve order

***RRN** The starting position is the relative record number specified for the second element of this parameter.

*KEYB

A record that precedes the record identified by the remaining search values (number of fields, record format name, and key value) is the first record read.

*KEYBE

The record identified by the search values is the first record read. If no record matches those values, the record that matches the largest previous value is selected.

***KEY** The record identified by the search values is the first record read. If a read-previous record operation is specified in the program, the preceding record is read.

*KEYAE

The record identified by the search values is the first record read. If there is no record that matches those values, the record with the next highest value is selected.

*KEYA

A record that follows the record identified by the remaining search values (number of fields, record format name, and key value) is the first record read.

Element 2: *RRN-rcd nbr *KEY-nbr key flds

relative-record-number

Specify the relative record number (its position from the beginning of the file) of the record that is read first. The value *RRN must be specified for the first element of this parameter. For example, POSITION(*RRN 480) specifies that record 480 is read next. If a read-previous record operation is requested, the 479th record in the file is read.

number-of-key-fields

Specify the number of key fields to use in the search, if *KEYB, *KEYBE, *KEY, *KEYAE, or *KEYA is specified for the first element of this parameter. The number of fields specified does not have to be the same as the actual number of fields in each key for the file. For example, if you

specify POSITION(*KEY 1 FMT1 A), the first record in the file format FMT1 that has a first key field value of A is read. If you specify a key value of zero, the search is based on all key fields. If zero is used, the key value contains the maximum key size. If it does not, no match occurs.

Element 3: *KEY-rec format having key

name Specify the name of the record format in the database file that contains the key value specified. If no record format name is specified, all record formats are searched for the first record that matches the other search values.

Element 4: *KEY-key value

character-value

Specify the first record read. The key value is specified as a character string enclosed in apostrophes for character or positive zoned decimal formats, or is specified in hexadecimal form (x'value'). You can specify up to 2000 characters in the character string.

For example, POSITION(*KEY 1 FMT2 X'123F') specifies that:

1. The system searches for a record from the record format FMT2.
2. A single key field is used in the search (even though the key value may have more key fields).
3. The record contains the hexadecimal value 123F (the hexadecimal equivalent of packed decimal value 123.0). You get this record when it is found.

The Distributed database programming topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> has more information on the effects of using the POSITION parameter with DDM files.

Top

Record format lock (RCDFMTLCK)

Specifies the lock state of the named record format while it is used by the program. The lock state indicates how the data associated with each format is locked. The following example shows the lock states that are specified for each record format and the operations allowed to other programs when the lock is in effect:

Lock State	Other Program Operations
-----	-----
*SHRRD (Shared read)	Read and update allowed
*SHRNUP (Shared read, no update)	Read allowed, update not allowed
*SHRUP (Shared update)	Read and update allowed
*EXCLRD (Exclusive allow read)	Read allowed, update not allowed
*EXCL (Exclusive no read)	Neither read nor update allowed

An explanation of each lock state is in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

For each record format, specify the record format name, followed by one lock state value. If the lock state specified for the file in an Allocate Object (ALCOBJ) command is more restrictive than the lock state specified in this parameter, this parameter is ignored. Thus, this parameter can only impose a more restrictive lock state on a record format than the lock state specified for the file.

You can specify 32 values for this parameter.

Element 1: Record format

name Specify the name of record format.

Element 2: Lock state

lock-state

Specify one of the lock state values from the above table.

Top

Records to force a write (FRCRATIO)

Specifies the number of insert, delete, or update operations that can occur on records before those records are forced into auxiliary (permanent) storage. If this physical file is being journaled, either a large number or *NONE should be used. *NONE may cause long synchronization of the journal and physical files. More information on this parameter is in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, Appendix A. More information on journal management is in the Recovering your system book, SC41-5304.

This parameter overrides the force-write ratio specified in the database file, in the program, or in other previously issued OVRDBF commands.

***NONE**

There is no force write ratio; the system determines when the records are written to auxiliary storage.

integer

Specify the number of records written the changes are forced to disk. If a physical file associated with this database file is recorded in a journal, specify a larger force-write ratio.

Top

Rcd format selector program (FMTSLR)

Specifies the record format selection program that is called when a logical file member contains more than one logical record format. The user-written selection program is called when a record is inserted into the database file and a record format name is not included in the high-level language program. More information about the use of format selector programs is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. This parameter overrides the value specified in the database file and in other previously issued OVRDBF commands.

A program specified as the format selector program cannot be created with USRPRF(*OWNER) specified in the Create CL Program (CRTCLPGM) command.

Note: This parameter cannot be specified if *ALL was specified previously on the **Overriding member (MBR)** parameter.

Qualifier 1: Rcd format selector program

name Specify the name of the selection 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

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

This parameter overrides the wait time specified in the database file, in the program, or in other previously issued OVRDBF commands.

More information on this parameter is in CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, Appendix A.

*IMMED

The program does not wait. When the file is opened, an immediate allocation of the file resources is attempted.

*CLS The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

integer

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767 seconds.

Top

Maximum record wait time (WAITRCD)

Specifies the number of seconds that a program waits for a record to be updated or deleted, or for a record read in the commitment control environment with LCKLVL(*ALL) specified. More information on record locking is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. If the record is not allocated in the specified wait time, an error message is sent to the program.

Note: This parameter overrides the record wait time specified in the database file, specified in the program, or in other previously issued OVRDBF commands. The minimum delay for DDM files is 60 seconds. This value may need to be longer than the delay specified for local database files.

*NOMAX

The program waits indefinitely for a record lock.

*IMMED

The program does not wait. An immediate lock of the record is obtained when the record is read.

integer

Specify the number of seconds that the program waits for the record lock. Valid values range from 1 through 32767 seconds.

Top

Records retrieved at once (NBRRCDS)

Specifies the number of records read from auxiliary storage as a unit and written to main storage as a unit. The amount of data actually read is equal to the number of records times the physical record length, not the logical record length.

This parameter is valid for sequential or random processing and is specified only when the data records are physically located in auxiliary storage in the sequence in which they are processed. This parameter overrides the number of records value specified in the program, or in other previously issued OVRDBF commands.

integer

Specify the number of records. Valid values range from 1 through 32767.

Top

EOF retry delay in sec (EOFDLY)

Specifies the number of seconds of delay before trying to read additional records when end of file is reached. This delay is used to allow other jobs an opportunity to add records to the file, and have the new records processed without having to start the job again. When the delay time ends, the job is made active, and data management determines whether any new records were added. If no new records were added, the job waits for another time delay without informing the application program. When a number of seconds is given, no end of file occurs on the given database file until an End Job (ENDJOB) command or forced end of data (FEOD) occurs.

Note: This parameter cannot be specified if *ALL was specified previously on the **Overriding member (MBR)** parameter.

There are several ways to end a job that is waiting for records due to an EOFDLY. They are:

- Write a record to the specified file which is recognized by the application program as a last record. The application program may then do a force end of data (FEOD) to start the end-of-file processing or close the file.
- End the job using the controlled value (ENDJOB OPTION(*CNTRLD)) with a delay time greater than the time specified on the EOFDLY time. The DELAY parameter time specified must allow for the EOFDLY time to run out, plus time to process any new records that may have been added to the file, and any end-of-file processing that is done in the user's application. The end-of-file is set by database, and a normal end-of-file condition occurs after new records are retrieved.
- End the job immediately (ENDJOB OPTION(*IMMED)).
- If the job is interactive, start a system request and end the previous request.

*NONE

Normal end-of-file processing is done.

1-99999

Specify the number of seconds that the program waits between attempts to get a record when an end of file condition occurs. No end of file is signaled until force end of data occurs, or until the job is ended with the *CNTRLD option.

Top

Record format level check (LVLCHK)

Specifies whether the level identifiers for the record formats of the database file are checked when the file is opened by a program. For this check, which is done while the member is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the database member. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO** The level identifiers are not checked when the file is opened.

Top

Check expiration date (EXPCHK)

Specifies whether the expiration date of the named member is checked. This date check is valid only on a physical file member. This parameter overrides the value specified in the program, or in other called OVRDBF commands.

***YES** The expiration date of the physical file member is checked. If the current date is later than the expiration date, an escape message is sent to the program.

***NO** The expiration date is not checked.

Top

Inhibit write (INHWRT)

Specifies whether the processed records are written, deleted, or updated in the database file. The inhibit write parameter allows you to test a program without storing the processed records in the database. This parameter overrides the INHWRT parameter in other previously issued OVRDBF commands.

Note: This parameter cannot be specified if *ALL is specified on the **Overriding member (MBR)** parameter.

***YES** Processed records are prevented from being written into the database. They are written only to an output device.

***NO** All new and changed processed records are written into the database unless the program is in debug mode with *NO specified on the **Update production files (UPDPROD)** parameter, and the file is in a production library. In that case, an escape message is sent to the program.

Top

Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of previously called file override commands.

***NO** This file is not protected from other file overrides. Its values are overridden by the effects of any file override commands that were previously called.

***YES** This file is protected from the effects of any file override commands that were previously called.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** If the member is opened more than once, the same ODP is shared with each program in the job that also specifies ***YES** on the **Share open data path (SHARE)** parameter when it opens the member. This includes several open operations in the same program.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN**

The scope of the open operation is determined by the activation group of the program that called the OVRDBF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Limit to sequential only (SEQONLY)

Specifies, for database files whose records are processed in sequential order only, whether sequential only processing is used on the file. This parameter also specifies the number of records transferred as a group to or from the database if sequential only processing is used. If a number is not specified, a default number is determined by the system. This parameter is used to improve the performance of programs that process database files in a sequential manner. This parameter overrides the value specified in the program or in other previously issued OVRDBF commands.

For files opened for *input* only in a program, the specified number of records is transferred as a group from the database to an internal data management buffer.

For files opened for *output* only in a program, a group of records is transferred to the database whenever the internal data management buffer receives the specified number of processed records from the program. For output files, sequential-only processing is valid for physical file members and for logical file members that are based on one physical file member only.

If SEQONLY(*YES) is specified, and any of the following conditions are true, the SEQONLY parameter is ignored and a message is issued.

- The program opened the member for output only and SEQONLY(*YES) is specified with the default number of records, and the member opened is either a logical member, a unique keyed physical member, or other access paths are built over the physical member.
- The program opened the member for other than input or output.
- The member opened by the program for output is based on many other members.
- The record length plus the feedback area sum exceeded 32,767 bytes.

Note: Unpredictable results occur when this parameter is used for alternate index files for DDM on a system other than an iSeries or AS/400 system.

Single values

***NO** The database file is not restricted to sequential only processing.

Element 1: Sequential only

***YES** The database file uses sequential only processing. A default value for the number of records transferred as a group is determined by the system based on how the file is used, the type of access path involved, and the file's record length:

- The default is approximately the number of records that fit in an internal buffer of 4K for:
 - All database files opened for input only
 - Physical files opened for output that are only processed in either arrival sequence or in non-unique keyed sequence and that have no logical file members based on them
- The default is 1 record for:
 - All logical files opened for output only
 - Physical files opened for output only that either have *unique* keyed sequence access paths or have at least one dependent logical file with a keyed sequence access path that does not share the access path of the keyed physical file member

Element 2: Number of records

integer

Specify the number of records transferred each time. Valid values range from 1 through 32767. The file uses sequential only processing, and you must specify a value indicating the number of records in each group transferred between the database and the internal buffer. The user must ensure that the buffer size specified is always available to the program in the storage pool in which the program is running. The file uses sequential-only processing.

While records are in the internal data management buffer, other jobs can make changes to the same records in the database, and the program performing sequential-only input processing does not see the updates. To ensure that no other updating is done to records while they are in the buffer, the Allocate Object (ALCOBJ) command can be used in the program to specify either an *EXCLRD or an *EXCL lock on the file.

If a program performs sequential-only output processing and does not handle output errors (such as duplicate keys and conversion mapping errors) that may occur when the records in the buffer are written to the database, records in the buffer after the first record in error are not written.

If the file is opened for output and the value specified in this parameter is not the same as the force write ratio specified for the file, the value used by the system is the smaller of the two; a message stating which value is changed is sent to the user.

When processing SEQONLY(*YES) for writing records into a database file, feedback information for each record (such as relative record number) is not always changed. If such feedback information is important, specify SEQONLY(*NO) or SEQONLY(*YES 1).

More information on sequence-only database files is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Distributed Data (DSTDTA)

Specifies the data retrieval method used for a distributed file. This parameter has no effect if used against a non-distributed file. Other parameters, such as SEQONLY, still affect how the data is retrieved from each system, and this parameter controls how all the data is managed when accessing a distributed file. This parameter overrides the distributed file data retrieval method selected by the system, or specified in other previously issued OVRDBF commands. More information on DSTDTA can be found in the DB2 Multisystem topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

*BUFFERED

In order to achieve the best performance, data from the remote system and the local system may be kept in a buffer until retrieved by the user.

*PROTECTED

Data can be buffered, but the file is locked to prevent updates by other jobs. This will give the same performance as *BUFFERED, but guarantees current data. While one job is using this option, other jobs will not be able to update the data in the file.

*CURRENT

Data is not buffered. This option results in fully live data, with maximum concurrency, but without optimal performance.

Top

Examples

Example 1: Overriding An Existing Member

```
OVRDBF FILE(ORDERSIN) MBR(MONDAY)
```

This command overrides the existing member with member MONDAY. With the override in effect, the member MONDAY will be processed when the file ORDERSIN is opened.

Example 2: Overriding a Share Specification

```
OVRDBF FILE(ORDERSIN) SHARE(*YES)
```

This command overrides the share specification for the file ORDERSIN. Because of this override, any subsequent opens of this file within the routing step share the ODP for the file.

Example 3: Overriding a File, Member and Lock State

```
OVRDBF FILE(INPUT) TOFILE(PAYROLL) MBR(MBR1)
      RCDFMLCK((EMPDATA *EXCL))
```

This command overrides the file, the member, and the lock state of the record format EMPDATA. The override will cause the following to occur when the file INPUT is opened:

- The file PAYROLL will be processed instead of the file INPUT.
- The member MBR1 will be processed instead of the previously specified member.
- The lock *EXCL will be placed on record format EMPDATA instead of the existing lock. (*EXCL prevents another program from using the record format while the override is in effect.)

[Top](#)

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

[Top](#)

Override with Display File (OVRDSPF)

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

Parameters
 Examples
 Error messages

The Override with Display File (OVRDSPF) command is used to: (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program *and* override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter. The OVRDSPF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRDSPF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other called file override commands.

More information on override files is in the Application Display Programming book, SC41-5715.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i>	Required, Positional 1
TOFILE	Overriding to display file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to display file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DEV	Device	Values (up to 50 repetitions): <i>Name</i> , *REQUESTER	Optional, Positional 3
CHRID	Character identifier	Single values: *DEV, *SYSVAL, *JOBCCSID, *CHRIDCTL Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	
DECfmt	Decimal format	*FILE, *JOB	Optional
SFLENDtxt	SFLEND text	*MSG, *FILE	Optional
IGCDTA	User specified DBCS data	*NO, *YES	Optional
IGCEXNCHR	DBCS extension characters	*YES, *NO	Optional
WAITFILE	Maximum file wait time	<i>Integer</i> , *IMMED, *CLS	Optional

Keyword	Description	Choices	Notes
WAITRCD	Maximum record wait time	1-32767, *NOMAX, *IMMED	Optional
LVLCHK	Record format level check	*NO	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSCOPE	Override scope	*ACTGRPDFN, *CALLLVL, *JOB	Optional
DTAQ	Data queue	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Data queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
SHARE	Share open data path	*YES, *NO	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the file in the using program to which this override command is applied. The specified file must be a display device file when *FILE is specified in the **Overriding to display file (TOFILE)** parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

name Specify the name of the file.

Top

Overriding to display file (TOFILE)

Specifies the display file that is used instead of the file specified in the **File being overridden (FILE)** parameter, or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the same parameters specified in the display device file, in the program, or in other called (OVRDSPF) commands.

Single values

***FILE** The display device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to display file

name Specify the name of the display device file that is used instead of the overridden 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 used to locate the device file. If no current library entry exists in the library list, QGPL is used.

name Specify the library where the device file is located.

Top

Device (DEV)

Specifies the names of one or more display devices that are used with the display device file. This parameter overrides the device names specified in the device file, in the program, or in other called Override with Display File (OVRDSPF) commands. The device name specified in the IBM-supplied display device file is *REQUESTER.

This parameter overrides the device names specified in the device file, in the program, or in other called OVRDSPF commands.

*REQUESTER

The display device from which the program is called is the device assigned to the file when the file is opened.

name Specify the names of one or more display devices that are used with this device file to pass data records between the users of the devices and the system. Each device name must already be known on the system (in a device description) before this device file is created. *REQUESTER can be specified as one of the names.

A maximum of 50 device names (including *REQUESTER, if it is specified) can be specified in this command, but the total number cannot exceed the number specified on the **Maximum devices (MAXDEV)** parameter when the file is opened.

Top

Character identifier (CHRID)

Specifies the character identifier (graphic character set and code page) for the file. When a display file that was created with the CHRID DDS keyword is used with a work station device, the system translates data sent to and received from the device (as necessary) to ensure that the correct characters are displayed, and the correct hexadecimal byte values are returned to the application program. More information about display file CHRID processing and the translation tables that are used to convert data sent to and received from the display are in the Application Display Programming book, SC41-5715.

Single values

*DEV D

The CHRID value specified in the device description of the work station on which the application is running is used. The *DEV D value means no translation is necessary because the file has the same character identifier as the work station. For a list of valid values, see the CHRID parameter of the Create Device Description Display (CRTDEV D SP) command.

*SYSVAL

The CHRID value specified for the system on which the application is running is used. Translation may be necessary depending on the character identifier of the work station.

*JOBCCSID

The character data is changed from the device CHRID to the CCSID (coded character set identifier) of the job on display file input, and from the CCSID of the job to the device CHRID on display file output. The character data is converted, if necessary, from the device CCSID (coded character set identifier) of the job during input, and from the CCSID of the job to the device CHRID on output.

Note: This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

*CHRIDCTL

The system checks the CHRIDCTL job definition attribute to determine whether to use *JOBCCSID or *DEV D on the CHRID command parameter for this file.

Element 1: Graphic character set

integer

Specify the graphic character set value that matches the attributes of the display device. Valid values range from 1 through 32,767.

Element 2: Code page

integer

Specify the code page value that matches the attributes of the display device. Valid values range from 1 through 32,767.

Top

Decimal format (DECFMT)

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

***FILE** Use the decimal format value stored with the file when the file was created.

***JOB** Use the decimal format value from the DECFMT job attribute when the file is opened.

Top

SFLEND text (SFLENDTXT)

Specifies where the 'More...' and 'Bottom' text is retrieved from when displaying a subfile. The 'More...' and 'Bottom' text is displayed in a subfile when the SFLEND(*MORE) DDS keyword is specified on the subfile control record.

***MSG** Use the 'More...' and 'Bottom' text retrieved from messages CPX6AB1 and CPX6AB2 which exist in the current active language of the system when the file is opened.

***FILE** Use the 'More...' and 'Bottom' text that is stored in the file during file creation. This text was retrieved from messages CPX6AB1 and CPX6AB2 which exist in the active language of the system when the file was created.

Top

User specified DBCS data (IGCDTA)

Specifies, for program-described files, whether the file processes double-byte character set (DBCS) data. Specifies, for externally described files, the DBCS attributes of the file.

For program-described files:

***NO** The file does not process double-byte character set (DBCS) data.

***YES** The file processes double-byte character set (DBCS) data.

For DDS files:

***NO** The only DBCS attributes of the file are those defined in the DDS.

***YES** DBCS attributes in addition to those defined in the DDS, include (1) putting the DDS keyword for alternative data type (IGCALTTYP) into effect, and (2) identifying DBCS attributes of fields or messages not identified in the DDS.

DBCS extension characters (IGCEXNCHR)

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***YES** The system processes DBCS extension characters.

***NO** The system does not process DBCS extension characters; it displays them as undefined characters.

Top

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

More information on this parameter is in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, Appendix A.

Note: An immediate allocation of the device by the device resource is required when an acquire operation is performed to the file.

This parameter overrides the wait time specified in the device file, in the program, or in other called OVRDSPF commands.

***IMMED**

The program does not wait. Immediate allocation of file resources is required.

***CLS** The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

integer

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767 seconds.

Top

Maximum record wait time (WAITRCD)

Specifies the number of seconds the program waits for the completion of a read-from-invited-program-devices operation to a multiple device file in an HLL program. More information on how to determine when a file is considered a multiple device file is in the appropriate HLL reference manual. The program performing the read operation waits for input from all invited devices currently acquired by the file. If a record is not returned from an invited device within the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to one device.

This parameter is also used to specify the number of seconds that a CL program waits to complete a WAIT command. If a record is not returned from any of the devices that should return a record, the CL program is sent an escape message.

This parameter overrides the wait record value specified in the device file, in the program, or in other called OVRDSPF commands.

***NOMAX**

There is no limit on the amount of time the program waits for completion of a read-from-invited-program-device operation for the file.

***IMMED**

The program does not wait. If a record is not available when the read-from-invited-devices operation is done, a notify message is sent to the program.

1-32767

Specify the number of seconds that the program waits for the completion of the read-from-invited-program-device operations.

Top

Record format level check (LVLCHK)

Specifies whether the level identifiers of the record formats in this device file are checked when the file is opened by a program. For this check, which is done when the file is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created.

Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO** The level identifiers are not checked when the file is opened.

Top

Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of previous call level file override commands.

***NO** This file is not protected from other file overrides. Its values can be overridden by the effects of previous call level file override commands.

***YES** This file is protected from the effects of previous call level file override commands.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Data queue (DTAQ)

Specifies the data queue that receives an entry from the system when a data-available event is signaled from an invited display device. The data queue need not exist when the display file is created since the name specified on this parameter is not evaluated until the file is used. More information on the data queue function is in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Single values

*NONE

No data queue is specified.

Qualifier 1: Data queue

name Specify the name of the data queue on which entries are placed.

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 is used to locate the data queue. If no library is specified as the current library, QGPL is used.

name Specify the library where the data queue is located.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

More information on shared database files is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDSPF commands.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDEFN

The scope of the open operation is determined by the activation group of the program that called the OVRDSPF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Examples

```
OVRDSPF FILE(DISPLAY75) WAITFILE(30)
```

This command overrides the file wait time value specified in the DISPLAY75 device file description, in the program, or in other called OVRDSPF commands. The program in which this command occurs waits up to 30 seconds (if necessary) to allocate the required file resources to the file named DISPLAY75.

Top

Error messages

*ESCAPE Messages

CPF1892

Function &1 not allowed.

Top

Override ICF Pgm Device Entry (OVRICFDEVE)

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

Parameters
 Examples
 Error messages

The Override with ICF Program Device Entry (OVRICFDEVE) command can be used to temporarily add the program device entry and the remote location name to the intersystem communications function (ICF) file, or to override a program device entry with the specified remote location name and attributes for an ICF file.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

More information on how overrides processing is performed is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, the Application Display Programming book, SC41-5715, and the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Parameters

Keyword	Description	Choices	Notes
PGMDEV	Program device	<i>Character value</i>	Required, Positional 1
RMTLOCNAME	Remote location	<i>Communications name</i> , *REQUESTER	Optional, Positional 2
CMNTYPE	Communication type	*ALL, *APPC, *ASYN, *BSC, *FINANCE, *INTRA, *RETAIL, *SNUF	Optional, Positional 3
DEV	Device	<i>Name</i> , *LOC	Optional
LCLLOCNAME	Local location	<i>Communications name</i> , *LOC, *NETATR	Optional
MODE	Mode	<i>Communications name</i> , *NETATR	Optional
RMTNETID	Remote network identifier	<i>Communications name</i> , *LOC, *NETATR, *NONE	Optional
FMTSLT	Format select	*PGM, *RECID, *RMTFMT	Optional
APPID	Application identifier	<i>Name</i> , *DEV, *USER	Optional
BATCH	Batch activity	*NO, *YES	Optional
HOST	Host type	*DEV, *CICS, *IMS, *IMSRTR	Optional
ENDSSNHOST	End session with host	*RSHUTD, *TERMSELF	Optional
SPCHOSTAPP	Special host application	*DEV, *NONE, *FLASH	Optional
INZSELF	Initialize self	*NO, *YES	Optional
HDRPROC	Header processing	*SYS, *USER	Optional
MSGPTC	Message protection	*YES, *NO	Optional
EMLDEV	Emulation device	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Device type	3278, 3284, 3286, 3287, 3288, 3289	
	Element 2: Data format	*UNFORMAT, *FIELD, *NOFIELD, *EXTFIELD	
CNVTYPE	Conversation type	*SYS, *USER, *SRCPGM	Optional

Keyword	Description	Choices	Notes
BLOCK	Blocking type	Single values: *DEV, *NONE, *ITB, *IRS, *NOSEP, *USER Other values: <i>Element list</i>	Optional
	Element 1: Blocking type	*SEP	
	Element 2: Record separator, if *SEP	<i>Hexadecimal value, X'1E'</i>	
RCDLEN	Record length	1-32767, *DEV	Optional
BLKLEN	Block length	1-32767, *DEV	Optional
TRNSPY	Transmit in transparent mode	*DEV, *NO, *YES	Optional
DTACPR	Data compression	*DEV, *NO, *YES	Optional
TRUNC	Truncate trailing blanks	*DEV, *NO, *YES	Optional
OVFLWDTA	Overflow data	*DISCARD, *RETAIN	Optional
GRPSEP	Group separator type	*DEV, *EOT, *DEV3740, *OFCSYS	Optional
RMTBSCSEL	Remote BSCSEL	*DEV, *NO, *YES	Optional
INLCNN	Initial connection	*CTLD, *DIAL, *ANS	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSCOPE	Override scope	*ACTGRPDFN, *CALLLVL, *JOB	Optional

Top

Program device (PGMDEV)

Specifies the program device name for an ICF file whose attributes are being overridden. The total number of devices that may be added to an ICF file is determined by the MAXPGMDEV parameter on the Create ICF File (CRTICFF) command or the Change ICF File (CHGICFF) command.

This is a required parameter.

character-value

Specify the name of the ICF program device entry with which the program communicates. This name is used in device-specific input/output operations to identify the program device and the session attributes. The program device name must be unique, although the same remote location name may be specified more than once. This allows more than one session to be at the same remote location, or to have different attribute values for each session at the same remote location. This program device name must be unique throughout the entries for the ICF file. If an override command is entered a second time for the same program device, then both (according to the override process rules) define the same program device entry.

Note: Refer to the APPC Programming book, SC41-5443 for information on how the system uses the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters to select an APPC device description.

Top

Remote location (RMTLOCNAME)

Specifies the remote location name with which your program communicates. A remote location must be specified using the Add ICF Program Device Entry (ADDICFDEVE) command or an applied program device override. If a remote location is not given, an escape message is sent when the program device is acquired.

communications-name

Specify the name of the remote location with which your program communicates. The remote location does not need to exist at the time this command is run but must exist, either configured on the system or in the advanced peer-to-peer networking (APPN) network, for this remote location at the time the program acquires the program device. A given remote location may be added many times using different program device names. When running, however, only one program device name associated with each asynchronous (ASYNC), binary synchronous communications equivalence link (BSCCEL), or system network architecture upline facility (SNUF) remote location may be acquired by the file at any one time. For each remote advanced program-to-program communication (APPC) location, more than one associated program device name may be acquired by the file at one time. For each SNUF remote location, there may be many devices. The system determines which device to use unless a device is specified on the **Device (DEV)** parameter.

***REQUESTER**

The name used to refer to the communications device through which the program was started is used. The session that is assigned when the program device is acquired is the same session on which the Program Start request was received. If the program is not started as a result of a Program Start request, the acquire of the program device fails. The target program always uses *REQUESTER as the remote location name in the ICF file to connect to the session that the source program uses to send the Program Start request.

*REQUESTER is valid only for a target communications job. If *REQUESTER is specified in any other type of job, an escape message is sent when the program device is acquired. When *REQUESTER is used in an acquire operation, the following parameters are ignored:

- DEV
- LCLLOCNAME
- MODE
- RMTNETID

Top

Communication type (CMNTYPE)

Specifies the communications type of the ICF device. This parameter is used only for prompting purposes; it is ignored when the command is run. The value specified for this parameter determines the subset of other parameters that are displayed (prompted) for the user.

***ALL** All parameters appear in the interactive prompt.

***APPC**

Only the APPC parameters appear in the interactive prompt.

***ASYNC**

Only the asynchronous parameters appear in the interactive prompt.

***BSCCEL**

Only the BSCCEL parameters appear in the interactive prompt.

***FINANCE**

Only the FINANCE parameters appear in the interactive prompt.

***RETAIL**

Only the RETAIL parameters appear in the interactive prompt.

***INTRA**

Only the INTRA parameters appear in the interactive prompt.

*SNUF

Only the SNUF parameters appear in the interactive prompt.

Top

Device (DEV)

Specifies the communications device that is used in the remote location. This parameter applies to all communications types, but should be specified only for the APPC, INTRA, and SNUF communications types. If the device is not valid for the remote location, an escape message is sent when the program device is acquired. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, DEV(*LOC) is used.

***LOC** The device associated with the remote location is used. If several devices can be associated with the remote location, the system determines which device is used.

name Specify the name of the device with which your program communicates. The device name applies to all communications types, but it should be specified only for the APPC and SNUF communications types if you want to indicate a specific device for the remote location. If the device name is not valid for the remote location, an escape message is sent when the program device is acquired.

Top

Local location (LCLLOCNAME)

Specifies the name of your location. This parameter applies to the APPC communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, LCLLOCNAME(*LOC) is used.

***LOC** The local location name associated with the remote location name is used.

*NETATR

The LCLLOCNAME value specified in the system network attributes is used.

communications-name

Specify the name of your location. The local location name is specified only in APPC if you want to indicate a specific local location name for the remote location. If the local location name is not valid for the remote location or remote location and device, an escape message is sent when the program device is acquired.

Top

Mode (MODE)

Specifies the mode name used. This parameter applies only to the APPC communications type. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, MODE(*NETATR) is used.

*NETATR

The mode in the network attributes is used.

*BLANK.

The mode name consisting of 8 blanks characters is used.

communications-name

Specify a mode name for the APPC communications device. If the mode is not valid for any combination of remote location device, local location, and remote network ID, an escape message is sent when the program device is acquired.

Top

Remote network identifier (RMTNETID)

Specifies the remote network ID that is used with the remote location. This parameter applies to the APPC communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device and this parameter is not overridden, RMTNETID(*LOC) is used.

***LOC** Any remote network ID for the remote location may be used. If several remote network IDs are associated with the remote location, the system automatically selects the remote network ID.

***NETATR**

The remote network identifier specified in the network attributes is used.

***NONE**

No remote network identifier is used.

communications-name

Specify a remote network ID for the APPC communications device.

Top

Format select (FMTSLT)

Specifies the record format selection that is used for input operations. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, and this parameter is not overridden, FMTSLT(*PGM) is used.

***PGM** The program determines record format selections. If an input (read) operation with a record format name is specified, that format is always selected. If a record format is not specified for the input operation, the default format (the first record format in the file) is always selected.

This also means that if there are any record identification keywords specified in the data description specifications (DDS) for the file, or if any remote formats are received, they are not taken into consideration when the record is selected.

***RECID**

The record identification keywords specified in the DDS for the file are used to do record selection. If there are no record identification keywords in the file, an error message is sent, the acquire operation of the program device ends, and the device is not acquired.

***RMTFMT**

The remote format names received from the sending system are used to do record selection. If the device is not an APPC device and *RMTFMT is specified, a run time error occurs at the time the program device is acquired.

Top

Application identifier (APPID)

Specifies (in characters) the VTAM identifier of the Customer Information Control System for Virtual Storage (CICS/VS) or Information Management System/Virtual Storage (IMS/VS) host subsystem sent with the sign-on message. This parameter applies to SNUF devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, APPID(*DEVVD) is used.

*DEVVD

The application identifier specified in the device description is used.

*USER

The application program can send messages or a sign-on to the host. This is valid only when using the 3270 program interface.

name Specify the application identifier that is sent with the sign-on message.

Top

Batch activity (BATCH)

Specifies, for CICS/VS and IMS/VS, whether this session is used for batch jobs. This parameter applies to SNUF, Retail, and INTRA devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BATCH(*NO) is used.

*NO Batch jobs do not occur.

*YES Batch jobs occur and SNUF does not assemble physical records into logical records. If *YES is specified, *NO must also be specified on the **Message protection (MSGPTC)** parameter.

Top

Host type (HOST)

Specifies the host or remote subsystem with which this session communicates. This parameter applies to SNUF devices only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HOST(*DEVVD) is used.

*DEVVD

The host system specified in the device description is used.

*CICS The session communicates with CICS/VS.

*IMS The session communicates with IMS/VS.

*IMSRTR

The session communicates with IMS/VS using the ready-to-receive option.

Top

End session with host (ENDSSNHOST)

Specifies how the SNA upline facility (SNUF) ends the session with the host.

*RSHUTD

SNUF sends a request for a turn off command to the host.

*TERMSELF

SNUF sends an end-session command to the host. This value may have to be used if the value *RSHUTD fails to end a session with a non-IBM host.

Special host application (SPCHOSTAPP)

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

*DEV

The special host application specified in the device description is used.

*NONE

SNUF does not customize support for special host applications.

*FLASH

SNUF customizes support for the Federal Reserve Flash application.

Initialize self (INZSELF)

Specifies whether a formatted INIT-SELF is built in place of the unformatted sign-on normally sent by SNUF to the host.

***NO** The unformatted default sign-on provided by SNUF is used.

***YES** The formatted INIT-SELF provided by SNUF is used.

Header processing (HDRPROC)

Specifies, for both Customer Information Control System for Virtual Storage (CICS/VS) and Information Management System for Virtual Storage (IMS/VS), whether received function management headers are passed to the application program. This parameter applies to the SNA upline facility (SNUF) communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, HDRPROC(*SYS) is used.

***SYS** SNUF removes function management headers before passing data to the program.

*USER

Function management headers are passed.

Message protection (MSGPTC)

Specifies, for both CICS/VS and IMS/VS, whether message protection is used for this session. This parameter applies to the SNUF communications type only. If the ADDICFDEVE command is not run for the specified program device, MSGPTC(*YES) is used.

***YES** Message protection is used. SNUF saves messages until they are responded to and tries to resynchronize if errors occur. *YES is valid only when *NO is also specified in the **Batch activity (BATCH)** parameter.

***NO** Message protection is not used.

Emulation device (EMLDEV)

Specifies that this program device entry is used to send and receive 3270 data streams. The emulation device parameter consists of an emulation device type and an emulation device data format. The emulation device data format specifies the format of the type 3270 data stream being sent or received. A 20- or 32-byte common header that contains type 3270 command and data flow information is located at the start of the I/O buffer that is sending or receiving the type 3270 data stream. This parameter applies only to SNUF communications. This parameter can be specified as a list of two values (elements) or as a single value (*NONE).

Single values

*NONE

This program device entry is not used to send and receive 3270 data streams.

Element 1: Device type

3278 The data stream is for a 3278, 3277, or 3279 display device.

3284 The data stream is for a 3284 printer device.

3286 The data stream is for a 3286 printer device.

3287 The data stream is for a 3287 printer device.

3288 The data stream is for a 3288 printer device.

3289 The data stream is for a 3289 printer device.

Element 2: Data format

*UNFORMAT

An unformatted 3270 data stream is sent or received. The user application program must translate the data stream into a display or printer image.

*FIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image that contains field definitions. The field definitions indicate the location and characteristics of the fields. *FIELD is valid only if *NO is specified on the BATCH parameter.

*NOFIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display or printer image without field definitions. *NOFIELD is valid only if *NO is specified on the BATCH parameter.

*EXTFIELD

A formatted 3270 data stream is sent or received. The formatted 3270 data stream contains a display image followed by field definitions. The field definitions indicate the location and characteristics of fields. *EXTFIELD is valid only if *NO is specified on the BATCH parameter and 3278 is specified as the emulation device type.

Top

Conversation type (CNVTYPE)

Specifies the conversation type for which the application program is designed. This parameter is valid for advanced program-to-program communications (APPC) communications types only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, CNVTYPE(*SYS) is used. More information on the APPC communications type can be found in the APPC Programming book, SC41-5443.

***SYS** The APPC mapped conversation support for the LU 6.2 architecture is used.

***USER**

The APPC basic conversation support for the LU 6.2 architecture is used.

***SRCPGM**

The target program accepts the conversation type specified by the source program.

Top

Blocking type (BLOCK)

Specifies whether the system or the user controls whether records are combined into blocks when they are sent. This parameter is for the BSCCEL communications type. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, BLOCK(*DEVVD) is used.

With this parameter, you may specify one of the following conditions of record formatting:

- No blocking or deblocking: The record format described in the DDS is the format for both the record and the block.
- User blocking or deblocking: Specify the BSCCEL controls needed to describe the record format of the system.
- System blocking with record separator characters: Specify the record separator character used by the system to determine record boundaries in the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks or deblocks records accordingly.

If you specify a parameter value other than *NONE or *USER, records are blocked as required by the system for output, and are deblocked on input.

Single values

***DEVVD**

The block option in the device description is used.

***NONE**

Blocking or deblocking is not done by the system.

***ITB**

The records are blocked or deblocked based on the location of an intermediate text block (ITB) control character. For input files, a record is delimited by locating the next intermediate text block character. An end-of-text or end-of-transmission block character is used as an intermediate text block character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an end-of-text or end-of-transmission block character.

***IRS**

The records are blocked or deblocked based on the location of an interrecord separator (IRS) character. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

***NOSEP**

No record separator character is contained in the block that is sent to or received from the device. The system blocks and deblocks the records using a fixed-length record, as specified in the DDS format specifications.

***USER**

The program gives all control characters, including record separator characters, BSCCEL framing characters, transparency characters, and so forth, necessary to send records.

Element 1: Blocking type

***SEP** The records are blocked or deblocked based on the location of a user-specified record separator character. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

Element 2: Record separator, if *SEP

X'1E' The record separator character X'1E' is used.

hexadecimal-value

Specify a record separator character that is unique and has a length of 1 byte. The record separator character may be specified as 2 hexadecimal characters, as in BLOCK(*SEP X'FD'), or the character may be specified as a single character, as in BLOCK(*SEP @).

Top

Record length (RCDLEN)

Specifies the maximum record length (in bytes) for data sent and received. This parameter applies to the SNUF and BSCSEL communications types only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RCDLEN(*DEVVD) is used.

***DEVVD**

The record length specified in the device description is used. If a record is longer than the specified record length, a run time error occurs at the time the record is sent or received.

1-32767

Specify the maximum record length to use for this device file. The value must be at least the size of the largest record sent. If a record is longer than the specified record length, a run time error occurs when the record is sent or received. Valid values range from 1 through 32767 for SNUF communication. For BSCSEL communication, the maximum record length is 8192 bytes.

Top

Block length (BLKLEN)

Specifies the maximum block length (in bytes) for data sent. This parameter applies to the BSCSEL and SNUF communications types only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, or is not overridden, this parameter defaults to *DEVVD.

***DEVVD**

The block length specified in the device description is used.

1-32767

Specify the maximum block length of records sent when this device file is used. The value must be at least the size of the largest record sent. Valid values range from 1 to 32767 for SNUF communication. For BSCSEL communication, the maximum block length is 8192 bytes.

Top

Transmit in transparent mode (TRNSPY)

Specifies whether text is sent in transparent text mode. Transparent text mode allows all 256 extended binary-coded decimal interchange code (EBCDIC) character codes to be sent. Use this feature when sending packed or binary data fields. This parameter is for the BSCSEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, TRNSPY(*DEVVD) is used.

***DEV**

The text transparency option specified in the device description is used.

***NO** Text transparency is not used.

***YES** Text transparency is used, which allows all 256 EBCDIC character codes to be sent. *YES is valid only when *NONE, *NOSEP, or *USER is specified in the **Blocking type (BLOCK)** parameter.

Top

Data compression (DTACPR)

Specifies whether blanks in BSCCEL data are compressed for output and decompressed for input. *YES cannot be specified if *YES is specified in the **Transmit in transparent mode (TRNSPY)** parameter. This parameter is for the BSCCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, DTACPR(*DEV) is used.

***DEV**

The data compression option specified in the device description is used.

***NO** No data compression or decompression occurs.

***YES** Data is compressed for output and decompressed for input.

Top

Truncate trailing blanks (TRUNC)

Specifies whether trailing blanks are removed from output records. *YES cannot be specified if *NOSEP is specified in the **Blocking type (BLOCK)** parameter. If *YES is specified and *YES is also specified in the **Data compression (DTACPR)** parameter, then truncation is ignored. This parameter is for BSCCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, DTACPR(*DEV) is used.

***DEV**

The trailing blanks specified in the device description are used.

***NO** Trailing blanks are not removed from output records.

***YES** Trailing blanks are removed from output records.

Top

Overflow data (OVRFLWDTA)

Specifies whether overflow data is discarded or kept.

***DISCARD**

Overflow data is not kept.

***RETAIN**

Overflow data is kept.

Top

Group separator type (GRPSEP)

Specifies a separator for groups of data, such as data sets and documents. This parameter is for the BSCSEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, GRPSEP(*DEV D) is used.

*DEV D

The group separator option specified in the device description is used.

***EOT** The BSCSEL control character EOT (end-of-transmission) is used as a data group separator.

*DEV3740

A null record (STXETX) is used as a data group separator.

*OFCSYS

A block sent with the BSCSEL control character ETX (end-of-information) is used as a data group separator.

Top

Remote BSCSEL (RMTBSCSEL)

Specifies whether the type of BSCSEL session is with a BSCSEL system. This parameter applies to the BSCSEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RMTBSCSEL(*DEV D) is used.

*DEV D

The option for BSCSEL specified in the device description is used.

***NO** An attribute of *NO indicates the remote system cannot recognize BSCSEL commands or messages. In most cases, *NO is used when communicating with remote systems such as a 3741 Data Entry Station, an Office System 6, a 5230 Data Collection System, or a System/38.

***YES** The remote system can recognize the BSCSEL transaction starting commands, transaction ending commands, and online messages. In most cases, *YES indicates that the remote system is either another iSeries computer, AS/400 system, a System/38, a System/36, or a System/34 with BSCSEL support.

Top

Initial connection (INLCNN)

Specifies the method of making a connection on the line for the session being acquired. This parameter is valid for the BSCSEL communications type only. If the Add ICF Program Device Entry (ADDICFDEVE) command is not run for the specified program device, or is not overridden, this parameter defaults to *CTLD.

*CTLD

The initial connection option specified in the controller description is used.

*DIAL

The local system starts the call.

***ANS** The remote system starts the call, and the local system answers the call.

Top

Secure from other overrides (SECURE)

Specifies whether this program device is protected from the effects of override commands issued in earlier programs.

- *NO** This program device override is not protected from other program device overrides. Its values can be overridden by the effects of any program device override commands issued in earlier programs.
- *YES** This program device override is protected from the effects of any program device override commands issued in earlier programs.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

***ACTGRPDFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

- *JOB** The scope of the override is the job in which the override occurs.

Top

Examples

Example 1: Overriding the Device Entry with the Record Format Selection Attributes

```
OVRICFDEVE  PGMDEV(BSCSEL2)  RMTLOCNAME(BSCNYC)
             FMTSLT(*RECID)
```

This command overrides the program device named BSCSEL2 with a corresponding remote location named BSCNYC for any ICF file associated with the job. The program device is overridden with the attributes of FMTSLT(*RECID).

Example 2: Overriding the Device Entry with the Record Format Selection and the Conversation Type Attributes

```
OVRICFDEVE  PGMDEV(APPC1)  RMTLOCNAME(*REQUESTER)
             FMTSLT(*RMTFMT)  CNVTYPE(*SYS)
```

This command overrides the program device entry named APPC1 with a remote location name of *REQUESTER. The program device entry is overridden with the FMTSLT(*RMTFMT) and CNVTYPE(*SYS) attributes.

Example 3: Overriding an Entry for Associated ICF Files

```
OVRICFDEVE  PGMDEV(JOE)  RMTLOCNAME(LU0MPLS)
```

This command overrides the program device entry named JOE with a remote location named LU0MPLS for any ICF file associated with the job.

Example 4: Specifying the Communications Device

```
OVRICFDEVE PGMDEV(APPC) RMTLOCNAME(APPCMPLS) DEV(MPLSLINE2)
```

This command overrides the program device entry named APPC with a remote location named APPCMPLS using device MPLSLINE2.

[Top](#)

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

[Top](#)

Override ICF File (OVRICFF)

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

Parameters
Examples
Error messages

The Override with Intersystem Communications Function File (OVRICFF) command overrides the file named in the program and overrides certain parameters of the file being processed. Parameters overridden by this command can be specified in the file description, in the program, or in other file override commands that are run later.

If a file named in the program is being overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file being processed) is specified in the TOFILE parameter.

This command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not being replaced but certain parameters of the file are being overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters being overridden are then specified by the other parameters of the OVRICFF command. Parameters that are not specified do not affect parameters specified in the file description, in the program, or in other override commands run later.

More information on how overrides processing is performed is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, the Application Display Programming book, SC41-5715, and the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i>	Required, Positional 1
TOFILE	Overriding to ICF file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to ICF file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
ACQPGMDEV	Acquire program device	<i>Element list</i>	Optional, Positional 3
	Element 1: Program device to acquire	<i>Character value</i> , *NONE	
MAXRCLEN	Maximum record length	<i>Integer</i> , *CALC	Optional
WAITFILE	Maximum file wait time	<i>Integer</i> , *IMMED, *CLS	Optional
WAITRCD	Maximum record wait time	1-32767, *NOMAX, *IMMED	Optional
LVLCHK	Record format level check	*NO	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional

Keyword	Description	Choices	Notes
OVRSCOPE	Override scope	*ACTGRPDFN, *CALLLVL, *JOB	Optional
DTAQ	Data queue	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Data queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the file to which this override command is applied. The specified file must be an intersystem communications function (ICF) file when *FILE is specified in the **Overriding to ICF file (TOFILE)** parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

name Specify the name of the ICF file.

Top

Overriding to ICF file (TOFILE)

Specifies the qualified name of the ICF file (up to 10 characters) that is used instead of the file specified in the **File being overridden (FILE)** parameter or, if *FILE is specified, indicates that certain attributes are overridden by parameters in this command. The parameters on this command override the other values in the ICF file or in the program.

Single values

***FILE** Some parameters of the ICF file named in the **File being overridden (FILE)** parameter are overridden by values specified in this command.

Qualifier 1: Overriding to ICF file

name Specify the name of the ICF file that is used instead of the overridden 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 used to locate the ICF file description. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the ICF file description is located.

Top

Acquire program device (ACQPGMDEV)

Specifies which program device is acquired by the file when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.

***NONE**

The file is opened without any devices acquired. All devices used with this file must be explicitly acquired before input/output can be directed to them.

character-value

Specify the name of the program device that is acquired when the file is opened. The name should be specified on the Add ICF Program Device Entry (ADDICFDEVE) command or the override ICF program device entry (OVRICFDEVE) command as a program device name before the file is opened.

Top

Maximum record length (MAXRCLEN)

Specifies the maximum record length used when the file is opened. This parameter overrides the value in the ICF file, in the program, or in the other OVRICFF commands run later.

***CALC**

The value calculated in the file is used when the file is opened.

integer

Specify the record length (in characters) that is used when the file is opened. Valid values range from 1 through 32767. If the record length is less than the calculated value in the file, the calculated value is used.

Top

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

***IMMED**

The program does not wait. Immediate allocation of file resources is required.

***CLS** The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

integer

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767.

Top

Maximum record wait time (WAITRCD)

Specifies the number of seconds the program waits for the completion of a read-from-invited-devices operation to a multiple device file in a high-level language program. Refer to the high-level language reference manual to determine when a file is treated as a multiple device file. The program performing the read operation waits for the input from all invited devices currently accessing the file. If a record is not returned from any of the invited program devices in the specified amount of time, a notify message is sent to the program. This parameter has no effect on an input operation directed to a single device.

***NOMAX**

There is no limit on the time the system waits for the completion of the operation.

***IMMED**

The program does not wait. If a record is not available when the read-from-invited-devices operation is done, a notify message is sent to the program.

1-32767

Specify the number of seconds that the program waits for the completion of the read-from-invited-program-devices operation.

Top

Record format level check (LVLCHK)

Specifies whether the level identifiers of the record formats in this device file are checked when the file is opened by a program. While the file is being opened, the system verifies the level identifiers and compares the record format identifiers of each record format used by the program with the same identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created.

Note: This command cannot override level checking from *NO to *YES.

***NO** The level identifiers of the record formats are not checked when the file is opened.

Top

Secure from other overrides (SECURE)

Specifies whether this file is protected from the effects of previous file override commands.

***NO** This file is not protected from other file overrides. Its values can be overridden by the effects of any file override commands started earlier.

***YES** This file is protected from the effects of any file override commands started earlier.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

***ACTGRPDEFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Data queue (DTAQ)

Specifies the data queue on which entries are placed. The specified data queue must have a minimum length of 80 characters. The data queue need not exist when the display file is created since the name specified for this parameter is not evaluated until the file is used.

Note: Keyed data queues are not supported for this parameter. If a keyed data queue is specified, a run-time error will occur; but because it is not required that a data queue exist at the time the command is issued, the error will not be flagged.

More information on the data queue function is in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Single values

*NONE

No data queue is specified.

Qualifier 1: Data queue

name Specify the name of the data queue on which entries are placed.

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 is used to locate the data queue. If no library is specified as the current library, QGPL is used.

name Specify the library where the data queue is located.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDEFN

The scope of the open operation is determined by the activation group of the program that called the OVRICFF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Examples

```
OVRICFF FILE(ICFHIST) TOFILE(PRSNNL/ICFCURT)
```

This command overrides the file named ICFHIST to the ICF file named ICFCURT in library PRSNNL.

Top

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

Top

Override Message File (OVRMSGF)

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

Parameters
Examples
Error messages

The Override with Message File (OVRMSGF) command overrides a message file used in a program. The overriding message file is used (specified in the TOMSGF parameter) whenever a message is sent or retrieved and the overridden message file is specified.

The overriding message file need not contain all the messages that the overridden file contains. When a message is received or retrieved and the message identifier cannot be found in the overriding message file, the overridden message file is searched for the identifier. Overriding message files can be overridden, resulting in a chain of overrides. This chain of overrides provides a list of message files that are searched when a message is received or retrieved. Up to 30 message file overrides can be specified in a program.

Restrictions:

- In a multithreaded job, this command may only be issued from the initial thread.
- In a multithreaded job, this command will only affect message file references performed in the initial thread. Message file references performed in secondary threads will be unaffected.

More information on overriding files is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, the Application Display Programming book, SC41-5715, and the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
MSGF	Message file being overridden	<i>Name</i>	Required, Positional 1
TOMSGF	Overriding to message file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Overriding to message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SECURE	Secure from other overrides	<i>*NO, *YES</i>	Optional

Top

Message file being overridden (MSGF)

Specifies the message file being used by the program to which this override command is applied.

This is a required parameter.

name Specify the name of the message file.

Top

Overriding to message file (TOMSGF)

Specifies the message file that is used instead of the message file specified in the **Message file being overridden (MSGF)** parameter; or, if the names are the same, specifies that the value specified in the **Secure from other overrides (SECURE)** parameter is used for the message file.

This is a required parameter.

Qualifier 1: Overriding to message file

name Specify the name of the message file that is used instead of of the overridden message 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 used to locate the message file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the message file is located.

Top

Secure from other overrides (SECURE)

Specifies whether this file is secured from the effects of message file override commands used in earlier calls. If this parameter is not specified, processing occurs as if *NO had been specified.

***NO** This message file is not protected from other file overrides. Its values can be overridden by the effects of any message file overrides used in earlier calls.

***YES** This message file is protected from the effects of any message file overrides used in earlier calls.

Top

Examples

```
OVRMSGF MSGF(WSUSRMSG) TOMSGF(ORDENTMSGD)
```

This override command causes the defaults for messages stored in ORDENTMSGD to be used instead of defaults stored in WSUSRMSG (which contains messages designed for work station users). As a result of this command, the messages received by the order entry users are tailored to their own environment.

Top

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

Override with Printer File (OVRPRTF)

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

Parameters
Examples
Error messages

The Override with Printer File (OVRPRTF) command is used to (1) override (replace) the file named in the program, (2) override certain parameters of a file that are used by the program, or (3) override the file named in the program and override certain parameters of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other file override commands that run in the following command.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file (the file processed) is specified in the TOFILE parameter. The OVRPRTF command also specifies parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRPRTF command. Parameters not specified do not affect parameters specified in the file description, in the program, or in other file override commands run later.

Restrictions:

- In a multithreaded job, this command may only be issued from the initial thread.
- In a multithreaded job, only Activation Group or Job scoped overrides will affect opens performed in a secondary thread.

More information on overriding files is in the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i> , *PRTF	Required, Positional 1
TOFILE	Overriding to printer file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to printer file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DEV	Device	<i>Element list</i>	Optional, Positional 3
	Element 1: Printer	<i>Name</i> , *SYSVAL, *JOB	
DEVTYPE	Printer device type	*SCS, *IPDS, *USERASCII, *AFPDS, *LINE, *AFPDSLIN	Optional

Keyword	Description	Choices	Notes
PAGESIZE	Page size	<i>Element list</i>	Optional
	Element 1: Page length	0.001-255.0	
	Element 2: Page width	0.001-378.0	
	Element 3: Measurement method	*ROWCOL , *UOM	
LPI	Lines per inch	3.0, 4.0, 6.0, 7.5, 8.0, 9.0, 12.0	Optional
CPI	Characters per inch	5.0, 10.0, 12.0, 13.3, 15.0, 16.7, 18.0, 20.0	Optional
FRONTMGN	Front margin	Single values: *DEVD Other values: <i>Element list</i>	Optional
	Element 1: Offset down	0.0-57.79	
	Element 2: Offset across	0.0-57.79	
BACKMGN	Back margin	Single values: *FRONTMGN , *DEVD Other values: <i>Element list</i>	Optional
	Element 1: Offset down	0.0-57.79	
	Element 2: Offset across	0.0-57.79	
OVRFW	Overflow line number	1-255	Optional
FOLD	Fold records	*NO , *YES	Optional
RPLUNPRT	Unprintable character action	Single values: *NO Other values: <i>Element list</i>	Optional
	Element 1: Replace character	*YES	
	Element 2: Replacement character	X'40'-X'FE', *BLANK	
ALIGN	Align page	*NO , *YES	Optional
DRAWER	Source drawer	1-255, *E1 , *FORMDF	Optional
OUTBIN	Output bin	1-65535, *DEVD	Optional
FONT	Font	Single values: *CPI , *DEVD Other values: <i>Element list</i>	Optional
	Element 1: Identifier	<i>Character value</i> , 2, 002, 3, 003, 5, 005, 8, 008, 10, 010, 11, 011, 12, 012, 13, 013, 18, 018, 19, 019, 20, 020, 21, 021, 25, 025, 26, 026, 30, 030, 31, 031, 36, 036, 38, 038, 39, 039, 40, 040, 41, 041, 42, 042, 43, 043, 44, 044, 46, 046, 49, 049, 50, 050, 51, 051, 52, 052, 55, 055, 61, 061, 62, 062, 63, 063, 64, 064, 66, 066, 68, 068, 69, 069, 70, 070, 71, 071, 72, 072, 74, 074, 75, 075, 76, 076, 78, 078, 80, 080, 84, 084, 85, 085, 86, 086, 87, 087, 91, 091, 92, 092, 95, 095, 96, 096, 98, 098, 99, 099, 101, 102, 103, 109, 110, 111, 112, 154, 155, 157, 158, 159, 160, 162, 163, 164, 167, 168, 173, 174, 175, 178, 179, 180, 181, 182, 183, 186, 187, 188, 189, 190, 191, 194, 195, 204, 205, 211, 212, 221, 222, 223, 225, 226, 229, 230, 232, 233, 234, 244, 245, 247, 248, 249, 252, 253, 254, 255, 256, 258, 259, 279, 281, 282, 285, 290, 300, 304, 305, 306, 307, 318, 319, 400, 404, 416, 420, 424, 428, 432, 434, 435, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 1051, 1053, 1056, 1351, 1653, 1803, 2103, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 4407, 4427, 4535, 4919, 4939, 5047, 5067, 5687, 5707, 5815, 5835, 5943, 6199, 6219, 6327, 6347, 8503, 8523, 8631, 8651, 8759, 8779, 8887, 8907, 12855, 12875, 16951, 16971, 17079, 17099, 33335, 33355, 33463, 33483, 33591, 33601, 33719, 33729, 34103, 34123, 34231, 34251, 37431, 41783, 41803	
	Element 2: Point size	0.1-999.9, *NONE	
FORMFEED	Form feed	*DEVD , *CONT , *CUT , *CONT2 , *AUTOCUT	Optional
PRTQLTY	Print quality	*STD , *DEVD , *DRAFT , *NLQ , *FASTDRAFT	Optional
CTLCHAR	Control character	*NONE , *FCFC , *MACHINE	Optional

Keyword	Description	Choices	Notes
CHLVAL	Channel values	Single values: *NORMAL Other values (up to 12 repetitions): <i>Element list</i>	Optional
	Element 1: Channel	1-12	
	Element 2: Line number for channel	<i>Element list</i>	
	Element 1: Line	1-255	
FIDELITY	Fidelity	*CONTENT, *ABSOLUTE	Optional
CHRID	Character identifier	Single values: *DEVD, *SYSVAL, *JOBCCSID, *CHRIDCTL Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	
DECfmt	Decimal format	*FILE, *JOB	Optional
FNTCHRSET	Font character set	Single values: *FONT Other values: <i>Element list</i>	Optional
	Element 1: Character set	<i>Qualified object name</i>	
	Qualifier 1: Character set	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 2: Code page	<i>Qualified object name</i>	
	Qualifier 1: Code page	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 3: Point size	0.1-999.9, *NONE	
CDEFNT	Coded font	Single values: *FNTCHRSET Other values: <i>Element list</i>	Optional
	Element 1: Coded font	<i>Qualified object name</i>	
	Qualifier 1: Coded font	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 2: Point size	0.1-999.9, *NONE	
PAGDFN	Page definition	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Page definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
FORMDF	Form definition	Single values: *NONE, *DEVD Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Form definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
AFPCHARS	AFP Characters	Single values: *NONE Other values (up to 4 repetitions): <i>Name</i>	Optional
TBLREFCHR	Table Reference Characters	*NO, *YES	Optional
PAGRIT	Degree of page rotation	*AUTO, *DEVD, *COR, 0, 90, 180, 270	Optional
MULTIUP	Pages per side	1-4, 1	Optional
REDUCE	Reduce output	*TEXT, *NONE	Optional
PRTTXT	Print text	<i>Character value</i> , *JOB, *BLANK, X''	Optional
JUSTIFY	Hardware justification	0, 50, 100	Optional
DUPLEX	Print on both sides	*NO, *YES, *TUMBLE, *FORMDF	Optional
UOM	Unit of measure	*INCH, *CM	Optional

Keyword	Description	Choices	Notes
FRONTOVL	Front side overlay	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Overlay	<i>Qualified object name</i>	
	Qualifier 1: Overlay	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 2: Offset down	0.0-57.79, <u>0</u>	
	Element 3: Offset across	0.0-57.79, <u>0</u>	
BACKOVL	Back side overlay	Single values: *FRONTOVL, *NONE Other values: <i>Element list</i>	Optional
	Element 1: Overlay	<i>Qualified object name</i>	
	Qualifier 1: Overlay	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
	Element 2: Offset down	0.0-57.79, <u>0</u>	
	Element 3: Offset across	0.0-57.79, <u>0</u>	
	Element 4: Constant back	*NOCONSTANT, *CONSTANT	
CVTLINDTA	Convert line data	*NO, *YES	Optional
IPDSPASTHR	IPDS pass through	*YES, *NO, *DEV	Optional
USRRSCLIBL	User resource library list	Single values: *DEV, *NONE, *JOB LIBL, *CURLIB Other values (up to 4 repetitions): <i>Character value</i>	Optional
CORNERSTPL	Corner staple	*NONE, *BOTRIGHT, *TOPRIGHT, *TOPLEFT, *BOTLEFT, *DEV	Optional
EDGESTITCH	Edge stitch	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Reference edge	*BOT, *RIGHT, *TOP, *LEFT, *DEV	
	Element 2: Reference edge offset	0.0-57.79, *DEV	
	Element 3: Number of staples	1-122, *DEV	
	Element 4: Staple offsets	Values (up to 122 repetitions): 0.0-57.79, *DEV	
SADLSTITCH	Saddle stitch	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Reference edge	*TOP, *LEFT, *DEV	
	Element 2: Number of staples	1-122, *DEV	
	Element 3: Staple offsets	Values (up to 122 repetitions): 0.0-57.79, *DEV	
FNTRSL	Font resolution for formatting	*DEV, *SEARCH, 240, 300	Optional
DFRWRT	Defer write	*YES, *NO	Optional
SPOOL	Spool the data	*YES, *NO	Optional
OUTQ	Output queue	Single values: *DEV, *JOB Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Output queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
FORMTYPE	Form type	<i>Character value</i> , *STD	Optional
COPIES	Copies	1-255	Optional
PAGERANGE	Page range to print	<i>Element list</i>	Optional
	Element 1: Starting page	<i>Integer</i> , 1, *ENDPAGE	
	Element 2: Ending page	<i>Integer</i> , *END	
MAXRCDS	Max spooled output records	1-999999, *NOMAX	Optional

Keyword	Description	Choices	Notes
FILESEP	File separators	0-9	Optional
SCHEDULE	Spooled output schedule	*JOBEND, *FILEEND, *IMMED	Optional
HOLD	Hold spooled file	*NO, *YES	Optional
SAVE	Save spooled file	*NO, *YES	Optional
OUTPTY	Output priority (on OUTQ)	*JOB, 1, 2, 3, 4, 5, 6, 7, 8, 9	Optional
USRDTA	User data	Character value, *SOURCE	Optional
SPLFOWN	Spool file owner	*CURUSRPRF, *JOB, *CURGRPPRF, *JOBGRPPRF	Optional
USRDFNOPT	User Defined Option	Single values: *NONE Other values (up to 4 repetitions): Character value	Optional
USRDFNDTA	User Defined Data	Character value, *NONE	Optional
USRDFNOBJ	User Defined Object	Single values: *NONE Other values: Element list	Optional
	Element 1: Object	Qualified object name	
	Qualifier 1: Object	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
	Element 2: Object type	*DTAARA, *DTAQ, *FILE, *PSFCFG, *USRIDX, *USRQ, *USRSFC	
SPLFNAME	Spool file name	Name, *FILE	Optional
EXPDATE	Expiration date for file	Date, *NONE, *DAYS	Optional
DAYS	Days until file expires	1-366	Optional
IGCDTA	User specified DBCS data	*NO, *YES	Optional
IGCEXNCHR	DBCS extension characters	*YES, *NO	Optional
IGCCHRRTT	DBCS character rotation	*NO, *YES	Optional
IGCCPI	DBCS characters per inch	*CPI, *CONDENSED, 5, 6, 10	Optional
IGCSOSI	DBCS SO/SI spacing	*YES, *NO, *RIGHT	Optional
IGCCDEFNT	DBCS coded font	Single values: *SYSVAL Other values: Element list	Optional
	Element 1: DBCS coded font	Qualified object name	
	Qualifier 1: DBCS coded font	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
	Element 2: Point size	0.1-999.9, *NONE	
TOSTMF	To stream file	Path name, *NONE	Optional
WSCST	Workstation customizing object	Single values: *NONE, *PDF Other values: Qualified object name	Optional
	Qualifier 1: Workstation customizing object	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
WAITFILE	Maximum file wait time	Integer, *IMMED, *CLS	Optional
LVLCHK	Record format level check	*NO	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSCOPE	Override scope	*ACTGRPDEFN, *CALLLVL, *JOB	Optional
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDEFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the file being used by the program to which this override command is applied. The specified file must be a printer device file when *FILE is specified in the **Overriding to printer file (TOFILE)** parameter. Otherwise, any device file name or database file name is specified.

This is a required parameter.

***PRTF** The *PRTF override is applied. This override applies to all printer files being opened except for those printer files that already have specific overrides. For example, if a *PRTF override is issued at call level 3, and an override is issued for QSYSPRT at call level 3, the *PRTF override is applied to all printer files being opened except for QSYSPRT since there is a specific override for it.

name Specify the names of one or more overridden files for which the overrides in the call level are applied.

Top

Overriding to printer file (TOFILE)

Specifies the printer file that is used instead of the file specified in the **File being overridden (FILE)** parameter; or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this OVRPRTF command override the same parameters specified in the printer file, in the program, or in other called (OVRPRTF) commands.

Single values

***FILE** The printer device file named in the FILE parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to printer file

name Specify the name of the printer device file that is used instead of the overridden 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 used to locate the file. If no library is used as the current library for the job, QGPL is used.

name Specify the library where the file is located.

Top

Device (DEV)

Specifies a printer device description. For nonspoiled output, this identifies the printer device used to produce the printed output. For spooled output, the file is placed on the output queue determined by the OUTQ parameter. If OUTQ(*DEV) is used, the file is placed on the output queue with the same name as the device.

*SYSVAL

The value in the system value QPRTDEV at the time the job is started is used as the printer device.

***JOB** The printer associated with the job is the printer device.

name Specify the name of the device that is used with the printer file. The device name must already be known on the system by a device description.

Double-byte character set considerations: When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).

Top

Printer device type (DEVTYPE)

Specifies the type of data stream that is created for a printer device file. This parameter indicates whether the resulting data stream is an Intelligent Printer Data Stream (IPDS) or an SNA Character Stream (SCS).

***SCS** An SNA Character Stream (SCS) data stream is created.

Note: When using double-byte character set (DBCS) printers (the 5553 and 5583 Printers), DEVTYPE(*SCS) must be specified.

***IPDS** An Intelligent Printer Data Stream (IPDS) is created. This parameter can be specified when using an IPDS printer. If *IPDS is specified and the spooled printer file is directed to a printer other than an IPDS printer, the IPDS printer file is converted to an SCS printer file. More information is in the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

***USERASCII**

An ASCII data stream is placed on a spooled output queue. You are responsible for placing the entire hexadecimal data stream in the buffer, since the AS/400 system does not change or validate the values that are passed. This value cannot be specified with SPOOL(*NO).

***AFPDS**

An advanced function print data stream (AFPDS) is created. Some systems refer to this data stream as MODCA-P.

***AFPDSLIN**

Mixed data (line data and AFPDS data) is created. This value can be specified when using any printer supported by PSF. The printer must be configured with AFP(*YES).

***LINE** Line data is created. This value can be specified when using any printer supported by PSF. The printer must be configured with AFP(*YES).

Top

Page size (PAGESIZE)

Specifies the length and width of the printer forms used by this device file. The length is specified in lines per page or by the units specified for the UOM parameter. The width is specified in print positions (characters) per line or by the units specified for the UOM parameter.

The page size must be specified with reference to the way the data is printed on the page. For example, if using 8.5 inch wide by 11.0 inch long forms and printing at 6 lines per inch with a 10-pitch font, specify PAGESIZE(66 85) PAGRTT(0). However, to rotate the page, specify the page size for an 11.0 inch wide by 8.5 inch long page and enter PAGESIZE(51 110) PAGRTT(90).

Note: Specify PAGRTT(*AUTO) or PAGRTT(*DEV) and PRTQLTY(*DRAFT) on this command to enable automatic reduction or rotation if the data does not fit on the paper.

Specify PAGRTT(*COR) on this command to enable automatic reduction whether or not the data fits on the paper.

Element 1: Page length

0.001-255.0

Specify the page length used by this device file. Although a value ranging from .001 through 255 is allowed, the value specified should not exceed the actual length of the forms used.

Element 2: Page width

0.001-378.0

Specify the page width used by this device file. The value specified should not exceed the actual width of the page used. Valid values for the 3203, 4245, 5211, 5256, 5262, and 3287 printers range from 1 through 132.

Element 3: Measurement method

*ROWCOL

Page-length and page-width are measured as numbers of rows and columns.

*UOM

Page-length and page-width are measured in the units specified on the UOM parameter.

Top

Lines per inch (LPI)

Specifies the line spacing setting on the printer, in lines per inch, used by this device file.

The line spacing on the 5256 printer must be set manually. When the lines per inch (LPI) value on this parameter changes (from the value on the previous printer file), an inquiry message is sent to the message queue associated with the printer that requests a change to the LPI value.

The line spacing on the 4214, 4224, 4230, 4234, 4245, and 5262 Printers is set by a print command. These also allow setting the lines per inch spacing on the control panel of the printer. The lines per inch value must not be set at the printer. If the LPI value is overridden at the control panel, the system overrides the value set with the LPI value of the next printer file received.

- 3 The line spacing on the printer is 3 lines per inch. This value is valid only for double-byte character set (DBCS) printers.
- 4 The line spacing on the printer is 4 lines per inch.
- 6 The line spacing on the printer is 6 lines per inch. This is the default value for this parameter on the CRTPRTF command.
- 7.5 The line spacing on the printer is 7.5 lines per inch. This value is valid only for double-byte character set (DBCS) printers.
- 8 The line spacing on the printer is 8 lines per inch.
Note: When printing double-byte character set (DBCS) data for a file specified with LPI(8), use double spacing. Otherwise, the DBCS data does not print correctly. Alphanumeric data, however, prints correctly in single spacing when LPI(8) is specified.
- 9 The line spacing on the printer is 9 lines per inch.
- 12 The line spacing on the printer is 12 lines per inch.

Double-byte character set considerations: When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).

Top

Characters per inch (CPI)

Specifies the printer character density (in characters per inch) used by this device file.

For the printers that support fonts, the value specified in the font special value implies the CPI. If FONT(*CPI) is specified, the font used is based on the CPI value. The following diagram describes the default font ID for each CPI value:

CPI	FONT ID DEFAULT
-----	-----------------

5	245
---	-----

10	011
----	-----

12	087
----	-----

13.3	204
------	-----

15	222
----	-----

16.7	400
------	-----

18	252
----	-----

20	281
----	-----

5	Character density is 5 characters per inch.
---	---

10	Character density is 10 characters per inch. (This is the default value for this parameter on the CRTPRTF command.)
----	---

12	Character density is 12 characters per inch.
----	--

13.3	Character density is 13.3 characters per inch. This value is valid only for double-byte character set (DBCS) printers.
------	--

15	Character density is 15 characters per inch.
----	--

16.7	Character density is 16.7 characters per inch.
------	--

18	Character density is 18 characters per inch. This value is valid only for double-byte character set (DBCS) printers.
----	--

20	Character density is 20 characters per inch. This value is valid only for double-byte character set (DBCS) printers.
----	--

Double-byte character set considerations: When printing a file that has double-byte character set (DBCS) data, specify a DBCS printer (5553, 5583).

Top

Front margin (FRONTMGN)

Specifies the offset, down and across, of the origin from the edge on the front side of the paper. The offsets are in the units of measure specified on the UOM parameter. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values

*DEVVD

The no-print border from the printer is used to place the text on the page when printing to a printer configured with AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured with AFP(*NO).

Element 1: Offset down

0.0-57.79

Specify the offset of the origin from the top of the page.

Element 2: Offset across

0.0-57.79

Specify the offset of the origin from the left side of the page.

Top

Back margin (BACKMGN)

Specifies the offset, down and across, of the origin from the edge on the back side of the paper. The offsets are in the units of measure specified on the UOM parameter. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values

*FRONTMGN

The offsets specified on the FRONTMGN parameter are used.

*DEVVD

The no-print border from the printer is used to place the text on the page when printing to a printer configured with AFP(*YES). A margin of 0 is used for IPDS printers without a no-print border, or which are configured with AFP(*NO).

Element 1: Offset down

0.0-57.79

Specify the offset of the origin from the top of the page.

Element 2: Offset across

offset-across

Specify the offset of the origin from the left side of the page.

Top

Overflow line number (OVRFLW)

Specifies the line number on the page at which overflow to a new page occurs. Generally, after the specified line is printed, the printer overflows to the next page before printing continues. Overflow is signaled when the specified line number is made the current line, whether printing has occurred on that line or not. The value specified must not exceed the forms length specified in the **Page size (PAGESIZE)** parameter for the file. Margins specified for the printer file are ignored when determining overflow. More information is in the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. This parameter overrides the overflow value specified in the printer file, in the program, or in other called OVRPRTF commands.

1-255 Specify the line number on the current page at which overflow to a new page begins, whether or not printing has occurred on that line. The value specified must not be greater than the page length (PAGESIZE). Margins specified for the printer file are ignored when determining overflow.

Top

Fold records (FOLD)

Specifies whether all positions in a record are printed when the record length exceeds the form width. If so, any portion of the record that cannot be printed on the first line is continued (folded) on the next line or lines until the entire record is printed.

The FOLD parameter is ignored under the following conditions:

- When DEVTYPE(*SCS) is not specified.
- When printing through the OfficeVision program.
- When in the S/36 processing environment.

***YES** Records whose length exceeds the form width are folded on the following lines.

***NO** Records are not folded; if a record is longer than the form width, only the first part of the record that fits on one line is printed.

Double-byte character set considerations: The system ignores this parameter when printing double-byte character set (DBCS) files. The system assumes that DBCS records fit on a printed line. If the record exceeds the form width, the system continues printing the record on the next line.

This parameter overrides the value specified in the printer file, in the program, or in other called OVRPRTF commands.

Top

Unprintable character action (RPLUNPRT)

Specifies whether unprintable characters are replaced and which substitution character (if any) is used. An *unprintable* character is a character that the printer is unable to print.

Note: If *IPDS is specified in the **Printer device type (DEVTYPE)** parameter, a hyphen (-) is printed for the unprintable characters. The substitution character is ignored for the 3287 printer.

Single values

***NO** Unprintable characters are not replaced. When an unprintable character is detected, a message is sent to the program.

Element 1: Replace character

***YES** Unprintable characters are replaced. The program is not notified when unprintable characters are detected.

Element 2: Replacement character

***BLANK**

A blank is used as the substitution character when an unprintable character is detected and *YES is specified.

X'40'-X'FE'

Specify the substitution character that is used each time an unprintable character is detected if *YES is also specified in this parameter. Any printable EBCDIC character can be specified.

Double-byte character set considerations: The system ignores the chosen replacement character when you specify *YES. Instead, the system replaces unprintable characters as follows:

1. If *YES is also specified in the **DBCS extension characters (IGCEXNCHR)** parameter, the system replaces unprintable characters with double-byte character set (DBCS) underscores.

2. If *NO is specified in the IGCEXNCHR parameter, the system replaces all extension characters with the undefined character.

Top

Align page (ALIGN)

Specifies whether the forms must be aligned in the printer before printing is started. If *YES is specified and *NO is specified in the **Spool the data (SPOOL)** parameter, and forms alignment is required, the system sends a message to the message queue specified for the printer, and waits for a reply to the message. If *YES is specified on the SPOOL parameter, and *FILE is specified on the **Align page (ALIGN)** parameter, of the Start Printer Writer (STRPRTWTR) command, this parameter is used to determine whether an alignment message should be sent by the system.

This parameter is ignored when cut sheets are used (spooled and direct output). Page alignment can be done only for text-only files. Page alignment cannot be done for print jobs containing graphics or bar codes.

***NO** No forms alignment is required.

***YES** The forms are aligned before the output is printed.

Top

Source drawer (DRAWER)

Specifies the source drawer used when single-cut sheets are fed into the printer. *AUTOCUT must be specified on the **Form feed (FORMFEED)** parameter.

***E1** The envelopes are fed from the envelope drawer on the sheet-feed paper handler.

***FORMDF**

The paper is fed from the source drawer specified in the form definition. If a form definition is not specified, then source drawer 1 is used.

1-255 Specify the drawer from which the paper is fed.

Top

Output bin (OUTBIN)

Specifies the destination of the output on printers capable of multiple output bins.

***DEV D**

The destination of the output is the device default output bin.

1-65535

Specify the output bin for the destination of the output.

Top

Font specifications (FONT)

Specifies the font identifier and point size used with this printer device file. If a font identifier and point size is not specified, the system automatically sets them.

Single values

***CPI** The identifier of the font with the specified pitch (characters per inch (CPI)) is used.

***DEV**

The font identifier and point size specified in the device description are used.

Element 1: Identifier

identifier

Specify the numeric font identifier being used with this printer device file.

Element 2: Point size

***NONE**

No point size is specified; the system sets one based on the type of printer being used.

0.1-999.9

Specify a point size.

Top

Form feed (FORMFEED)

Specifies the form feed attachment used by this printer device file.

***DEV**

The forms are fed into the printer in the manner specified in the device description.

***CONT**

Continuous forms are used by the printer. The tractor-feed attachment must be put on the printer if this value is specified.

***CONT2**

Continuous forms are used by the printer. The form is fed from the secondary tractor feed attachment. The secondary tractor feed attachment must be on the printer device.

***CUT** Single-cut sheets are used by the printer. Each sheet must be loaded manually. For cut sheets, the forms alignment message is not issued.

***AUTOCUT**

Single-cut sheets are semiautomatically fed into the printer. The sheet-feed attachment must be put on the printer if this value is specified. For cut sheets, the forms alignment message is not issued.

Top

Print quality (PRTQLTY)

Specifies, for the 3812 SCS, 3816 SCS, 4214, 4224, 4230, 4234, and 5219 printers, the quality of print produced.

For the 5219 Printer, different print qualities are produced by varying the speed at which the print ribbon advances. Quality mode (*STD or *NLQ) results in normal print ribbon advancement. In draft mode (*DRAFT), the ribbon advances at a rate of one-third the distance it advances in quality mode. In other words, the 5219 Printer conserves printer ribbon when in draft mode by not advancing it as fast per character printed. The 5219 Printer has a conserve ribbon switch that overrides the value of *DRAFT specified by this parameter.

For the 3812 SCS and 3816 SCS Printers, the automatic hardware selection of computer output reduction printing selected through soft switches on the printers occurs only when *DRAFT is specified for PRTQLTY and PAGRTT is *DEVD. If PAGRTT(*COR) is specified, the PRTQLTY parameter does not affect the printed output.

For the 4224, 4230, and 4234 Printers, standard print quality is produced by varying the density of the dot matrix pattern used to create printable characters. Standard mode (*STD) is the normal mode. Quality mode (*NLQ) requires multiple passes by the printer to produce a line of data. Draft mode (*DRAFT) results in high-speed printing.

For the 4214 printer, only draft (*DRAFT), quality (*NLQ), and device default (*DEVD) modes are supported. Other values are set to quality (*NLQ) mode.

NOTES:

- For the 4214 Printer, quality mode (*STD or *NLQ) is only supported for 10 and 12 characters per inch. If PRTQLTY(*STD or *NLQ) and 5, 15, or 16.7 characters per inch is specified, the data is printed in draft mode.
- For the 4234 Printer, only a limited character set (62 characters) is supported when PRTQLTY(*DRAFT) is specified. A description of the character set supported with draft print quality is in the 4234 Printer Operator's Guide.
- For the 4224 and 4230 printers, the fonts supported are not available for all three print qualities. The OCR-A and OCR-B fonts are supported only with PRTQLTY(*NLQ). The Courier and Essay fonts are available only with PRTQLTY(*NLQ) and PRTQLTY(*STD). The Gothic font is available only with PRTQLTY(*DRAFT) or PRTQLTY(*FASTDRAFT). If there is a mismatch between the print quality and the font selected, the font is changed to match the print quality.
- Specify PAGRTT(*DEVD) and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

***STD** The output is printed with standard quality.

***DRAFT**
The output is printed with draft quality.

***DEVD**
The print quality is set on the printer by the user. It is not set in the data stream.

***NLQ** The output is printed with near letter quality.

***FASTDRAFT**
The output is printed at a higher speed and with lower quality than it would be if you specified *DRAFT. This value is only supported by the 4230 printer.

Top

Control character (CTLCHAR)

Specifies whether the printer device file supports input with print control characters. Any control characters found that are not valid are ignored, and single spacing is assumed.

***NONE**
No print control characters are passed in the data printed.

***FCFC** The first character of every record contains an ANSI forms-control character. If *FCFC is specified, the record length must include one position for the first-character forms-control code. This value is not valid for externally described printer files.

***MACHINE**
The first character of every record contains a machine code control character. If *MACHINE is

specified, the record length must include one extra position for the first character forms control code. This value is not valid for externally described printer files.

If TBLREFCHR(*YES) is also specified, then the record length must include two extra positions for the control character and the table reference character.

Top

Channel values (CHLVAL)

Specifies a list of channel numbers with their assigned line numbers. Use this parameter only if *FCFC is specified in the **Control character (CTLCHAR)** parameter.

Note: If one or more channel-number/line-number combinations are changed, all other combinations must be re-entered.

Single values

*NORMAL

The default values for skipping to channel identifiers are used.

Element 1: Channel

1-12 Specify an American National Standard channel number to be associated with a corresponding 'skip to' line number. Values for this parameter correspond to channels 1 through 12. The CHLVAL parameter associates the channel number with a page line number. For example, if you specify CHLVAL(2 20), channel identifier 2 is allocated with line number 20; therefore, if you place the forms-control 2 in the first position of a record, the printer skips to line 20 before printing the line.

Note: If the printer stops and the next record processed has a channel value forms-control number that is the same value as the line number the printer is on, the printer advances to that value (line number) on the next page. However, if the printer is positioned at the top of the page (line number one) and the channel value forms-control value is associated with line number one, the printer does not advance to a new a new page.

If no line number is specified for a channel identifier, and that channel identifier is encountered in the data, a default of 'space one line' before printing is used. Each channel number can be specified only once.

Element 2: Line number for channel

Element 1: Line

1-255 Specify the line number assigned for the channel number in the same list. Valid line numbers range from 1 through 255. If no line number is assigned to a channel number, and that channel number is encountered in the data, a default of 'space one line' before printing is used. Each line number should be specified only once.

Top

Fidelity (FIDELITY)

Specifies whether printing continues when print errors are found for printers configured with AFP(*YES).

***CONTENT**

Printing continues when errors are found.

***ABSOLUTE**

Printing stops when errors are found.

Top

Character identifier (CHRID)

Specifies the character identifier (graphic character set and code page) for the file. This parameter allows you to print text that is in different character identifier (graphic character set and code page) coding. The value specified on this parameter is used to command the printer device to interpret the hexadecimal byte string by printing the same characters that were intended when the text was created.

Single values

***DEV**

The default CHRID value that the device is designed to handle is used. Character selection is normal because the file has the same character identifier as the device default.

***SYSVAL**

The CHRID value specified for the system on which the application runs is used.

***JOBCCSID**

The character identifier for the printer file is taken from the coded character set identifier (CCSID) of the job.

Note: This value is not allowed if the file was created on a system at an earlier release level than V2R3M0.

Element 1: Graphic character set

integer

Specify the graphic character set value that matches the printer. Valid values range from 1 through 32767.

Element 2: Code page

integer

Specify the code page value that matches the printer. Valid values range from 1 through 32767.

Top

Decimal format (DECFMT)

Specifies which decimal format value is used when editing numeric fields with the EDTCDE DDS keyword. The decimal format value determines the use of commas and periods for the decimal position and three digit positional separators on edited fields.

***FILE** Use the decimal format value stored with the file when the file was created.

***JOB** Use the decimal format value from the DECFMT job attribute when the file is opened.

Top

Font character set (FNTCHRSET)

Specifies a downloaded font consisting of a character set and code page. For an outline font, a point size is required. For a raster font, the point size is ignored. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values

*FONT

The value specified on the FONT parameter is used.

Element 1: Character set

Qualifier 1: Character set

name Specify the name of the font character set.

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 font character set. 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 font character set is located.

Element 2: Code page

Qualifier 1: Code page

name Specify the name of the code page.

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 code page name. 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 code page name is located.

Element 3: Point size

*NONE

The point size is supplied by the system and is determined by the specified font character set.

0.1-999.9

Specify a point size.

Top

Coded font (CDEFNT)

Specifies the coded font that the system uses for single-byte character set (SBCS) printing. This parameter can only be used for printer files with DEVTYPE(*AFPDS) specified.

Single values

***FNTCHRSET**

The font specified on the FNTCHRSET parameter is used.

Element 1: Coded font

Qualifier 1: Coded font

name Specify the coded font name to use.

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 coded font name. 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 coded font name is located.

Element 2: Point size

***NONE**

The point size is supplied by the system and is determined by the specified font character set.

0.1-999.9

Specify a point size.

Top

Page definition (PAGDFN)

Specifies the page definition to be used to format line data.

You can specify a page definition with *LINE, *AFPDSLIN, or *USERASCII data. PSF/400 will convert the line data and page definition to IPDS.

When you specify a page definition on the printer file, some printer file parameters will be ignored when the spooled file is printed by PSF/400. The following print file parameters will be ignored:

- CDEFNT
- CHRID
- CPI
- FNTCHRSET
- FOLD
- FONT
- LPI

- MULTIUP
- PAGESIZE
- PAGRTT
- REDUCE

Single values

*NONE

No page definition is specified.

Because PSF/400 requires a page definition when *LINE or *AFPDSLINe is specified, an inline page definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

Qualifier 1: Page definition

name Specify the name of the page definition that must exist in the library specified. Valid values range from 1 to 8 characters. Device type *AFPDSLINe, *LINE, or *USERASCII must be specified when using a page definition.

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 definition (FORMDF)

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form, including overlays, position of page data on the form, and number of copies of pages and modifications to pages. The form definition is located inline with the file being printed, or in a library.

When you specify a form definition (*DEVd or form definition name) on the printer file, some printer file parameters will be ignored when the spooled file is printed by PSF/400. The following print file parameters will be ignored:

- DUPLEX (If *FORMDF specified)
- DRAWER (If *FORMDF specified)
- PAGRTT
- PRTQLTY
- FORMFEED
- FRONTMGN
- BACKMGN
- MULTIUP
- REDUCE
- CORNERSTPL
- EDGESTITCH
- SADLSTITCH

Single values

***NONE**

No form definition is used.

Because PSF/400 requires a form definition, an inline form definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

***DEV D**

The name of the form definition is specified in the printer device description.

Qualifier 1: Form definition

name Specify the name of the form definition that must exist in the library specified. Valid values range from 1 to 8 characters.

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

AFP Characters (AFPCHARS)

Specifies one or more AFP characters (coded fonts) to be used with line data and a page definition.

Single values

***NONE**

No AFP characters (coded fonts) specified.

Other values (up to 4 repetitions)

name Specify up to four 4-byte names of coded fonts to be specified with the line data and a page definition. The 4-byte names are concatenated to X0 to identify up to four coded fonts which are to be used when TBLREFCHR is being used within the data.

Top

Table Reference Characters (TBLREFCHR)

Specifies whether table reference characters are present in the line data.

***NO** No table reference character is present in line data.

***YES** Table reference characters are present in line data.

If forms control characters are used with the data, the table reference character follows the forms control character but precedes the data bytes. If forms control characters are not used, the table reference character is the first byte of the data record. As with forms control character, if table reference characters are used, every data record must contain a TRC byte.

Top

Degree of page rotation (PAGRTT)

Specifies the degree of rotation of the text on the page with respect to the way the form is loaded into the printer.

*AUTO

Indicates that automatic rotation of output is done to fit the printed data on the form. If rotation does not accomplish this, computer output reduction is performed automatically (regardless of the print quality being used). This parameter is valid only for printers supporting rotation.

*DEV D

The operating system sends a device default rotation value to the printer. Page rotation is dependent on your printer's specifications. See your printer or printer emulation documentation to determine how page rotation is affected.

*COR Computer output reduction is done. Computer output reduction allows printed output intended for a 13.2-inch wide by 11.0-inch long form to be printed on an 8.5-inch wide by 11.0-inch long form.

0 No rotation is done. Printing starts at the edge loaded into the printer first, and is parallel to that edge.

90 Rotation of the text is done 90 degrees clockwise from the 0 degree writing position.

180 Rotation of the text is done 180 degrees clockwise from the 0 degree writing position.

270 Rotation of the text is done 270 degrees clockwise from the 0 degree writing position.

Top

Pages per side (MULTIUP)

Specifies, for spooled output only, whether multiple pages of output are printed on 1 physical page.

1 One page of output is printed on one physical sheet of paper.

2 Two pages of output are printed on 1 physical sheet of paper.

3 Three pages of output are printed on 1 physical sheet of paper.

4 Four pages of output are printed on 1 physical sheet of paper.

Top

Reduce output (REDUCE)

Specifies whether to reduce the output when doing multiple up printing.

For examples and more details see the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

*TEXT

The text output is reduced when doing multiple up printing.

*NONE

The output is not reduced when doing multiple up printing.

Top

Print text (PRTTXT)

Specifies the text that is printed at the bottom of each page of printed output and on separator pages.

***JOB** The value from the current job is used.

***BLANK**
No text is printed.

character-value
Specify no more than 30 characters of text, enclosed in apostrophes.

Top

Hardware justification (JUSTIFY)

Specifies the printing positions of the characters on a page so that the right-hand margin of printing is regular.

0 No justification occurs. (This is the default value for this parameter on the CRTPRTF command.)

50 Spaces are added to the blanks in the text so that the right margin is more closely aligned, but not flush.

100 The text is expanded by spaces (added where the blanks already exist) until the right margin is flush.

Top

Print on both sides (DUPLEX)

Specifies whether output is printed on one side or two sides of the paper.

***NO** The output is printed on one side of the paper.

***YES** The output is printed on both sides of the paper, with the top of each printed page at the same end of the sheet of paper.

***TUMBLE**
The output is printed on both sides of the paper, with the top of one printed page at the opposite end from the top of the other printed page. This is usually used for output that will be bound at the top.

***FORMDF**
The output is printed on both sides of the paper if the duplex value is specified in the form definition. If a form definition is not specified, then the output is printed on one side of the paper.

Top

Unit of measure (UOM)

Specifies the unit of measurement to be used.

***INCH**
The inch is used as the unit of measurement.

***CM** The centimeter is used as the unit of measurement.

Top

Front side overlay (FRONTOVL)

Specifies the qualified name of the object that contains both the overlay that is printed on the front side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

Single values

*NONE

No overlay is used.

Element 1: Overlay

Qualifier 1: Overlay

name Specify the name of the overlay.

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 overlay. 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 overlay is located.

Element 2: Offset down

0 No offset down from the point of origin is used.

0.0-57.79

Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset across

0 No offset across from the point of origin is used.

0.0-57.79

Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Top

Back side overlay (BACKOVL)

Specifies the object name and library name containing both the overlay that is printed on the BACK side of the page and the offset, down and across, from the point of origin used when the overlay is printed.

The constant back function allows you to print overlays on blank pages without adding blank pages to the print application. Specifying the constant back function would cause, for each page generated by the application program, a blank page to be generated onto which the specified back overlay could be printed. The generated blank pages are called constant forms because no variable data from the user's

program is printed on the pages. The constant back function is only supported for duplex printing. It is ignored when DUPLEX(*NO) is specified on the printer file.

Note that the offset down and offset across values are ignored when *CONSTANT is specified for constant back. An offset of 0.0 is assumed for these values.

Single values

*FRONTOVL

The values that are specified on the FRONTOVL parameter are used.

*NONE

No overlay is used.

Element 1: Overlay

Qualifier 1: Overlay

name Specify the name of the overlay.

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 overlay. 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 overlay is located.

Element 2: Offset down

0 No offset down from the point of origin is used.

0.0-57.79

Specify the offset down from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 3: Offset across

0 No offset across from the point of origin is used.

0.0-57.79

Specify the offset across from the point of origin at which to begin printing the overlay. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57.

Element 4: Constant back

*NOCONSTANT

No constant back is specified.

*CONSTANT

Constant back is specified.

Top

Convert line data (CVTLINDTA)

Specifies whether line data and a page definition should be converted to AFPDS before the data is spooled.

***NO** No AFPDS conversion is done.

***YES** Specifies that AFPDS conversion is to be done on the line data and page definition before the data is spooled.

Top

IPDS pass through (IPDSPASTHR)

Specifies whether IPDS (intelligent printer data stream) pass-through is done for the spooled file.

***DEV D**

The value specified for IPDSPASTHR in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of ***NO** is used.

***NO** No IPDS pass-through is done.

***YES** Specifies that IPDS pass-through is to be done if the spooled file is eligible for IPDS pass-through.

Note: Not all SCS or IPDS spooled files are eligible for IPDS pass-through. They may contain special functions that require transform to AFPDS for correct printing. Specifying IPDS pass-through on the printer file allows only those spooled files eligible for IPDS pass-through to bypass the extra transforms. Those spooled files not eligible for IPDS pass-through will still undergo the transforms to AFPDS and back to IPDS.

IPDS pass-through will not be valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts can not support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts which are downloaded to the printer. All IBM IPDS printers, except for the following, can be supported with IPDS pass-through: 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900-001 and any printer attached using Print Services Facility for OS/2's Distributed Print Function.

For V3R7, V4R1 and V4R2, IPDSPASTHR can be specified with the USRDFN DTA parameter in a printer file. You may continue using this support with existing printer files and PSF configuration objects by specifying IPDSPASTHR(*DEV D) in the printer file. If you specify a value of anything other than *DEV D for the IPDSPASTHR parameter, any IPDS pass-through value in the USRDFN DTA parameter is ignored.

Top

User resource library list (USRRSCLIBL)

Specifies the list of user resource libraries to be used for searching for AFP resources for a spooled file. If the AFP resource is not found in the user resource libraries, then the library list specified in the DEVRSCLIBL parameter of the PSF configuration object is searched. If no PSF configuration object is specified for the device, then libraries QFN TCPL, QFN T01-QFN T19, and QFN T61-69 are searched.

Single values

***DEV D**

The value specified for USRRSCLIBL in the PSF configuration object specified for a printer device description is used. If no PSF configuration object is specified for the device, a value of *JOB LIBL is used.

***NONE**

No user libraries are specified.

***JOBLIBL**

Specifies that the library list of the job that created the spool file is used in searching for AFP resources. This library list is saved with the spool file when it is created.

***CURLIB**

Specifies that the current library of the job that created the spool file is used for searching for AFP resources. If no library is specified as the current library for the job, then library QGPL is used.

Other values (up to 4 repetitions)

character-value

Specify the name of a library that will be used to search for AFP resources. Up to four library names may be specified.

For V3R7, V4R1 and V4R2, USRRSCLIBL can be specified with the USRDFNDDTA parameter in a printer file. PSF/400 uses that value if USRRSCLIBL(*PRTF) is specified in a PSF configuration object which is specified in the printer device description. You may continue using this support with existing printer files and PSF configuration objects by specifying USRRSCLIBL(*DEVDD) in the printer file. If you specify a value of anything other than *DEVDD for the USRRSCLIBL parameter, any user resource library value in the USRDFNDDTA parameter is ignored.

Top

Corner staple (CORNERSTPL)

Specifies the reference corner to be used for a corner staple. A staple is driven into the media at the reference corner. Refer to your printer's documentation for information as to which reference corners are supported. Page rotation does not affect the placement of a corner staple.

***NONE**

A corner staple is not specified.

***DEVDD**

The reference corner is the default reference corner used by the device.

***BOTRIGHT**

The reference corner is the bottom right corner of the media.

***TOPRIGHT**

The reference corner is the top right corner of the media.

***TOPLEFT**

The reference corner is the top left corner of the media.

***BOTLEFT**

The reference corner is the bottom left corner of the media.

Top

Edge stitch (EDGESTITCH)

Specifies where one or more staples are driven into the media along the finishing operation axis. Refer to your printer's documentation for information about which elements of this parameter are supported and which values for each element are supported. If specification of a value for an element is not supported by a printer, specify a value of *DEV D for that element. Page rotation does not affect the placement of an edge stitch.

Single values

*NONE

An edge stitch is not specified.

Element 1: Reference edge

Specifies the reference edge to be used for an edge stitch. An edge stitch is formed by having one or more staples driven into the media along the finishing operation axis.

*DEV D

The reference edge is the default reference edge used by the device.

*BOTTOM

The reference edge is the bottom edge of the media.

*RIGHT

The reference edge is the right edge of the media.

*TOP The reference edge is the top edge of the media.

*LEFT The reference edge is the left edge of the media.

Element 2: Reference edge offset

Specifies the offset of the edge stitch from the reference edge toward the center of the media.

*DEV D

The reference edge offset is the default reference edge offset used by the device.

0.0-57.79

Specifies the offset of the edge stitch from the reference edge. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

Element 3: Number of staples

Specifies the number of staples that are to be applied along the finishing operation axis.

*DEV D

The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEV D is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

1-122 Specify the number of staples to be used for the edge stitch. If the number of staples is specified, then *DEV D must be specified for staple offsets. The device default for the spacing of each staple will be used.

Element 4: Staple offsets

Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

***DEV D**

The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

0.0-57.79

Specify the staple offset for each staple in the edge stitch. Up to 122 staple offsets may be specified. If one or more staple offset values are specified, then *DEV D must be specified for the number of staples. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

Top

Saddle stitch (SADLSTITCH)

Specifies where one or more staples are driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge. Page rotation does not affect the placement of a saddle stitch.

Single values

***NONE**

A saddle stitch is not specified.

Element 1: Reference edge

Specifies the reference edge to be used for a saddle stitch. A saddle stitch is formed by having one or more staples driven into the media along the finishing operation axis, which is positioned at the center of the media parallel to the reference edge.

***DEV D**

The reference edge is the default reference edge used by the device.

***TOP** The reference edge is the top edge of the media.

***LEFT** The reference edge is the left edge of the media.

Element 2: Number of staples

Specifies the number of staples that are to be applied along the finishing operation axis.

***DEV D**

The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEV D is also specified or defaulted for the Staple Offsets element value, then the number of staples is the default number of staples used by the device. If one or more offsets are specified for Staple Offsets, the number of staples is the same as the number of staple offsets specified.

1-122 Specify the number of staples to be used for the saddle stitch. If you specify the number of staples, then *DEV D must be specified for staple offsets. The device default for the spacing of each staple will be used.

Element 3: Staple offsets

Specifies the offset of the staples along the finishing operation axis. The offset is measured from the point where the finishing operation axis intersects either the bottom edge or the left edge of the media, toward the center of the media. Each consecutive value is used to position a single finishing operation centered on the specified point on the finishing operation axis.

***DEVD**

The staple offsets are the default staple positions used by the device. If a value was specified for the Number of Staples element, the staple position of each staple will be calculated automatically by the printer.

0.0-57.79

Specify the staple offset for each staple in the saddle stitch. Up to 122 staple offsets may be specified. If one or more staple offset values are specified, then *DEVD must be specified for the number of staples. If UOM(*CM) is specified, valid values range from 0 through 57.79, and if UOM(*INCH) is specified, valid values range from 0 through 22.57. This value is converted to millimeters for the printer. Fractional millimeters are not supported and are discarded when conversion to millimeters is performed.

Top

Font resolution for formatting (RNTRSL)

Specifies the resolution PSF/400 uses when printing to a multiple resolution printer configured to report multiple resolutions, but the spooled file does not specify the font metrics and resolution or the font is not available at the resolution that is contained in the spooled file.

For more information regarding the algorithm used for searching a library list for a font resource, see the Printer Device Programming manual section entitled User and Device Resource Library Lists in the chapter called Working With PSF configuration objects.

***DEVD**

The value specified in the FNTRSL parameter of the PSF configuration object for the device is used. If no PSF configuration object is specified for the device, a value of *SEARCH is used.

***SEARCH**

Specifies to search the library list for the first occurrence of a host font with a name match. The resolution of that font is used to print the spool file. Message PQT3546 is sent to specify the resolution of the font that was selected.

240 The font resolution is 240 pels per inch.

300 The font resolution is 300 pels per inch.

Top

Defer write (DFRWRT)

Specifies whether output is held in the system buffer before being sent to the printer.

***YES** The system controls the amount of output that is held in the buffer before it is sent to the printer.

***NO** If *NO is specified on this parameter and if *NO is specified on the **Spool the data (SPOOL)** parameter, output is not held in the buffer. Instead, output is sent immediately to the printer once the program has performed a write operation.

If *NO is specified on this parameter and if *YES is specified on the SPOOL parameter and if *IMMED is specified on the **Spooled output schedule (SCHEDULE)** parameter, output is held in the buffer until a page of output is available or until the system buffer is full.

Spool the data (SPOOL)

Specifies whether the output data for the printer device file is spooled.

***YES** The data is spooled for processing by a diskette writer or printer writer.

***NO** The data is not spooled. It is sent directly to the device to print as the output becomes available.

Top

Output queue (OUTQ)

Specifies the output queue used for spooled files that specify OUTQ(*JOB). This parameter applies only to printer files that have *JOB specified for the OUTQ parameter.

Single values

***DEV** The output queue associated with the printer specified on the DEV parameter is used. The output queue has the same name as the printer.

***JOB** The output queue associated with this job is used for the spooled output.

Qualifier 1: Output queue

name Specify the name of the output queue to which the output data is spooled.

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 library is specified as the current library for the job, QGPL is used.

name Specify the library where the output queue is located.

Top

Form type (FORMTYPE)

Specifies the type of forms used in the printer for printed output that is produced using this device file. If a form type other than *STD is specified, the system sends a message that identifies the form type to the system operator when the output is produced, and requests that the specified type of forms be put in the printer. This parameter overrides the form type value specified in the printer file or in other called OVRPRTF commands.

***STD** The standard printer form for your computer system is used.

character-value

Specify a form type identifier, having 10 characters or less, for the printer forms used.

Top

Copies (COPIES)

Specifies, for spooled output only, the number of copies of the output being printed.

Page range to print (PAGERANGE)

Specifies, for spooled output files only, the starting and ending pages to print.

Element 1: Starting page

*ENDPAGE

Use the end page value as the starting page.

integer

Specify the starting page to print.

Element 2: Ending page

*END Printing continues until the end of file.

integer

Specify the ending page to print.

Max spooled output records (MAXRCDS)

Specifies, for spooled output only, the maximum number of records that can be in the spooled file for jobs using the printer file. This parameter overrides the value specified in the printer file or in other called OVRPRTF commands.

*NOMAX

There is no maximum on the number of records that can be in the spooled file.

1-999999

Specify the maximum number of records that can be in the spooled output file.

File separators (FILESEP)

Specifies, for spooled output only, the number of separator pages placed at the beginning of each printed file, including the pages between multiple copies of the same output. Each separator page has the following items printed on it: file name, file number, job name, user name, and the job number. This parameter overrides the separator value specified in the printer file or in other called OVRPRTF commands.

0-9 Specify the number of separator pages used at the start of each printed output file produced by this device file. If 0 is specified, no separator pages are printed for the file. In this case, the printed output for each file (or copy of a file) starts at the top of a new page.

Spooled output schedule (SCHEDULE)

Specifies, for spooled output files only, when the spooled output file is made available to a spooling writer. This parameter overrides the scheduling value specified in the printer file or in other called OVRPRTF commands.

***JOBEND**

The spooled output file is available to the spooling writer only after the entire job is completed.

***FILEEND**

The spooled output file is available to the spooling writer as soon as the file is closed in the program.

***IMMED**

The spooled output file is made available to the writer as soon as the file is opened in the program.

Top

Hold spooled file (HOLD)

Specifies, for spooled output files only, whether the spooled file is held. The spooled file can be released by using the Release Spooled File (RLSSPLF) command.

Note: This parameter overrides the hold value specified in the printer file or in other called OVRPRTF commands.

***NO** The spooled output file is not held on the output queue. The spooled output is available to a spooling writer based on the **Spooled output schedule (SCHEDULE)** parameter value.

***YES** The spooled output file is held until it is released by the Release Spooled File (RLSSPLF) command.

Top

Save spooled file (SAVE)

Specifies, for spooled output only, whether the spooled file is saved after the output is produced. This parameter overrides the save value specified in the printer file or in other called OVRPRTF commands.

***NO** The spooled file data is not kept (saved) on the output queue after it is produced.

***YES** The spooled file data is kept on the output queue until the file is deleted. After the file is produced, the number of copies is set to 1, and the status of the file is changed from WTR to SAV.

Top

Output priority (on OUTQ) (OUTPTY)

Specifies the output priority for spooled output files that are produced by this job. The highest priority is 1 and the lowest priority is 9.

***JOB** The output priority associated with the job that created the spooled file is used.

output-priority

Specify a number ranging from 1 (high) through 9 (low) to indicate the output priority.

Top

User data (USRDTA)

Specifies, for spooled output, user-specified data that identifies the file.

***SOURCE**

If the file was created by a System/36 procedure, the name of the procedure is assigned. If the file was created by an application program, the name of the program is assigned.

character-value

Specify up to 10 characters of user-specified text.

Top

Spool file owner (SPLFOWN)

Specifies, for spooled output only, who the owner of the spooled file is.

***CURUSRPRF**

The spooled file is owned by the current effective user of the current job or thread. See the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more detailed information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsyssetuid() - Set User ID
- qsysseteuid() - Set Effective User ID
- qsyssetreuid() - Set Real and Effective User ID

***JOB** The spooled file is owned by the original user profile of the job. If the job has switched to a new user profile, the original user profile is still the owner of the spooled file.

***CURGRPPRF**

The spooled file is owned by the current effective group profile of the current job or thread. If there is no current effective group profile, ownership of the spooled file is determined in the same manner as *CURUSRPRF. See the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more detailed information on how the SPLFOWN parameter is affected when using any of the following APIs:

- QWTSETP - Set Profile
- qsyssetgid() - Set Group ID
- qsyssetegid() - Set Effective Group ID
- qsyssetregid() - Set Real and Effective Group ID

***JOBGRPPRF**

The spooled file is owned by the group profile of the original user profile of the job. If the job has switched to a new user profile, the group profile of the original user profile is still the owner of the spooled file. If no group profile exists, ownership of the spooled file is determined the same way as *JOB.

Top

User Defined Option (USRDFNOPT)

Specifies, for spooled output only, one or more user-defined options to be used by user applications or user-specified programs that process spooled files. A maximum of four user-defined options can be specified.

This parameter overrides the user-defined options specified in the printer file or in other called OVRPRTF commands.

Single values

***NONE**

No user-defined options specified.

Other values (up to 4 repetitions)

character-value

Specify a user-defined option to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

Top

User Defined Data (USRDFNDA)

Specifies, for spooled output only, the user-defined data to be used by user applications or user-specified programs that process spooled files.

This parameter overrides the user-defined data specified in the printer file or in other called OVRPRTF commands.

***NONE**

No user-defined data specified.

character-value

Specify a user-defined data to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

Top

User Defined Object (USRDFNOBJ)

Specifies, for spooled output only, the user-defined object to be used by user applications or user-specified programs that process spooled files.

This parameter overrides the user-defined object name specified in the printer file or in other called OVRPRTF commands.

Single values

***NONE**

No user-defined object specified.

Element 1: Object

Qualifier 1: Object

name Specify the name of the user-defined object to be used by user applications or user-specified programs that process spooled 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.

Element 2: Object type

object-type

The user object type can be one of the following:

***DTAARA**

Data Area

***DTAQ**

Data Queue

***FILE** File

***PSFCFG**

PSF Configuration Object

***USRIDX**

User Index

***USRQ**

User Queue

***USRSPC**

User Space

Top

Spool file name (SPLFNAME)

Specifies, for spooled output only, the spooled output file name.

***FILE** The name of the printer file is used for the spooled output file name.

name Specify the name of the spooled output file. A maximum of 10 characters can be used.

Top

Expiration date for file (EXPDATE)

Specifies the expiration date for the spooled file. The spooled file will expire at 23:59:59, system local time on the date specified.

***NONE**

No expiration date is specified.

***DAYS**

The expiration date is to be calculated using the value specified for the **Days until file expires (DAYS)** parameter.

date Specify the date after which the spooled file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXPSPLF) command. The date must be enclosed in apostrophes if date separator characters are used in the value.

Top

Days until file expires (DAYS)

Specifies the number of days to keep the spooled file.

Note: A value must be specified for this parameter if the **Expiration date for file (EXPDATE)** parameter has a value of *DAYS. If the EXPDATE parameter has a value other than *DAYS, no value is allowed for this parameter.

1-366 Specify an interval in days after which the spooled file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXPSPLF) command. The actual expiration date applied to the spooled file is calculated by adding the number of days specified to the date the printer file is opened.

Top

User specified DBCS data (IGCDTA)

Specifies, for program-described files, whether the file processes double-byte character set (DBCS) data. Specifies, for externally described files, the DBCS attributes of the file.

For program-described files:

***NO** The file does not process double-byte character set (DBCS) data.

***YES** The file processes double-byte character set (DBCS) data.

For externally-described files:

***NO** The only double-byte character set (DBCS) attributes of the file are those defined in the DDS.

***YES** DBCS attributes, in addition to those defined in the DDS, include putting the DDS keyword for alternative data type (IGCALTTYP) into effect, and identifying DBCS attributes of fields or messages not identified in the DDS.

Top

DBCS extension characters (IGCEXNCHR)

Specifies whether the system processes double-byte character set (DBCS) extension characters.

***YES** The system processes extension characters.

***NO** The system does not process extension characters; it prints them as undefined characters.

Top

DBCS character rotation (IGCCHRRTT)

Specifies whether the printer should rotate double-byte character set (DBCS) data 90 degrees counterclockwise when printing. The system prints rotated DBCS characters vertically so that they appear in proper reading sequence. Alphanumeric characters are not rotated.

***NO** The system does not rotate DBCS data when printing.

***YES** The system rotates DBCS data 90 degrees counterclockwise when printing.

Top

DBCS characters per inch (IGCCPI)

Specifies the printer character density of double-byte character set (DBCS) data, in characters per inch (cpi).

***CPI** DBCS density is based on the values specified for the **Characters per inch (CPI)** parameter. The system prints one double-byte character for every two alphanumeric characters. (*CPI is the default value for this parameter on the CRTPRTF command.)

- For CPI(10), DBCS characters print at 5 characters per inch.
- For CPI(12), DBCS characters print at 6 characters per inch.
- For CPI(13.3), DBCS characters print at 6.7 characters per inch (same as IGCCPI(*CONDENSED)).
- For CPI(15), DBCS characters print at 7.5 characters per inch.
- For CPI(18), DBCS characters print at 9 characters per inch.
- For CPI(20), DBCS characters print at 10 characters per inch.

***CONDENSED**

Condensed printing, where the system prints 20 double-byte characters every three inches, is used.

- 5 DBCS density is 5 characters per inch.
- 6 DBCS density is 6 characters per inch.
- 10 DBCS density is 10 characters per inch.

Top

DBCS SO/SI spacing (IGCSOSI)

Specifies how the system prints shift-in and shift-out characters.

***YES** The system prints shift control characters as blanks.

***NO** The system does not print shift control characters. These characters do not occupy a position on printed output.

***RIGHT**

The system prints two blanks when printing shift-in characters, but does not print shift-out characters.

Top

DBCS coded font (IGCCDEFNT)

Specifies the coded font that the system uses for double-byte character set (DBCS) printing. This parameter is only used when using printers configured with AFP(*YES).

Single values

***SYSVAL**

The DBCS coded font specified in the system value is used.

Element 1: DBCS coded font

Qualifier 1: DBCS coded font

name Specify the DBCS coded font name to use.

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 coded font name. 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 coded font name is located.

Element 2: Point size

***NONE**

The point size is supplied by the system and is determined by the specified font character set.

0.1-999.9

Specify a point size.

Top

To stream file (TOSTMF)

Specifies the directory or stream file where the output data is to be written. All directories in the path name must exist. New directories are not created. This parameter is only valid when the **Workstation customizing object (WSCST)** parameter is not *NONE, or the **Printer device type (DEVTYPE)** parameter is *AFPDS.

If the TOSTMF value refers to a directory, the system will create a stream file in that directory with a unique name derived from the printer file name. If the TOSTMF value is a file name, a stream file by that name must not already exist.

***NONE**

The output is written to a spooled file if the **SPOOL** parameter is *YES, or directly to the printer device (**DEV**) if the **SPOOL** parameter is *NO.

path-name

Specify the path name for the directory or stream file where you want the output data to be written.

Top

Workstation customizing object (WSCST)

Specifies the workstation customizing object to use to transform the printer file output to final form before writing it to a stream file. This parameter is only valid when the **To stream file (TOSTMF)** parameter is not *NONE.

Single values

***NONE**

The output is not transformed to final form before writing to a stream file.

***PDF** The output is transformed to Portable Document Format (PDF) before it is written into a stream file.

Qualifier 1: Workstation customizing object

name Specify the name of the customizing object.

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

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

***IMMED**

The program does not wait. Immediate allocation of file resources is required.

***CLS** The default wait time specified in the class description is used as the wait time for the file resources to be allocated.

integer

Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767.

Top

Record format level check (LVLCHK)

Specifies whether the level of the device file is checked when the file is opened by a program. For this check, which is done while the file is opened, the system compares the record format identifiers of each record format used by the program with the corresponding identifiers in the device file. Because the same record format name can exist in more than one file, each record format is given a unique internal system identifier when the format is created. Level checking cannot be done unless the program contains the record format identifiers. This command cannot override level checking from *NO to *YES.

***NO** The level identifiers are not checked when the file is opened.

Top

Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of previously called file override commands.

***NO** This file is not protected from other file overrides. Its values are overridden by the effects of any previously called file override commands.

***YES** This file is protected from the effects of any previously called file override commands.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDFN

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLVL

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDFN

The scope of the open operation is determined by the activation group of the program that called the OVRPRTF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Examples

Example 1: Printing Output

```
OVRPRTF FILE(PRINTOUT) TOFILE(PRINT3) SPOOL(*YES)
        COPIES(5) OUTQ(OUTPUT1)
```

This command overrides the file named PRINTOUT and uses the printer file named PRINT3 to produce the spooled output on the printer. The output from the program is sent to the OUTPUT1 output queue. Five copies of the spooled file are printed on the printer specified on the Start Printer Writer (STRPRTWTR) command.

Example 2: Rotating Double-Byte Characters

```
OVRPRTF FILE(IGCPRT) IGCDDTA(*YES) IGCCHRRTT(*YES)
```

This command overrides the IGCPRT printer file to make the file DBCS-capable.. The override puts the IGCALTTYP DDS keyword into effect to change character output fields to DBCS fields, and rotates the double-byte characters when printing.

Example 3: Generating Stream File Output

```
OVRPRTF FILE(QSYSPRT) DEVTYPE(*AFPDS)  
TOSTMF('/home/user/mypdf.pdf') WSCST(*PDF)
```

This command overrides the QSYSPRT printer file to generate a stream file "mypdf.pdf" in directory "/home/user". As it is generated, the output stream file will be converted to Portable Document Format (PDF).

[Top](#)

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF7343

Channel number specified more than once on CHLVAL.

[Top](#)

Override with Save File (OVRSAVF)

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

Parameters
Examples
Error messages

The Override with Save File (OVRSAVF) command is used (1) to override or replace a file named in a program, (2) to override certain attributes of a file that are used by a program, or (3) to override the file named in a program and certain attributes of the overriding file.

This command does not apply to save and restore commands.

More information on overriding files is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i>	Required, Positional 1
TOFILE	Save file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
EXTEND	Extend file	*NO, *YES	Optional
POSITION	Starting position in file	Single values: *START Other values: <i>Element list</i>	Optional
	Element 1: Retrieve order	*RRN	
	Element 2: *RRN—Record number	<i>Unsigned integer</i>	
WAITFILE	Maximum file wait time	<i>Integer</i> , *IMMED, *CLS	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSOPE	Override scope	*ACTGRPDEFN, *CALLLVL, *JOB	Optional
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDEFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the save file in the using program to which this override command is applied. The specified file must be a save file when *FILE is specified in the **Save file (TOFILE)** parameter.

Note: The information in a save file has meaning only to Operating System/400 save and restore; redirecting another type of file to a save file or vice versa is not recommended.

This is a required parameter.

name Specify the name of the save file.

Top

Save file (TOFILE)

Specifies the save file that is used instead of the file specified on the **File being overridden (FILE)** parameter or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified on this command. The parameters specified on this command override the other values specified in the save file or in the program.

Single values

***FILE** The save file named in the FILE parameter has certain parameters overridden by the values specified in this command.

Qualifier 1: Save file

name Specify the name of the save file that is used instead of the overridden file name.

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 save file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the save file is located.

Top

Extend file (EXTEND)

Specifies, for output operations only, whether new records are added to the end of the data currently in the save file. This option is used to start processing after an application or a system failure. When this operation is completed, the file must contain the image of a single save operation made by a save command, or it may not be possible to restore objects from the save file. This parameter overrides the extend value specified in the program. The sequencing information in the file's records guarantees that after a system failure, a record cannot be skipped or sent twice.

***NO** Records are not added to the end of the specified save file, but they replace existing records in the file. If the save file already contains records, an inquiry message is sent that clears the file or cancels the operation. If no value is specified for this parameter by the program or in an override, this is the default action assumed when the file is opened for output.

***YES** New records are added to the end of the records contained in the save file.

Top

Starting position in file (POSITION)

Specifies the starting position for getting records from the save file. The first record to get is either at the beginning of the file (*START) or at a particular relative record number position in the file (*RRN). This parameter overrides the value specified in the program.

Single values

*START

Get the first record in the file first. If no value is specified for this parameter by the program, or in an override, this is the default action assumed when the file is opened for input.

Element 1: Retrieve order

*RRN The relative record number specified for the second element of this parameter is the first record to get.

Element 2: *RRN—Record number

unsigned-integer

Specify the record number (its position from the beginning of the file) of the record that you get first.

Top

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

*IMMED

The program does not wait. Immediate allocation of file resources is required.

*CLS The default wait time specified in the class description is used as the wait time for the allocation of the file resources.

integer

Specify the number of seconds that the program waits for the file resources to be allocated. Valid values range from 1 through 32767.

Top

Secure from other overrides (SECURE)

Specifies whether this file is protected from the effects of file override commands that were previously called.

*NO This file is not protected from other file overrides; its value is overridden by the effects of any file override commands that were previously called.

*YES This file is protected from the effects of any file override commands that were previously called.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

*ACTGRPDEFN

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

*CALLLVL

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** The ODP is not shared with other programs in the routing step. A new ODP for the file is created and used every time a program opens the file.

***YES** The same ODP is shared with each program in the job that also specifies *YES when it opens the file.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

*ACTGRPDEFN

The scope of the open operation is determined by the activation group of the program that called the OVRSAVF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Examples

```
OVRSAVF FILE(ONLINE) POSITION(*RRN 100) SECURE(*YES)
```

This command overrides the file named ONLINE so that the first record gotten after the file is opened for input is relative record number 100. The file is also safe from overrides (in previous program calls).

Top

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

[Top](#)

Override with Tape File (OVRTAPF)

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

Parameters
Examples
Error messages

The Override with Tape File (OVRTAPF) command is used to (1) override (replace) a file named in a program, (2) override certain attributes of a file that are used by a program, or (3) override the file named in a program and override certain attributes of the file processed. Parameters overridden by this command are specified in the file description, in the program, or in other called file override commands.

If a file named in the program is overridden, the name of that file is specified in the FILE parameter and the name of the overriding file is specified in the TOFILE parameter. The OVRTAPF command can also specify parameters to override values contained in the file description of the overriding file. If the file named in the program is not replaced, but certain parameters of the file are overridden, the name of the file is specified in the FILE parameter and *FILE is specified in the TOFILE parameter. The parameters overridden are then specified by the other parameters of the OVRTAPF command. Parameters that are not specified do not affect the parameters specified in the file description, in the program, or in other called file override commands.

More information on overriding files is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, the Application Display Programming book, SC41-5715, and the Printing category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: Using this command does not cause a file to be overridden immediately. Information provided on this command is stored until the file is used, at which time the file is overridden.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File being overridden	<i>Name</i>	Required, Positional 1
TOFILE	Overriding to tape file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to tape file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DEV	Device	Values (up to 4 repetitions): <i>Name</i>	Optional, Positional 3
VOL	Volume identifier	Single values: *NONE Other values (up to 50 repetitions): <i>Character value</i>	Optional
REELS	Tape reels specifications	<i>Element list</i>	Optional
	Element 1: Label processing type	*SL, *NL, *NS, *BLP, *LTM	
	Element 2: Number of reels	1-255	
SEQNBR	Sequence number	1-16777215, *END, *NEXT	Optional
LABEL	File label	<i>Character value</i>	Optional
RCDLEN	Record length	<i>Integer</i> , *CALC	Optional

Keyword	Description	Choices	Notes
BLKLEN	Block length	1-524288, *CALC	Optional
BUFOFSET	Buffer offset	<i>Integer</i> , *BLKDSC	Optional
RCDBLKFMT	Record block format	*F, *FB, *V, *VB, *D, *DB, *VS, *VBS, *U	Optional
EXTEND	Extend	Single values: *NO Other values: <i>Element list</i>	Optional
	Element 1: Extend file	*YES	
	Element 2: Check file	* NOCHECK , *CHECK	
DENSITY	Tape density	<i>Character value</i> , *DEVTYPE, *CTGTYPE, *FMT3480, *FMT3490E, *FMT3570, *FMT3570E, *FMT3590, *FMT3590E, *QIC120, *QIC525, *QIC1000, *QIC2GB, *QIC2DC, *QIC4GB, *QIC4DC, *QIC3040, *QIC5010, *MLR3, *SLR60, *SLR100, *FMT2GB, *FMT5GB, *FMT7GB, *FMT20GB, *FMT60GB, *ULTRIUM1, 1600, 3200, 6250	Optional
COMPACT	Data compaction	*DEV, *NO	Optional
CODE	Code	*EBCDIC, *ASCII	Optional
CRTDATE	Creation date	<i>Date</i> , *NONE	Optional
EXPDATE	File expiration date	<i>Date</i> , *NONE, *PERM	Optional
ENDOPT	End of tape option	*REWIND, *LEAVE, *UNLOAD	Optional
USRLBLPGM	User label program	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: User label program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
TBL	Conversion table	Single values: *DFT, *NONE, *CCSID Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Conversion table	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
FROMCCSID	From CCSID	1-65533	Optional
TOCCSID	To CCSID	1-65533	Optional
IGCDTA	User specified DBCS data	*NO, *YES	Optional
WAITFILE	Maximum file wait time	<i>Integer</i> , *IMMED, *CLS	Optional
SECURE	Secure from other overrides	*NO, *YES	Optional
OVRSCOPE	Override scope	*ACTGRPDFN, *CALLLVL, *JOB	Optional
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the file being used by the program to which this override command is applied. The specified file must be a tape device file when *FILE is specified in the **Overriding to tape file (TOFILE)** parameter. Otherwise, any device file or database file name can be specified.

This is a required parameter.

name Specify the name of the file.

Top

Overriding to tape file (TOFILE)

Specifies the tape file that is used instead of the file specified in the **File being overridden (FILE)** parameter; or, if *FILE is specified, specifies that certain attributes are overridden by parameters specified in this command. The parameters specified on this command override the other values specified in the tape device file or in the program.

Single values

***FILE** The tape device file named on the FILE parameter has some of its parameters overridden by values specified in this command.

Qualifier 1: Overriding to tape file

name Specify the name of the tape device file that is used instead of the overridden 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 used to locate the tape device file. If no library is specified as the current library for the job, QGPL is used.

name Specify the library where the tape device file is located.

Top

Device (DEV)

Specifies the names of up to four tape devices, one virtual tape device, or one media library device that are used with the tape device file to perform input/output operations. A media library device is a tape storage device that contains one or more tape drives, tape cartridges, and a part (carriage and picker assembly) for moving tape media between the cartridge storage slots and the tape drives. The order in which the device names are specified here is the order in which tapes placed in the devices are processed. Specify the device names (no more than four) that override the device names specified in the program or in the tape device file. When the number of volumes processed exceeds the number of devices in the DEV list, the devices are used in the same order as specified, wrapping around to the first device as needed.

name Specify the name of the tape device or media library device.

Top

Volume identifier (VOL)

Specifies one or more volume identifiers of the tapes that are used by the tape device file. The tapes (volumes) must be written on the devices in the same order as their identifiers are specified here, and in the same order as the device names are specified on the **Device (DEV)** parameter. If the tape file is opened for read backward, the volume identifiers in the list are processed from last to first, while the devices in the device list are used in first to last order.

Single values

***NONE**

No tape volume identifiers are specified for this file. They are supplied before the device file is opened, either in a CHGTAPF or in another Override with Tape File (OVRTAPF) command. If no

volume identifiers are specified before the device file is opened, no volume checking is performed beyond verifying that the correct label type volume is put on the device, and no volume names are provided in operator messages.

Other values (up to 50 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed on the device. Each volume identifier contains a maximum of 6 alphanumeric characters. Use a blank as a separator character when listing multiple identifiers. Up to 50 volume identifiers can be specified. These identifiers are used in messages sent to the operator during processing. The maximum number of reels processed for an *NL, *NS, *BLP, or *LTM input file is determined by the number of volume identifiers in the list.

Note: If the VOL parameter value used for the file specifies a list of identifiers rather than VOL(*NONE), the number-of-reels part of the REELS parameter is ignored regardless of where it is specified. A description of how the parameter values for the file are determined when overrides are used, the high-level language interface, and the device file when the file is opened is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. To ensure that the number-of-reels part of the REELS parameter is used (rather than a VOL identifier list) to control the volumes processed by the tape device file, specify VOL(*NONE) in the same command in which the REELS parameter is specified.

Top

Tape reels specifications (REELS)

Specifies the type of labeling that is used on the tape reels, and the maximum number of reels processed if there is no list of volume identifiers specified on the **Volume identifier (VOL)** parameter and this device file is used with either *NL, *LTM, *NS, or *BLP input files. When the number of reels is specified, the volume identifiers of any labeled volumes placed in the tape device are ignored; instead, the order in which the reels are used must be checked by the operator.

The maximum number of reels specification (the second part of this parameter) is not a limiting value for standard-label input files or output files. For a standard-label input file, the data file labels limit the number of volumes processed by indicating end of file. For an output file, the maximum number of reels value is ignored; the system requests that additional volumes be placed in the device until the file is closed.

The system checks the first record following the load point on the tape to see (1) whether it has exactly 80 bytes for EBCDIC or at least 80 bytes for ASCII and (2) whether the first 4 bytes contain the values VOL and 1. If so, the reel contains a standard-label tape. *SL and *BLP files require standard-label tape volumes. *NL, *NS, and *LTM tape files cannot process standard-label volumes.

Note: The values *SL, *NL, and *LTM can be specified if the device file is used for either reading or writing on tapes. The values *NS and *BLP are valid only if the device file is used to read tapes.

This parameter overrides the values specified in the device file, in the program, or in other called OVRTAPF commands.

Element 1: Label processing type

***SL** The volumes have standard labels. If a list of volume identifiers is specified (with the VOL parameter), the system checks that the correct tape volumes are on the device in the specified sequence.

- If no volume identifier list is given and the file is opened for *output*, any standard-label volumes may be installed on the device.
- If no volume identifier list is given and the file is opened for *input*, the first volume may have any volume identifier, but if the file is continued, the system requires the correct continuation volumes to be processed (verified by checking the data file labels). For an input file, the end-of-file message is sent to the program being used when the labels on the last volume processed indicate that it is the last volume for the data file.

***NL** The volumes are not labeled. On a nonlabeled volume, tape marks are used to indicate the end of each data file and the end of the volume. For an *input* file, the end-of-file message is sent to the program when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***NS** The volumes have nonstandard labels. Each volume must start with some kind of label information, optionally preceded by a tape marker and always followed by a tape marker. This nonstandard label information is ignored. The system spaces forward to a point beyond the tape marker that follows the nonstandard labels and positions the tape at the file's data. Each reel must have a tape marker at the end of the file's data. Information beyond this ending tape marker is ignored. Only a single data file can exist on a nonstandard tape. Standard-label volumes *cannot* be processed by using the *NS label processing.

For an *input* file, the end-of-file message is sent to the program using the file when the number of volumes specified in the volume list have been processed, or, if no list of volume identifiers is provided, when the number of reels specified in the REELS parameter are processed.

***BLP** Standard-label processing is bypassed. Each reel *must* have standard labels. Although each reel is checked for a standard volume label and each file must have at least one standard header label (HDR1) and one standard trailer label (EOV1 or EOF1), most other label information (such as the data file record length or block length) is ignored. The sequence number of each file on the volume is determined only by the number of tape markers between it and the start of tape (in contrast to *SL processing in which the file sequence number stored in the header and trailer labels of each file are used to locate a data file).

Most of the information in the data file trailer label is ignored, but if an end-of-file (EOF) trailer label is found, the end-of-file message is sent to the program using the tape file. If no end-of-file trailer label is encountered by the time the specified number of volumes or reels have been processed (volume identifier list and REELS parameter), the end-of-file message is immediately sent to the program using the tape file. Bypass label processing can be used when the user does not know the name of the file used or when some file label information is incorrect.

***LTM** The volumes have no labels but do have a single leading tape marker before the first data file. REELS(*LTM) is processed the same as REELS(*NL) except that when SEQNBR(1) is specified for an output file to create the first data file on the tape, a leading tape marker is written at the start of the tape before the first data block.

Element 2: Number of reels

1-255 Specify the maximum number of reels that are processed for an *NL, *LTM, *NS, or *BLP input tape operation when there is no list of volume identifiers used on the **Volume identifier (VOL)** parameter. If the next reel is not on the device when the end of the currently-processing tape is reached, a message is sent to the operator requesting that the next tape be installed on the next tape device. The number-of-reels value is ignored for a standard label (*SL) processing file, or for any output file.

Top

Sequence number (SEQNBR)

Specifies the sequence number of the data file on the tape that is processed.

- When standard-label tapes are used, the four-position file sequence number is read from the first header label of the data file.
- When bypass label processing is used, or when standard-label tapes are not used, the system counts the tape marks from the beginning of the tape to locate the data file with the correct sequence number.
- When multiple file, multiple volume tapes are processed using *SL on the **Tape reels specifications (REELS)** parameter, the file sequence numbers continue consecutively through all of the volumes; each new data file has a sequence number that is one greater than the previous file, regardless of which volume contains the file.

1-16777215

Specify the file sequence number that overrides the sequence number specified in the program or device file.

***END** The file sequence number is added to the end of the tape.

An error message is shown on the display when a tape device file is used to read from a tape and the *END special value is specified in the tape device file.

***NEXT**

The next file on the tape is processed. If the tape is currently positioned before the first file, the first file on the tape is processed. This value can only be specified in tape files that are used to read from tape. An error message is issued when a tape file is used to write to a tape and *NEXT is specified in the tape file.

Top

File label (LABEL)

Specifies the identifier of the data file that is processed by this tape device file. A label identifier is *required* for a standard label output file, but is *optional* for an input file.

If a data file identifier is specified for any type of label processing other than *SL, it is ignored.

character-value

Specify the tape data file identifier.

Top

Record length (RCDLEN)

Specifies (in bytes) the length of the records that are contained in the data file that is processed with this device file. The system always uses the record length and block length specified in the data file labels for any standard label input file or output file with *YES specified in the **Extend file (EXTEND)** parameter, if a second header label (HDR2) is found on the tape and *BLP label processing is not specified. This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

***CALC**

No record length is specified for the data file processed. If *CALC is specified, the system attempts to calculate an appropriate record length when the file is opened. *CALC is used for tapes that are not labeled or when there is no HDR2 label if a value other than *CALC is specified in the **Block length (BLKLEN)** parameter for the file, and if the **Record block format (RCDBLKFMNT)** parameter does not specify spanned or blocked records. In this case, the system calculates an appropriate record length from the block length, record block format, and buffer

offset (for an ASCII file) specified for the file. In any other case, the actual record length must be specified by a Change Tape File (CHGTAPF) command or Override with Tape File (OVRTAPF) command, or in the high-level language program that opens the device file.

integer

Specify a value ranging from 1 through 32767 that specifies the length of each record in the data file. The minimum and maximum record lengths that are allowed for a file are dependent on the record block format, block length, buffer offset (for an ASCII file), and recording code.

Top

Block length (BLKLEN)

Specifies (in bytes) the maximum length of the data blocks transferred to or from the tape for input or output operations. The system always uses the block length and record length specified in the data file labels for any standard label input file or output file with *YES specified in the **Extend file (EXTEND)** parameter, if a second header label (HDR2) is found on the tape and *BLP label processing has not been specified.

This parameter overrides the value specified in the device file, in the program, or in other OVRTAPF commands.

*CALC

No block length is specified for the data file processed. If *CALC is specified, the system attempts to calculate an appropriate block length when the file is opened. *CALC can be used for tapes that are not labeled or when there is no HDR2 label if a value other than *CALC is specified in the **Record length (RCDLEN)** parameter for the file, and if the **Record block format (RCDBLKFMT)** parameter does not specify spanned or blocked records. In this case, the system calculates an appropriate block length from the record length, record block format, and buffer offset (for an ASCII file) specified for the file. In any other case, the actual block length must be specified by a Change Tape File (CHGTAPF) command or Override with Tape File (OVRTAPF) command, or in the high-level language program that opens the device file.

1-524288

Specify the maximum length, in bytes, of each block in the data file to be processed. The minimum block length that can be successfully processed is determined by the tape device hardware and AS/400 system machine support functions.

The maximum block length is always 524288 bytes for an input file, but is limited to 9999 bytes if block descriptors must be created for an ASCII output file.

The following table shows the minimum and maximum block length values allowed for an output file:

CODE	BUFOFSET	MIN BLKLEN	MAX BLKLEN
*EBCDIC	Ignored	18	524288
*ASCII	0	18	524288
*ASCII	*BLKDSC	18	9999

Top

Buffer offset (BUFOFSET)

Specifies the buffer offset value for the start of the first record in each block in the tape data file. A buffer offset value is used for any record block format ASCII file, and is ignored for an EBCDIC tape file. The system always uses the buffer offset specified in the data file labels for any standard label input file or output file with *YES specified in the **Extend file (EXTEND)** parameter, if a value is contained in the second header label (HDR2) on the tape, and *BLP label processing is not specified.

The buffer offset parameter specifies the length of any information that precedes the first record in the block. For record block formats *D, *DB, *VS, and *VBS, each record or record segment is preceded by a descriptor that indicates the length of the record or segment. A buffer offset value is used to indicate that there is information ahead of the descriptor word for the first record in each block, or ahead of the data of the first fixed-length record or undefined format record in each block.

This parameter is not needed for a standard-labeled file that is processed for input if the tape includes a second file header label (HDR2) that contains the buffer offset value. A buffer offset must be provided by the Create Tape File (CRTTAPF) command, Change Tape File (CHGTAPF) command, or Override with Tape File (OVRTAPF) command, or by the file labels for an input file that contains any information (such as a block descriptor) ahead of the first record in each block. If you do not specify a buffer offset when a tape file is created, it is not necessary to specify an offset value when the file is read.

The only buffer offset values allowed for an output file are zero and *BLKDSC. An existing standard-labeled data file with a buffer offset value in the HDR2 label is extended only if the offset value is either 0 or 4. An offset value of 0 in the HDR2 label adds data blocks with no buffer offset. *BLKDSC must be specified to extend an existing tape data file that contains an offset value of 4 in the HDR2 label.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

***BLKDSC**

Block descriptors that are 4 bytes in length are created in any tape file that is created using this device file. Any input file read using this device file assumes 4 bytes of buffer offset information preceding the first record in each data block. This value is valid only for a record block format *D or *DB file. The contents of the buffer offset information of each output data block when BUFOFSET(*BLKDSC) is specified is the actual length of the data block, expressed in zoned decimal format.

integer

Specify a value ranging from zero through 99 that specifies the length of the buffer offset information that precedes the first record in each data block.

Top

Record block format (RCDBLKFMF)

Specifies the type of format blocking attribute of records in the tape data file being processed.

Record block format *V and *VB records can be processed only for an EBCDIC file; *D and *DB records can be processed only for an ASCII file. If a standard-label tape (label type *SL or *BLP) is being processed and an inconsistent record block format is specified for the volume code, the correct record type is assumed (V or D) for the volume code and a warning message is sent to the program that opens the file. If the record type and code are inconsistent for a nonlabeled volume (label type *NL, *LTM, or *NS), an error message is sent and the file is *not* opened, because there are no labels to verify the correct volume code.

If a valid record length, block length, and buffer offset value (for an ASCII file) are specified for fixed-length records but the block attribute is incorrect, the correct block attribute is assumed (changing record block format *F to *FB or record block format *FB to *F), and a warning message is sent to the program that opens the file.

If a block length is specified that is longer than required to process a maximum length record, then record block format *V, *D, or *VS is changed to *VB, *DB, or *VBS and a warning message is sent to the program that opens the file.

Note: When BUFOFSET(*BLKDSC) is specified for the file, a value of 4 should be used for the BUFOFSET part of any BLKLEN calculations, unless existing file labels on the tape specify a different value.

This parameter overrides the value specified in the device file, in the program, or in other called OVRTAPF commands.

- *F Fixed length, deblocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *FB, based on other file parameters.
- *FB Fixed length, blocked, unspanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *F, based on other file parameters.
- *V Variable length, deblocked, records in EBCDIC type V format are processed. The system may change this record block format to *VB, *D, or *DB, based on other file parameters.
- *VB Variable length, blocked, unspanned records in EBCDIC type V formats are processed. The system may change this record block format to *DB, based on the volume code.
- *D Variable length, deblocked, unspanned records in ASCII type D formats are processed. The system may change this record block format to *DB, *V, or *VB, based on other file parameters.
- *DB Variable length, blocked, unspanned records in ASCII type D formats are processed. The system may change this record block format to *VB, based on the volume code.
- *VS Variable length, deblocked, spanned records in either EBCDIC or ASCII code are processed. The system may change this record block format to *VBS, based on other file parameters. The representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.
- *VBS Variable length, blocked, spanned records in either EBCDIC or ASCII code are processed. The representation of spanned records on the tape is different for EBCDIC and ASCII files, but the system selects the correct format based on the file code.
- *U Undefined format records in either EBCDIC or ASCII code are processed. Records are processed as variable length records, where each record written or read is in a separate tape block.

Top

Extend file (EXTEND)

Specifies, for output operations to tape, whether new records are added to the end of a data file that is currently on the tape. The specific data file is identified by the **Sequence number (SEQNBR)** parameter and, for a standard-label file, by the **File label (LABEL)** parameter. If the data file is extended, it becomes the last file on the tape volume. Any data files that follow this data file are overwritten as the specified file is extended.

This parameter overrides the extend value specified in the device file, in the program, or in other called OVRTAPF commands.

Single values

- *NO Records are not added to the end of the specified data file. Regardless of whether there is already a data file with the specified sequence number on the tape, a new data file is created that overwrites an existing data file and any files that follow it.

Element 1: Extend file

- *YES New records are added to the end of the specified data file.

Element 2: Check file

***NOCHECK**

The file is extended without being checked to determine whether it is active.

***CHECK**

Before the file is extended, it is checked to determine whether it is active.

Top

Tape density (DENSITY)

Specifies the density of the data that is written on the tape volume when this device file is created. This parameter is used only for tape files being written to tape; it is ignored for tape files being read from the tape (in the case of files being read from tape, the density on the tape is used).

The density of a standard-label volume is specified on the INZTAP command, which initializes tapes as standard-label volumes by writing volume labels on them. If the density specified on this parameter is different than the density of a standard-labeled tape, the tape must be reinitialized to the specified density.

***DEVTYPE**

The highest capacity density or format supported by the tape device will be used.

Device

Highest capacity density or format

3480 *FMT3480

3490E *FMT3490E

3570-Bxx
*FMT3570

3570-Cxx
*FMT3570E

3580-001
*ULTRIUM1

3580-002
*ULTRIUM2

3580-003
*ULTRIUM3

3580-004
*ULTRIUM4

3590-Bxx
*FMT3590

3590-Exx
*FMT3590E

3590-Hxx
*FMT3590H

3592-E05
*FMT3592A2

3592-J1A
*FMT3592A1

4685-001 *VXA2
 5755 *ULTRIUM2
 6258 *DAT72
 6279 *VXA3
 6344 *QIC2GB
 6349 *QIC2GB
 6369 *QIC2GB
 6380 *QIC2GB
 6381 *QIC2DC
 6382 *QIC4DC
 6383 *QIC5010
 6384 *SLR60
 6386 *MLR3
 6387 *SLR100
 6390 *FMT7GB
 63B0 *VRT256K
 7207-122 *QIC4DC
 7208-002 *FMT2GB
 7208-012 *FMT5GB
 7208-222 *FMT7GB
 7208-342 *FMT20GB
 7208-345 *FMT60GB
 9348 6250

***CTGTYPE**

The highest capacity density or format supported by the device for the mounted cartridge type will be used. If the device does not support special cartridge type information, *DEVTYPE is used.

character-value

Specify the density or format to use.

- 1600 The data density on the tape volume is 1,600 bits per inch, which is used for 1/2 inch reel tapes.
- 3200 The data density on the tape volume is 3,200 bits per inch, which is used for 1/2 inch reel tapes.

- 6250** The data density on the tape volume is 6,250 bits per inch, which is used for 1/2 inch reel tapes.
- *DAT72**
The format of this tape is DAT72. It is used by 4mm cartridge tape devices that can store 36 gigabytes of data on a standard length cartridge.
- *DDS3**
The format of this tape is DDS3. It is used by 4mm cartridge tape devices that can store 12 gigabytes of data on a standard length cartridge.
- *DDS4**
The format of this tape is DDS4. It is used by 4mm cartridge tape devices that can store 20 gigabytes of data on a standard length cartridge.
- *FMT3480**
The format of this tape is FMT3480. The data density on this tape volume is formatted to support a 3480 device. This density is used for 1/2 inch cartridge tapes.
- *FMT3490E**
The format of this tape is FMT3490E. The data density on this tape volume is formatted to support a 3490E device. This density is used for 1/2 inch cartridge tapes.
- *FMT3570**
The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.
- *FMT3570E**
The format of this tape is FMT3570E. The data format is written on the tape volume with a 3570E device.
- *FMT3590**
The format of this tape is FMT3590. The data format is written on the tape volume with a 3590 device. This density is used for 1/2 inch cartridge tapes.
- *FMT3590E**
The format of this tape is FMT3590E. The data format is written on the tape volume with a 3590E device. This density is used for 1/2 inch cartridge tapes.
- *FMT3590H**
The format of this tape is FMT3590H. The data format is written on the tape volume with a 3590H device. This density is used for 1/2 inch cartridge tapes.
- *FMT3592A1**
The format of this tape is FMT3592A1. It is used by 3592 tape devices that can store 300 gigabytes of data on a standard length cartridge.
- FMT3592A1E**
The format of this tape is FMT3592A1E. It is used by 3592 tape devices that can store 300 gigabytes of encrypted data on a standard length cartridge.
- *FMT3592A2**
The format of this tape is FMT3592A2. It is used by 3592 tape devices that can store 500 gigabytes of data on a standard length cartridge.
- FMT3592A2E**
The format of this tape is FMT3592A2E. It is used by 3592 tape devices that can store 500 gigabytes of encrypted data on a standard length cartridge.
- *QIC120**
The format of this tape is QIC120, which is used for 1/4 inch cartridge tapes that can hold 120 megabytes of data.

***QIC525**

The format of this tape is QIC525, which is used for 1/4 inch cartridge tapes that can hold 525 megabytes of data.

***QIC1000**

The format of this tape is QIC1000, which is used for 1/4 inch cartridge tapes that can hold 1200 megabytes of data.

***QIC2GB**

The format of this tape is QIC2GB. It is used by 1/4 inch tape devices which can store 2.5 gigabytes of data on a standard length QIC2GB cartridge.

***QIC2DC**

The format of this tape is QIC2DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC2GB format.

***QIC4GB**

The format of this tape is QIC4GB. It is used by 1/4 inch tape devices which can store 4 gigabytes of data on a standard length QIC4GB cartridge.

***QIC4DC**

The format of this tape is QIC4DC. It is used to write compacted data to a 1/4 inch cartridge that supports the QIC4GB format.

***QIC3040**

The format of this tape is QIC3040, which is used for 1/4 inch minicartridge tapes that can hold 840 megabytes of data.

***QIC5010**

The format of this tape is QIC5010, which is used for 1/4 inch cartridge tapes that can hold 13.5 gigabytes of data.

***MLR3**

The format of this tape is MLR3. It is used by 1/4 inch tape devices which can store 25 gigabytes of data on a standard length MLR3 cartridge.

***SLR60**

The format of this tape is SLR60. It is used by 1/4 inch tape devices which can typically store 60 gigabytes of compacted data on a standard length SLR60 cartridge.

***SLR100**

The format of this tape is SLR100. It is used by 1/4 inch tape devices which can typically store 100 gigabytes of compacted data on a standard length SLR100 cartridge.

***FMT2GB**

The format of this tape is FMT2GB, which is used for 8 millimeter cartridge tapes that can hold 2 gigabytes of data.

***FMT5GB**

The format of this tape is FMT5GB, which is used for 8 millimeter cartridge tapes that can hold 5 gigabytes of data.

***FMT7GB**

The format of this tape is FMT7GB, which is used for 8 millimeter cartridge tapes that can hold 7 gigabytes of data.

***FMT20GB**

The format of this tape is FMT20GB. It is used by 8 millimeter tape devices that can store 20 gigabytes of data on a standard length cartridge.

***FMT60GB**

The format of this tape is FMT60GB. It is used by 8 millimeter tape devices that can store 60 gigabytes of data on a standard length cartridge.

***ULTRIUM1**

The format of this tape is ULTRIUM1. It is used by 1/2 inch cartridge tape devices that can store 100 gigabytes of data on a standard length cartridge.

***ULTRIUM2**

The format of this tape is ULTRIUM2. It is used by 1/2 inch cartridge tape devices that can store 200 gigabytes of data on a standard length cartridge.

***ULTRIUM3**

The format of this tape is ULTRIUM3. It is used by 1/2 inch cartridge tape devices that can store 400 gigabytes of data on a standard length cartridge.

***ULTRIUM4**

The format of this tape is ULTRIUM4. It is used by 1/2 inch cartridge tape devices that can store 800 gigabytes of data on a standard length cartridge.

***VRT32K**

The format of the volume is VRT32K. It is used to write data to a virtual volume using a maximum data block size of 32KB. Volumes written using this format can be duplicated to all supported tape devices.

***VRT64K**

The format of the volume is VRT64K. It is used to write data to a virtual volume using a maximum data block size of 64KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 64KB or greater.

***VRT240K**

The format of the volume is VRT240K. It is used to write data to a virtual volume using a maximum data block size of 240KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 240KB or greater.

***VRT256K**

The format of the volume is VRT256K. It is used to write data to a virtual volume using a maximum data block size of 256KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 256KB or greater.

***VXA1**

The format of this tape is VXA1. It is used by VXA cartridge tape devices that can store 33 gigabytes of data on a standard length cartridge.

***VXA2**

The format of this tape is VXA2. It is used by VXA cartridge tape devices that can store 80 gigabytes of data on a standard length cartridge.

***VXA3**

The format of this tape is VXA3. It is used by VXA cartridge tape devices that can store 160 gigabytes of data on a standard length cartridge.

Note: Self-configured tape devices may define additional valid values for the density parameter. Use System i5 Navigator (Configuration and Service) (Hardware) (Tape Devices) (Tape Libraries) (Tape Resources) (Properties) or (Configuration and Service)(Hardware) (Tape Devices) (Stand-Alone Devices) (Properties) to find additional valid density values for a specific device, or use the F4=Prompt key on the "Tape density" field of the CL command to see a list of all valid density values for the attached tape devices.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed. If the tape devices being used do not support data compaction, this parameter will be ignored when the file is opened.

This parameter overrides the value specified in the device file, in the program or in other called OVRTAPF commands.

*DEV D

Device data compaction is performed if the devices being used support data compaction.

*NO Device data compaction is not performed.

Top

Code (CODE)

Specifies the type of character code that is used by the tape device file when the system is reading or writing tape data.

*EBCDIC

The EBCDIC character code is used with this tape device file.

*ASCII

The ASCII character code is used.

Note: For standard labeled (*SL) tapes the CODE parameter is used to determine how the labels are processed. For all label types the TBL, FROMCCSID, and TOCCSID parameters control what conversion, if any, is used for the data portion of the files.

Top

Creation date (CRTDATE)

Specifies, for tape input data files and for tape output for which *YES is specified in the **Extend file (EXTEND)** parameter, the date when the data file was written to tape. The creation date of the data file is stored in file labels on the tape. If a creation date is specified for any type of label processing other than *SL, it is ignored.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

*NONE

The creation date of the data file is not checked.

date Specify the creation date of the data file. The date must be specified in job-date format.

Top

File expiration date (EXPDATE)

Specifies, for tape output data files, the expiration date of the data file used by this device file. The data file expiration date is stored in file labels on the tape. If an expiration date is specified for any type of label processing other than *SL, it is ignored. If a date is specified, the data file is protected and cannot be overwritten until the specified expiration date.

This parameter overrides the value specified in the program, device file, or in other called OVRTAPF commands.

***NONE**

No expiration date for the data file is specified; the file is not protected. An expiration date is written in the data file labels so the file can be used as a scratch data file.

***PERM**

The data file is protected permanently. The date written in the tape data file is 999999.

date Specify the date on which the data file expires. The date must be specified in job-date format.

Top

End of tape option (ENDOPT)

Specifies the operation that is automatically performed on the tape volume after the operation ends. If more than one volume is included, this parameter applies only to the last tape volume used; all other tape volumes are rewound and unloaded when the end of the tape is reached.

***REWIND**

The tape is rewound, but not unloaded.

***UNLOAD**

The tape is automatically rewound and unloaded after the operation ends.

***LEAVE**

The tape does not rewind or unload after the operation ends. It remains at the current position on the tape drive.

Top

User label program (USRLBLPGM)

Specifies that a program processes user header and trailer labels on a standard-labeled tape. This parameter is valid only when *SL is specified in the **Tape reels specifications (REELS)** parameter.

Single values

***NONE**

Only standard label processing is used. No program is called to process user labels.

Qualifier 1: User label program

name Specify the name of the program that is called to process user labels.

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

Conversion table (TBL)

Specifies the qualified name of a conversion table to be used for single-byte conversion of input files or output files. The specified conversion is only used for the data portion of the files. When the specified code is *ASCII (CODE parameter) any labels will be converted between ISO/ASCII 8-Bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the labels, if any, are not converted.

Note: See system supplied conversion tables QSYS/QASCII and QSYS/QEBCDIC for an example of the conversion used to translate between ISO/ASCII 8-Bit code and EBCDIC.

Single values

***DFT** When the specified code is *ASCII (CODE parameter) the data and labels will be converted between ISO/ASCII 8-bit code and EBCDIC. When the specified code is *EBCDIC (CODE parameter) the data and labels will not be converted.

***NONE**
The data will not be converted.

***CCSID**
The CCSID parameters are used to generate a conversion table to use for converting the data portion of the files.

Qualifier 1: Conversion table

name Specify the name of a conversion table to be used for conversion of the data between single-byte character sets.

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 name of the library to be searched.

Top

From CCSID (FROMCCSID)

Specifies a single-byte CCSID used for the input data. The input data is the data read from the tape for input operations, or read from a file for output operations.

1-65533
The requested CCSID value is used. The value is validated to ensure that a single-byte CCSID is specified.

Top

To CCSID (TOCCSID)

Specifies the single-byte CCSID used for the output data. The output data is the data written to the tape for output operations, or written to a file for input operations.

1-65533
The requested CCSID value is used. The value is validated to ensure that a single-byte CCSID is specified.

User specified DBCS data (IGCDTA)

Specifies whether the file processes double-byte character set (DBCS) data.

- ***YES** For program described files, indicates that the file is used to process DBCS data. For field level files, the fields that are defined with the ALTTYP keyword are changed from ALPHA fields or character fields to DBCS fields.
- ***NO** For program described files, indicates that the file is not used to process DBCS data. For field level files, the fields that are defined with the ALTTYP keyword remain ALPHA fields or character fields. The file attribute remains at whatever it was set to when the file was created.

Top

Maximum file wait time (WAITFILE)

Specifies the number of seconds that the program waits for the file resources to be allocated when the file is opened, or the device or session resources to be allocated when an acquire operation is performed to the file. If the file resources cannot be allocated in the specified wait time, an error message is sent to the program.

***IMMED**

The program does not wait. Immediate allocation of file resources is required.

- ***CLS** The default wait time specified in the class description is used as the wait time for the file resources to be allocated.

integer

Specify the number of seconds that the program waits for the file resources to be allocated to the tape file when the file is opened, or the wait time for the device allocated when an acquire operation is performed to the file. Valid values range from 1 through 32767 seconds.

Top

Secure from other overrides (SECURE)

Specifies whether this file is safe from the effects of file override commands started in previously called programs.

- ***NO** This file is not protected from other file overrides; its values are overridden by the effects of any file override commands started in previously called programs.
- ***YES** This file is protected from the effects of any file override commands started in previously called programs.

Top

Override scope (OVRSCOPE)

Specifies the extent of influence (scope) of the override.

***ACTGRPDEFN**

The scope of the override is determined by the activation group of the program that calls this command. When the activation group is the default activation group, the scope equals the call

level of the calling program. When the activation group is not the default activation group, the scope equals the activation group of the calling program.

***CALLLVL**

The scope of the override is determined by the current call level. All open operations done at a call level that is the same as or higher than the current call level are influenced by this override.

***JOB** The scope of the override is the job in which the override occurs.

Top

Share open data path (SHARE)

Specifies whether the open data path (ODP) is shared with other programs in the same routing step. When an ODP is shared, the programs accessing the file share facilities such as the file status and the buffer.

***NO** An ODP created for this file open operation is not shared. Every time a program opens the file, a new ODP to the file is created and started.

***YES** If the file is opened more than once, the same ODP is shared with each program in the routing step that also specifies *YES for this parameter when it opens the file. This includes multiple open operations in the same program.

Note: When SHARE(*YES) is specified and control is passed to a program, a read operation in that program retrieves the next input record. A write operation produces the next output record.

Top

Open scope (OPNSCOPE)

Specifies the extent of influence (scope) of the open operation.

***ACTGRPDFN**

The scope of the open operation is determined by the activation group of the program that called the OVRTAPF command processing program. If the activation group is the default activation group, the scope is the call level of the caller. If the activation group is a non-default activation group, the scope is the activation group of the caller.

***JOB** The scope of the open operation is the job in which the open operation occurs.

Top

Examples

Example 1: Overriding a File

```
OVRTAPF FILE(OUT) VOL(DPT706) LABEL(STATUSR)
```

This command overrides a file named OUT in the program using the data file STATUSR on tape volume DPT706.

Example 2: Allowing DBCS Data

```
OVRTAPF FILE(IGCLIB/IGCTAP) IGCDA(*YES)
```

This command overrides the tape device file named IGCTAP, which is stored in the library IGCLIB, so the file may contain double-byte character set data.

Example 3: Using Data Density of 1600 Bits Per Inch

```
OVRTAPF FILE(OUT) DENSITY(1600)
```

This command overrides a file named OUT to use a data density of 1600 bits per inch when writing to the tape volume.

Example 4: Using a Conversion Table to Process a Tape with EBCDIC Labels.

```
OVRTAPF FILE(FILE1) REELS(*SL) CODE(*EBCDIC)
        TBL(LIB1/TABLE1)
```

This command overrides a tape device file named FILE1 to specify that a conversion table named LIB1/TABLE1 is to be used to convert all data read from, or written to, the tape volume.

Example 5: Using Specified CCSIDs to Process a Non-labeled Tape.

```
OVRTAPF FILE(FILE2) REELS(*NL) TBL(*CCSID)
        FROMCCSID(819) TOCCSID(37)
```

This command overrides a tape device file named FILE2 to specify that any data read from, or written to, the tape volume is to be converted from CCSID 819 to CCSID 37.

[Top](#)

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

[Top](#)

Parameter Definition (PARM)

Parameters
Examples
Error messages

The Parameter (PARM) command definition statement defines a parameter of a command being created. A parameter is the means by which a value is passed to the command processing program. One PARM statement must be used for each parameter that appears in the command being defined. The order in which the PARM statements are entered into the source file determines the order in which the parameters must be specified when the command is entered in positional form and the order in which they are passed to the validity checker and the command processing program. A maximum of 99 parameters can be defined for one command. A command with a large number of parameters will require more processing time before the command processing program is called, regardless of how many parameters are actually coded.

Note: The PARM statement contains certain parameters and predefined values that can be used only when IBM-supplied command processing programs are called by the command being defined. Because of limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. These parameters and values are identified by the phrase (*For IBM-supplied commands*) that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.

Top

Parameters

Keyword	Description	Choices	Notes
KWD	Keyword	<i>Simple name</i>	Required, Positional 1
TYPE	Type of value	<i>Simple name</i> , *DEC, *LGL, *CHAR, *INT2, *INT4, *NAME, *GENERIC, *VARNAME, *DATE, *TIME, *CMD, *X, *HEX, *ZEROELEM, *NULL, *CMDSTR, *PNAME, *UINT2, *UINT4, *SNAME, *CNAME	Required, Positional 2
LEN	Value length	Values (up to 3 repetitions): <i>Integer</i>	Optional, Positional 3
RTNVAL	Return value	* <u>NO</u> , *YES	Optional
CONSTANT	Constant value	<i>Character value</i>	Optional
RSTD	Restricted values	* <u>NO</u> , *YES	Optional
DFT	Default value	<i>Character value</i>	Optional
VALUES	Valid values	Values (up to 300 repetitions): <i>Character value</i>	Optional
REL	Relational expression	<i>Element list</i>	Optional
	Element 1: Relational operator	*GT, *EQ, *GE, *NL, *LT, *NE, *LE, *NG	
	Element 2: Value or keyword reference	<i>Character value</i>	
RANGE	Range of values	<i>Element list</i>	Optional
	Element 1: Lower value or keyword ref	<i>Character value</i>	
	Element 2: Upper value or keyword ref	<i>Character value</i>	

Keyword	Description	Choices	Notes
SPCVAL	Special values	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: From value	<i>Character value</i>	
	Element 2: To replacement value	<i>Character value</i>	
SNGVAL	Single values	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: From value	<i>Character value</i>	
	Element 2: To replacement value	<i>Character value</i>	
MIN	Minimum values required	0-300, <u>0</u>	Optional
MAX	Maximum values allowed	<i>Integer</i> , <u>1</u>	Optional
ALWUNPRT	Allow unprintable characters	<u>*YES</u> , *NO	Optional
ALWVAR	Allow variable names	<u>*YES</u> , *NO	Optional
PGM	Is PARM a program name	<u>*NO</u> , *YES	Optional
DTAARA	Is PARM a data area name	<u>*NO</u> , *YES	Optional
FILE	If a file parameter, how used	<u>*NO</u> , *IN, *OUT, *UPD, *INOUT, *UNSPFD	Optional
FULL	Full field required	<u>*NO</u> , *YES	Optional
EXPR	Value an expression	<u>*NO</u> , *YES	Optional
VARY	Varying length	Single values: <u>*NO</u> Other values: <i>Element list</i>	Optional
	Element 1: Return length value	*YES	
	Element 2: Value length	<u>*INT2</u> , *INT4	
PASSATR	Pass attribute byte	<u>*NO</u> , *YES	Optional
PASSVAL	Value to pass if unspecified	<u>*DFT</u> , *NULL	Optional
CASE	Case of value	<u>*MONO</u> , *MIXED	Optional
CCSID	CCSID of value	<u>*JOB</u> , *UTF16	Optional
LISTDSPL	List displacement	<u>*INT2</u> , *INT4	Optional
DSPINPUT	Display input	<u>*YES</u> , *PROMPT, *NO	Optional
CHOICE	Choice text	<i>Character value</i> , <u>*VALUES</u> , *NONE, *PGM	Optional
CHOICEPGM	Choice program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Choice program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
PMTCTL	Prompt control	<i>Simple name</i> , <u>*NONE</u> , *PMTRQS	Optional
PMTCTLPGM	Prompt control program	Single values: <u>*NONE</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Prompt control program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
KEYPARM	Key parameter	<u>*NO</u> , *YES	Optional
INLPMTLEN	Initial prompt length	<u>*CALC</u> , *PWD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 25, 32, 50, 80, 132, 256, 512	Optional
PROMPT	Prompt specifications	Single values: <u>*NONE</u> Other values: <i>Element list</i>	Optional
	Element 1: Prompt text or message ID	<i>Character value</i>	
	Element 2: Order prompt is displayed	<i>Integer</i>	

Keyword (KWD)

Specifies the keyword name of the parameter being defined. Every CL command parameter has an associated keyword name. Command parameters specified using the parameter keyword can be specified in any order. Parameters may be specified in positional form (without a keyword name) up to the positional limit specified by the MAXPOS parameter on the Create Command (CRTCMD) command.

simple-name

Specify a keyword name for the parameter. The name can be up to ten alphanumeric characters. The first character must be alphabetic.

Type of value (TYPE)

Specifies the type of the value that can be specified for the parameter named in **Keyword (KWD)**. The value can be an integer, a decimal, hexadecimal, or logical value, or a character string (optionally enclosed in apostrophes) that can be a name, date, or time. The value can also be a command. Type one of the following options to specify the parameter type:

***DEC** The parameter value is a packed decimal number.

***LGL** The parameter value is a logical value of one ('1') or zero ('0').

***CHAR**

The parameter value is a character string that can optionally be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it *must* be enclosed in apostrophes. The maximum length of the character string is 5000 bytes if *JOB is specified for the CCSID parameter, or 10000 bytes if *UTF16 is specified for the CCSID parameter.

***NAME**

The parameter value is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters, \$, @, or #. The remaining characters can be alphanumeric, a period, an underscore, or one of special characters, \$, @, or #. The name can also be a string of characters starting and ending with double quotation marks (") or enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it should be specified on the **Special values (SPCVAL)** parameter.

***SNAME**

The parameter value is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters \$, @, or #. The remaining characters can be alphanumeric, an underscore, or one of the special characters \$, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the SPCVAL parameter.

***CNAME**

The parameter value is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters, \$, @, or #. The remaining characters can be alphanumeric or one of special characters, \$, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the SPCVAL parameter.

***PNAME**

The parameter value is a character string that represents a path name string. Optionally the path name string may be enclosed in apostrophes. If the path name string contains any special characters (not including an asterisk (*)), it must be enclosed in apostrophes. The maximum

length of the path name string is 5000 bytes if *JOB is specified for the CCSID parameter, or 10000 bytes if *UTF16 is specified for the CCSID parameter.

*GENERIC

The parameter value is a character string that represents a generic name. A generic name contains one or more characters followed by an asterisk (*) and must conform to the rules for generic names. The name identifies a group of objects whose names all begin with the characters preceding the asterisk (*). If an asterisk (*) is not included, the system assumes that the generic name is a complete object name.

*CMDSTR

The parameter value is a command that will be checked for validity by the system. It is passed to the command processing program as a command string.

The command analyzer rebuilds the command string when it checks it for validity. When the command is rebuilt, keywords are added to parameters that were specified positionally, parameters can be reordered, and parameters that contain characters that cannot be printed (X'FF' and X'00 - X'3F') are converted to hexadecimal notation. As a result, the rebuilt command string may be substantially longer than the original command string. If the length of the rebuilt command is longer than the allowed length specified with the LEN keyword, the command will fail.

Note: Selective prompting is not allowed with the *CMDSTR parameter.

*DATE

The parameter value is a character string that represents a date. When entering the command, the year may be specified with either 2 digits or 4 digits. If a 2-digit year is specified, the date is assumed to be in the range of January 1, 1940 through December 31, 2039. If a 4-digit year is specified, the date may be in the range of August 24, 1928 through May 9, 2071. When it is passed to the command processing program, it is always passed in the format *Cyyymmdd*, where C = century, yy = year, mm = month, and dd = day. The century digit is set to 0 (zero) for years 19xx, and it is set to 1 (one) for years 20xx. When a date value is specified in this PARM statement, it must be specified in one of the following formats *mmddy*, *mmddyyyy*, or *Cyyymmdd*. When a user types a date in the command at run time, it must be specified in the job-date format. The job date separator may be used when the date is entered. If the separator character is used, the date must be enclosed in apostrophes.

*TIME

The parameter value is a character string that represents a time. It is passed to the command processing program in a 6-byte character string as *hhmmss.*, where hh = hours, mm = minutes, and ss = seconds. Values specified in this statement must be in the format *hhmmss*. When a user types a time in the command at run time, it must be specified in the format *hhmmss*. The job time separator may be used when the time is entered. If the separator character is used, the time must be enclosed in apostrophes.

***HEX** The parameter value is hexadecimal in form. The specified characters must be 0 through F. They are converted to hexadecimal (EBCDIC) characters (2 hex digits per byte), right-justified, and padded on the left with zeros. If the value is enclosed in apostrophes, an even number of digits is required. If the value is not enclosed in apostrophes, an even number of digits is *not* required.

*ZEROELEM

The parameter is always considered as a list of zero elements, for which no value can be specified in the command. It is used to prevent a value from being entered for a parameter that is a list even though the command processing program expects one. For example, if two commands use the same command processing program, one command could pass a list for a parameter and the other command may not have any values to pass. The second command would be coded with *ZEROELEM specified for this parameter.

***X** (For IBM-supplied commands) The parameter value is a character string, variable name, or

numeric value. The value is passed as a numeric value if it contains only digits, a + or - sign, or a decimal point; otherwise, it is passed as a character string.

***INT2** The parameter value is an integer that is passed as a 2-byte signed binary number.

***INT4** The parameter value is an integer that is passed as a 4-byte signed binary number.

***UINT2**

The parameter value is an integer that is passed as a 2-byte unsigned binary number.

***UINT4**

The parameter value is an integer that is passed as a 4-byte unsigned binary number.

***VARNAME**

(For IBM-supplied commands) The parameter value is a CL variable name that is passed as a character string.

***CMD** (For IBM-supplied commands) The parameter value is a command. For example, the IF command has a parameter called THEN whose value must be another command. The command is checked for validity by the system.

***NULL**

The parameter value is a null pointer, which can be used as a constant place-holder. A DEP statement or the REL and RANGE keywords of other PARM statements may not refer to the value of a parameter defined with *NULL specified for this parameter.

statement-label

Specify a qualified name or a mixed list of values. The statement label specified here is the statement label that identifies the first of a series of QUAL or ELEM statements that further describe the qualified name or the mixed list being defined by this PARM statement.

Top

Value length (LEN)

Specifies the length of the parameter value that is passed to the command processing program (CPP). Depending on the value specified for the **Type of value (TYPE)** parameter, the LEN parameter may allow up to three length values to be specified.

- If *INT2, *INT4, *UINT2, *UINT4, *DATE, *TIME, *CMD, *ZEROELEM, *NULL, or a statement label is specified for the TYPE parameter, the LEN parameter is not allowed.
- If *DEC is specified for the TYPE parameter, LEN value 1 specifies the total number of digits in the value and LEN value 2 specifies the number of allowable decimal digits to the right of the decimal point. If LEN value 2 is not specified, a value of zero is assumed.
- If *HEX is specified for the TYPE parameter, only LEN value 1 can be specified. This length specifies the number of **bytes** passed after the hexadecimal digits have been converted to character digits. Because 2 hexadecimal digits are converted to 1 byte, the number of hexadecimal digits allowed is twice the value of LEN value 1.
- If *X is specified for the TYPE parameter, the LEN parameter is used as follows:
 - For character data, LEN value 1 specifies the minimum length to be passed. If a longer value is entered, the entire value is passed.
 - For decimal data, LEN values 2 and 3 specify the length and decimal positions for a constant value. If a decimal CL variable is entered, it is passed according to the variable's attributes.
 - For a logical value, LEN value 1 specifies the length of the value, which is always 1.
- If TYPE is other than *DEC, *HEX, or *X, LEN value 1 specifies the maximum length of the string passed to the CPP, and LEN values 2 and 3 cannot be specified.
 - If *PNAME or *CHAR is specified for the TYPE parameter *and* *UTF16 is specified for the **CCSID of value (CCSID)** parameter, the number of bytes passed to the CPP will be *twice* the number specified

for LEN value 1. In UTF16 format, most characters require two bytes, so LEN value 1 will be approximately the maximum number of UTF16 characters allowed. If the UTF16 string is less than two times the value specified for LEN value 1, the value passed to the CPP will be padded on the right with UTF16 blank characters.

See the description of the CCSID and **Varying length (VARY)** parameters for more information related to handling values in UTF16 form.

- Otherwise, the value specified for LEN value 1 will be the number of **bytes** passed to the command processing program. Shorter values will be padded on the right with blanks before being passed to the CPP.

If the LEN parameter is allowed but not specified, a default length will be used based on the value specified for the TYPE parameter. The following table shows the default length value for each TYPE and the maximum value that can be specified for the LEN parameter.

TYPE	Default Length	Maximum Length
*DEC	(15 5)	(24 9)
*LGL	1	1
*CHAR	32	5000
*NAME	10	256
*GENERIC	10	256
*SNAME	10	256
*CNAME	10	256
*PNAME	32	5000
*HEX	1	256
*X	(1 15 5)	(256 24 9)
*VARNAME	11	11
*CMDSTR	256	20000

Top

Return value (RTNVAL)

Specifies whether a value is returned by the command processing program through the parameter being defined in this PARM statement.

- *NO** No value can be returned in the parameter being defined. The parameter is an input parameter only.
- *YES** A value is to be returned by the command processing program in the parameter. A CL variable name must be specified (on the CALL command) to receive the value. *YES is valid only if *DEC, *CHAR, *LGL, *INT2, *INT4, *UINT2, *UINT4, or *X is specified for the **Type of value (TYPE)** parameter. Also, *YES is valid only on commands that are limited to CL programs. That is, if either *BPGM or *IPGM is specified in the Create Command (CRTCMD) command that uses the source file containing this PARM statement, *YES can be specified here. *YES must be specified on the **Varying length (VARY)** parameter, if *YES is specified here and on the **Pass attribute byte (PASSATR)** parameter. If *NO is specified for the **Allow variable names (ALWVAR)** parameter,

or if the **Maximum values allowed (MAX)** parameter has a value of greater than 1, *YES is not valid. *YES is also not valid with the following parameters:

- Constant value (CONSTANT parameter)
- Default value (DFT parameter)
- Restricted values (RSTD parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- If a file parameter, how used (FILE parameter)
- Full field required (FULL parameter)
- Value an expression (EXPR parameter)
- CCSID of value (CCSID parameter)

Top

Constant value (CONSTANT)

Specifies that a value is passed to the command processing program as a constant when the command being defined is processed; the parameter does not appear externally on the command. The value specified in this parameter (if any) must satisfy the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Full field required (FULL parameter)

If a character constant is specified in this parameter, it can be no longer than 32 bytes.

If a constant is specified for the parameter being defined, no prompt text can be specified for the **Prompt specifications (PROMPT)** parameter because the parameter will not be prompted.

This parameter is not valid for the following:

- *CMD, *NULL, or *ZEROELEM specified for the **Type of value (TYPE)** parameter
- A value greater than 1 specified for the **Maximum values allowed (MAX)** parameter
- The **Default value (DFT)** parameter
- *YES specified for the **Return value (RTNVAL)** parameter
- *YES specified for the **Value an expression (EXPR)** parameter

Variables cannot be coded for this parameter.

Top

Restricted values (RSTD)

Specifies whether the value entered for the parameter (specified in the PARM statement) is restricted to only one of the values given in the **Valid values (VALUES)** parameter, the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter, or whether the value can be any value that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- Full field required (FULL parameter)

***NO** The value entered for the parameter specified by the **Keyword (KWD)** parameter can be anything that matches the requirement specified by the following parameters in this PARM statement:

- Type of value (TYPE parameter)
- Value length (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)
- Single values (SNGVAL parameter)
- Full field required (FULL parameter)

***YES** The value entered for the parameter specified by KWD in this PARM statement is restricted to one of the values in the **Valid values (VALUES)** parameter, or to one of the from-values in either the **Special values (SPCVAL)** parameter or the **Single values (SNGVAL)** parameter. *YES cannot be specified if a statement label, *CMD, *NULL, or *ZEROELEM is specified for the **Type of value (TYPE)** parameter, or if *YES is specified for the **Return value (RTNVAL)** parameter.

Top

Default value (DFT)

Specifies the default value that is assigned to the parameter if a value is not specified by the user. That is, the default value is used as the value of the parameter if the user omits the parameter while entering the command or if the user specifies *N as the parameter value. The default value must satisfy one of the following:

- It must match the requirements specified by the following parameters.
 - Type of value (TYPE parameter)
 - Value length (LEN parameter)
 - Relational expression (REL parameter)
 - Range of values (RANGE parameter)
 - Full field required (FULL parameter)
- It must be one of the from-values in the **Special values (SPCVAL)** parameter, or the **Single values (SNGVAL)** parameter.
- If the default is a character constant, it can have no more than 32 bytes.
- If *YES is specified on the **Restricted values (RSTD)** parameter, it must be in the list of values in the **Valid values (VALUES)** parameter, or in the list of from-values of either the **Special values (SPCVAL)** parameter or the **Single values (SNGVAL)** parameter.

- It must be a from-value on the **Single values (SNGVAL)** parameter if the parameter being defined is a list of unlike values or it is a qualified name. This is true when a statement label is specified for **Type of value (TYPE)** parameter; the label is used to identify a QUAL or ELEM statement.

This parameter is not valid if the **Constant value (CONSTANT)** parameter is specified. This parameter is valid only if 0 is specified for the **Minimum values required (MIN)** parameter, which means the parameter named in the **Keyword (KWD)** parameter is optional. No default can be specified if RTNVAL(*YES) is specified on the **Return value (RTNVAL)** parameter; instead, a null pointer is passed for the default. A default cannot be specified if *CMD, *ZEROELEM, or *NULL is specified on the **Type of value (TYPE)** parameter. If *VARNAME is specified on the **Type of value (TYPE)** parameter, a default special value can be specified; a default variable name cannot be specified.

An *assumed* default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified, it is displayed by the prompt exactly as specified.

value Specify the default value that meets the specified requirements or that is one of the values specified in the **Valid values (VALUES)** parameter, the **Special values (SPCVL)** parameter, or the **Single values (SNGVAL)** parameter.

Variables cannot be coded for this value.

Top

Valid values (VALUES)

Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the parameter named on the **Keyword (KWD)** parameter. This parameter is valid only if all of the following are true:

- *YES is specified on the **Restricted values (RSTD)** parameter.
- Both the **Range of values (RANGE)** parameter and the **Relational expression (REL)** parameter are *not* specified.
- Each constant matches the attributes specified by the following parameters.
 - Type of value (TYPE parameter)
 - Value length (LEN parameter)
 - Full field required (FULL parameter)

Character constants specified in this parameter can be no longer than 32 bytes. Type the constants (not more than 300) that can be specified as the value of the parameter. This parameter is not valid if *CMD, *CMDSTR, *X, *NULL, statement label, *VARNAME, or *ZEROELEM is specified on the **Type of value (TYPE)** parameter, or if *YES is specified on the **Return value (RTNVAL)** parameter.

Top

Relational expression (REL)

Specifies the relationship between the parameter value of this parameter and the value of a constant or another parameter. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that it is the value of the keyword that is to be tested. The value associated with the referred to keyword is the value passed to the command processing program, not the user-specified value. If the relationship is with another parameter whose value is a list of values or a qualified name, the first value only is used in the comparison.

To specify the relationship, enter one of the following relational operators followed by either a constant or the keyword name of the other parameter (which must be preceded by an &).

- *LT less than
- *LE less than or equal to
- *EQ equal to
- *GE greater than or equal to
- *GT greater than
- *NL not less than
- *NE not equal to
- *NG not greater than

The REL parameter is not valid in the following cases:

- If *YES is specified on the **Return value (RTNVAL)** parameter.
- If either the **Range of values (RANGE)** parameter or the **Valid values (VALUES)** parameter is specified.
- If *LGL, *VARNAME, *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

If a *CHAR (character type) is specified on the **Type of value (TYPE)** parameter, the EBCDIC value of the character string is used as an unsigned integer in the comparison. If a character constant is specified in this parameter, it can be no longer than 32 bytes.

Variables can be coded for this element.

Top

Range of values (RANGE)

Specifies the range, or limits, for the parameter value. The parameter value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For example, 15 would be valid if RANGE was specified as (0 16).

For nonnumeric data types, such as character, the range of values and the data specified are right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length.

Variables can be coded for this element.

The upper and lower limits of the range can be specified either by a keyword representing the value or by the value itself. If a keyword is specified, it must be preceded by an ampersand (&) to indicate that the value of the keyword is to be tested. The value of its parameter at the time of the check is used to determine the range. The value that is tested is the value passed to the command processing program, not the user-specified value. If the keyword identifies a list of values or a qualified name, only the first value is used as the range limit. A keyword may not refer to a parameter that is defined with *NULL specified on the **Value to pass if unspecified (PASSVAL)** parameter. This parameter is not valid with *NULL specified on the **Value to pass if unspecified (PASSVAL)** parameter.

This parameter is also not valid in the following cases:

- If *YES is specified on the **Return value (RTNVAL)** parameter.
- If either the **Relational expression (REL)** parameter or the **Valid values (VALUES)** parameter is specified.

- If *LGL, *VARNAME, *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

Character constants specified in this parameter can be no longer than 32 bytes.

Variables can be coded for this element.

Top

Special values (SPCVAL)

Specifies a list of up to 300 entries that define special values that can be entered on the parameter. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the command processing program (CPP) without further checking. If the to-value is omitted, the from-value is passed to the CPP. This parameter is not valid if *YES is specified on the **Return value (RTNVAL)** parameter, or if *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

The from-value is a character string, but the to-value can be anything that is passable. However, for TYPE(*DATE) the to-value must be specified not quoted in the *mmddy*, *mmddyyyy*, or the *Cyyymmdd* format. If a CL variable is used for the from-value, its type must be *CHAR. The to-value must be no longer than specified on the **Value length (LEN)** parameter, and, if *DEC, *INT2, *INT4, *UINT2 or *UINT4 is specified for the **Type of value (TYPE)** parameter, the type of the to-value must be the same. If the **Type of value (TYPE)** parameter is a character type (such as *CHAR, *LGL or *DATE), the to-value must be a character string. Character constants specified in this parameter can be no longer than 32 bytes. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current library, rather than the value *CURLIB, is passed to the CPP. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the CPP.

Variables cannot be coded for this element.

Top

Single values (SNGVAL)

Specifies a list of up to 300 single values that can be specified for a parameter being defined as a mixed list or as a qualified name, when a statement label is specified for the **Type of value (TYPE)** parameter, or specifies that it is to accept two or more values as defined by the **Maximum values allowed (MAX)** parameter. Any one of the single values can be entered instead of a list of values or a qualified name that the parameter is defined to accept. Each entry specifies a character string (a from-value) that can be entered. If an entered character string matches the from-value of one of the entries and the to-value is specified, the data is replaced with the to-value and is then passed to the command processing program without further checking. If the to-value is omitted, the from-value is passed to the command processing program.

The to-value (or the from-value, if the to-value is omitted) must be passable, as specified in the **Special values (SPCVAL)** parameter. Character constants specified in this parameter can be no longer than 32 bytes. This parameter can be specified only if the **Maximum values allowed (MAX)** parameter is greater than 1 or if TYPE is specified as a statement label of a QUAL or ELEM statement. Each single value can only be substituted for a list of values or a qualified name; it cannot be a list item or qualifier. It is passed as the first and only element of the list.

This parameter is not valid if *YES is specified on the **Return value (RTNVAL)** parameter, or if *CMD, *CMDSTR, *X, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

If a to-value of *CURLIB is specified, the name of the current library, rather than the value *CURLIB, is passed to the command processing program. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the command processing program.

Variables cannot be coded for this element.

Top

Minimum values required (MIN)

Specifies the minimum number of values that must be entered for the parameter being defined. For a parameter that does not allow multiple like values, only zero (0) for optional and 1 for required can be specified as the minimum number of values.

Note: Required parameter statements must precede optional statements. If required parameter statements are not specified first, the system assumes that the specified parameter is optional, and the minimum number of values for required parameters is ignored.

For a parameter that allows multiple like values, because a value greater than 1 is specified for the **Maximum values allowed (MAX)** parameter, zero (0) indicates that no values need be entered; therefore, it is an *optional* parameter. A value of 1 or greater than 1 indicates the minimum number of values that must be entered for the parameter, and, therefore, it is a *required* parameter. The value cannot exceed 1 if *NULL is specified for the **Type of value (TYPE)** parameter.

0 The parameter is optional; it does not have to be entered.

minimum-number

Specify the minimum number of elements that must be specified for this parameter. If 1 is the assigned value, it specifies that at least one value is required for the parameter. If a number greater than 1 is specified, the parameter is a list that must have at least as many elements as the number specified.

Top

Maximum values allowed (MAX)

Specifies, if this PARM statement is defining a simple list parameter, the maximum number of list items that this list parameter can contain. If a value greater than 1 is specified, the parameter is capable of accepting multiple like values (that is, a simple list). This support is primarily intended for IBM-supplied commands. All values entered for this parameter (at the time the command is run) must satisfy the validity checking requirements specified by the other parameter values on this PARM statement.

Note: The values for a list parameter are passed consecutively, preceded by a 2-byte binary value that indicates the number of values entered in the parameter by the user. CL programs do not support the handling of binary values in variables.

1 The parameter accepts only one value; the parameter is not a list parameter.

maximum-number

Specify the maximum number of elements that the list parameter can accept. The specified maximum must be greater than or equal to the value specified in the **Minimum values required (MIN)** parameter, and less than or equal to 300. If the maximum is greater than 1 and a statement

label that identifies a QUAL or ELEM statement is not specified for the **Type of value (TYPE)** parameter, the parameter is a simple list of like elements (that is, each element in the list has the same requirements, such as type and length). If a statement label is specified and it points to the label of an ELEM or QUAL statement, a number greater than 1 should only be specified for this parameter if a list of lists or a list of qualified names is accepted. A maximum greater than 1 is not valid if *CMD, *CMDSTR, or *NULL is specified for the **Type of value (TYPE)** parameter, or if *YES is specified for the **Return value (RTNVAL)** parameter, or if the **Constant value (CONSTANT)** parameter is specified.

Top

Allow unprintable characters (ALWUNPRT)

Specifies whether this parameter will accept the hexadecimal character X'FF' or those hexadecimal characters in the range of X'00' through X'3F'. This parameter is valid only if *CHAR or *X is specified for the **Type of value (TYPE)** parameter.

- *YES** Characters can be passed to the command processing program and sent to the display or printer.
- *NO** Unprintable characters cannot be passed to the command processing program.

Top

Allow variable names (ALWVAR)

Specifies whether to allow variable names for the parameter. *NO is not allowed on this parameter if *VARNAME, *ZEROELEM, *NULL, or a statement label is specified on the **Type of value (TYPE)** parameter.

- *YES** Variable names can be used for the parameter.
- *NO** Variable names cannot be used for the parameter.

Top

Is PARM a program name (PGM)

Specifies whether this parameter element is a program name. *YES is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter. Specifying *YES here does not have any effect on the parameter element being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a program name. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

- *NO** The parameter defined in this PARM statement is not a program name.
- *YES** The parameter defined in this PARM statement is a program name.

Top

Is PARM a data area name (DTAARA)

Specifies whether the parameter is a data area name. *YES is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter. Specifying *YES here does not have any effect on the parameter being defined by the PARM statement; it only indicates to the compiler that the value for this parameter is a data area. This information is stored so that it can be included in the output of the Display Program References (DSPPGMREF) command.

***NO** The parameter defined in this PARM statement is not a data area name.

***YES** The parameter defined in this PARM statement is a data area name.

Top

If a file parameter, how used (FILE)

Specifies the expected use of the file and whether the parameter is a file name. The parameter can be specified as the name of a file that has a specific use so that, at compile time, the names can be used to get file reference information about where the files are used. The specification in this parameter does not have any effect on the operation of the parameter being defined; it only indicates to the compiler that the value for this parameter is a file name and what type of file it is. This information is stored so it can be included in the output of the Display Program References (DSPPGMREF) command. This parameter is valid only if a statement label, *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter. It is not valid if *YES is specified on the **Return value (RTNVAL)** parameter.

***NO** The parameter is not a file name.

***IN** The parameter value is an input file name.

***OUT** The parameter value is an output file name.

***UPD** The parameter value is an update file name.

***INOUT**

The parameter value is the name of a file that is used for both input and output.

***UNSPFD**

The parameter value is the name of a file, but its use cannot be specified.

The use of the file must match the type of file specified. For example, if *IN is specified, the file can be used only for input; if *UPD is specified, it can be used only to update existing records.

Top

Full field required (FULL)

Specifies whether the number of bytes in the parameter value must be exactly the same as the number specified on the **Value length (LEN)** parameter (if specified) or its default length (if LEN is not specified).

***NO** The number of bytes in the parameter value can be less than that specified by the LEN parameter.

***YES** The number of bytes in the parameter value must equal the number specified by LEN or the default length for that type. The exact length is valid only if *LGL, *CHAR, *NAME, *SNAME, *CNAME, *PNAME, *GENERIC, *VARNAME, or *HEX is specified for the **Type of value (TYPE)** parameter. Specifying *YES here is valid with *YES specified on the **Return value (RTNVAL)** parameter.

Top

Value an expression (EXPR)

Specifies whether the parameter named in the KWD parameter can accept an expression containing a character concatenation or a built-in function (%SUBSTRING or %BIN).

Restrictions: Expressions are not allowed on parameters where *CMD, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter.

***NO** The parameter value cannot be a concatenation expression or a built-in function.

***YES** The parameter value can be a concatenation expression or a built in function.

Top

Varying length (VARY)

Specifies whether the parameter value that is passed to the command processing program is preceded by a length value that indicates the number of bytes entered for the command parameter.

Note: The length value is the actual number of bytes entered for the command parameter, with trailing blanks removed. The length value passed may be different than the defined parameter length or the declared variable length. The length of the field containing the character string data is determined by the defined length for the parameter or the declared LEN for CL program variables. The length value defines how many bytes were actually entered for the command parameter. If *PNAME or *CHAR is specified for the TYPE parameter *and* *UTF16 is specified for the **CCSID of value (CCSID)** parameter, the length value passed to the command processing program represents the number of *bytes* for the parameter value. This could be up to twice the number specified for the **Value length (LEN)** parameter because the parameter value will be converted to UTF16 format and most UTF16 characters require two bytes. See the description of the CCSID and LEN parameters for more information related to handling values in UTF16 format.

Single values

***NO** The parameter value is not preceded by a length value.

Element 1: Return length value

***YES** The parameter value passed to the CPP is preceded by a field that indicates the number of bytes actually specified for the parameter. *YES is valid only for the following parameter types: *CHAR, *NAME, *SNAME, *CNAME, *PNAME, *GENERIC, *LGL, *VARNAME, *CMD, *CMDSTR, and *X. *YES must be specified if PASSATR(*YES) and RTNVAL(*YES) are specified.

Element 2: Value length

***INT2** The parameter length value is an integer passed as a 2-byte signed binary number.

***INT4** The parameter length value is an integer passed as a 4-byte signed binary number.

Top

Pass attribute byte (PASSATR)

Specifies whether an attribute byte is passed to the command processing program with the parameter data.

The attribute byte precedes the parameter data. If the parameter allows multiple values to be specified, an attribute byte precedes each value.

***NO** An attribute byte is not passed with the parameter.

***YES** An attribute byte is passed with the parameter.

The attribute byte has two fields:

1. The leftmost bit of the attribute byte indicates whether or not a value was specified. If the leftmost bit is '0'B, the value passed to the command processing program is a default value and was not specified in the command string. If the leftmost bit is '1'B, the value passed to the command processing program was specified in the command string.
2. The remaining seven bits describe the value passed to the command processing program when *CHAR is specified for the **Type of value (TYPE)** parameter.

Attribute	Description
'0000010'B	Meets *NAME rules, like A_B
'0000100'B	Meets *GENERIC rules, like AB*
'1000101'B	Quoted character string, like 'A B'
'0000101'B	Unquoted character string, like 5A
'1001000'B	Logical constant, '0' or '1'
'0001100'B	Hexadecimal value, like X'C1C2'
'0100001'B	Unsigned numeric value, like 5
'0101001'B	Unsigned numeric with decimal point, like 5.2
'0110001'B	Signed numeric value, like -5
'0111001'B	Signed numeric with decimal point, like -5.2

Top

Value to pass if unspecified (PASSVAL)

Specifies whether a value is passed to the command processing program for this parameter. *NULL is not valid if the parameter is a constant parameter (a parameter in which a value has been specified for the **Constant value (CONSTANT)** parameter, or a parameter for which *ZEROELEM or *NULL has been specified for the **Type of value (TYPE)** parameter, or a list/qualified name defined by all constant ELEM or QUAL statements). *NULL also is not valid if *YES has been specified on the **Return value (RTNVAL)** parameter, or if the value specified for the **Minimum values required (MIN)** parameter is greater than zero. A DEP statement or the REL and RANGE keywords of other PARM statements may not refer to the value of a parameter defined with *NULL.

***DFT** The default value is always passed to the command processing program.

***NULL**

A null pointer is passed to the command processing program if the parameter is not specified.

Top

Case of value (CASE)

Specifies whether the value that is passed to the CPP is changed from lowercase to uppercase, or is preserved in the case specified on for the command parameter.

***MONO**

The parameter value is changed from lowercase to uppercase. Parameters enclosed with apostrophes preserve the case whether or not this value is specified.

***MIXED**

The parameter value is preserved in the case specified on the command parameter. This value can be specified only on *CHAR and *PNAME parameter types.

Top

CCSID of value (CCSID)

Specifies the coded character set identifier (CCSID) to use when passing the parameter value to the command processing program.

Note: If the command string is run in batch using the Start Database Reader (STRDBRDR) or Submit Database Jobs (SBMDBJOB) command from a source file created with a CCSID of 1208 (UTF8) or was compiled from a CL source file created with a CCSID of 1208 or is being run using the QCAPCMD API and specifies that the CCSID of the command is either 1200 (UTF16) or 1208, the command string is assumed to be in Unicode format.

***JOB** If the command string is in Unicode format, the value for this parameter will be converted to the job CCSID before it is passed to the command processing program. Otherwise, the command string is assumed to already be in the CCSID of the job and no conversion is done for this parameter.

*UTF16

The parameter value will be passed as a UTF16 string to the command processing program (CPP). UTF16 is a Unicode format which is the same as CCSID 1200. In UTF16 format, most characters require two bytes (16 bits). If the command string is in a supported Unicode CCSID (1208 or 1200), little or no conversion of the parameter value will be performed. Otherwise, the command string is assumed to already be in the CCSID of the job, and the parameter value will be converted from the CCSID of the job to CCSID 1200 (UTF16).

If the parameter is optional and no value is specified in the command string, the defined default value will be converted to UTF16 format before it is passed to the CPP. If the parameter is a constant, the defined constant value will be converted to UTF16 format before it is passed to the CPP. If the parameter value is specified as a hexadecimal literal in the command string, the hexadecimal string is first converted to a character string using the CCSID of the job and the resulting character string is converted to UTF16 format before it is passed to the CPP.

*UTF16 is only allowed if *CHAR or *PNAME was specified for TYPE.

See the description of the **Value length (LEN)** parameter and **Varying length (VARY)** parameter for more information related to handling values in UTF16 format.

Top

List displacement (LISTDSPL)

Specifies whether the displacement to a list within a list is 2-bytes or 4-bytes long. These displacements are generated when a parameter being passed to a CPP has a list within a list. This parameter is ignored if the value being built for the CPP does not contain a list within a list.

***INT2** The displacement value is an integer passed as a 2-byte signed binary number.

***INT4** The displacement value is an integer passed as a 4-byte signed binary number.

Top

Display input (DSPINPUT)

Specifies whether the keyword value is shown in the job log or in a prompt display.

***YES** The default response, *YES, indicates that the parameter value is shown on the prompt display and in the job log.

*PROMPT

The response *PROMPT indicates that the parameter value is shown on the prompt display but not in the job log.

*NO The response *NO indicates that the value is not shown on either the prompt display or in the job log.

Top

Choice text (CHOICE)

Specifies the choices text that is displayed to the right of the input field on the prompt screen. Up to 30 characters of text can be displayed.

*VALUES

The choices text is generated based on the values specified for the TYPE, RSTD, RANGE, SNGVAL, SPCVAL, and VALUES parameters. If constants are specified for the RANGE parameter, the choices text begins with the minimum value and the maximum value separated by a hyphen. If RANGE is not specified with constants as the minimum and maximum values, and RSTD(*NO) is specified, the choices text begins with a short description of the parameter type based on the value specified for the TYPE parameter. Values specified for the SNGVAL parameter are added to the choices text, in the order the values are defined in the command definition source and separated by a comma and a blank. The last entries added to the choices text are values specified for the SPCVAL or VALUES parameter, in the order the values are defined in the command definition source and separated by a comma and a blank. If there are too many values to fit in 30 characters, the last value is followed by three periods.

The following are examples of possible choices text generated by CHOICE(*VALUES):

- If TYPE(*DEC) and RANGE(1.0 999.9) and SPCVAL((*NOMAX -1)) are specified, the choices text will be:
1.0-999.9, *NOMAX
- If TYPE(*NAME) and RSTD(*NO) and SNGVAL(*ALL) and SPCVAL(*LIBL *CURLIB) are specified, the choices text will be:
Name, *ALL, *LIBL, *CURLIB
- If RSTD(*YES) and SNGVAL(*ALL) and SPCVAL(*ALRTBL *BNDDIR *CHTFMT *CLD *CLS *CMD) are specified, the choices text will be:
*ALL, *ALRTBL, *BNDDIR...

*NONE

No values are displayed.

*PGM A program that is called determines the values that are displayed. The program that is called is identified in **Choice program (CHOICEPGM)** parameter of the PARM statement.

message-identifier

Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text'

Specify no more than 30 characters, enclosed in apostrophes.

Top

Choice program (CHOICEPGM)

Specifies the program to be called during command prompting to fill in the possible choices text and the permissible values. This parameter must be specified if *PGM is specified on the **Choice text (CHOICE)** parameter and may not be specified otherwise.

Single values

*NONE

No program is identified to fill in the possible choices text and permissible values.

Qualifier 1: Choice program

name Specifies the name of the program to be called during prompting to fill in the possible choices text or permissible values. If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

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 name of the library where the program is located.

Top

Prompt control (PMTCTL)

Specifies how prompting is to be controlled for this parameter. Prompting may be controlled by another parameter, specified by a Prompt Control (PMTCTL) statement referred to by label in this parameter, or by user request by pressing the F10 key.

*NONE

The parameter is always prompted, unless it is omitted due to selective prompting.

*PMTRQS

The parameter is not prompted unless:

- The user requests optional parameters to be prompted.
- A value was entered for the parameter before the prompt was called.
- The parameter was selected by selective prompt characters.

statement-label

Specify the label of the Prompt Control (PMTCTL) statement that is used to determine whether this parameter is prompted. The parameter is not prompted unless:

- The conditions specified on the referred to PMTCTL statement have been met.
- A value was entered for the parameter before the prompt was called.
- The parameter was selected by selective prompt characters.

Top

Prompt control program (PMTCTLPGM)

Specifies the program to be called to convert the value specified for the parameter into a value used on a Prompt Control (PMTCTL) statement. This parameter is valid only on parameters that are referred to in the **Controlling keyword (CTL)** parameter of a PMTCTL statement.

Single values

*NONE

No program is to be called to convert the parameter value for prompt control statements. If the parameter is specified in a Prompt Control (PMTCTL) statement, the actual value is compared in that PMTCTL statement.

Qualifier 1: Prompt control program

name Specify the name of the program to be called to convert the parameter value.

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 name of the library where the program is located.

Top

Key parameter (KEYPARM)

Specifies that this parameter is initially displayed when the command is prompted and a prompt override program was specified when the command was created or changed. If no prompt override program is specified, KEYPARM(*NO) is assumed for all parameters.

*NO The parameter is not displayed initially.

*YES The parameter is displayed initially.

Top

Initial prompt length (INLPMTLEN)

Specifies the length of the input field initially displayed for the parameter when the command is prompted. The user can extend the field to a maximum length of 512 bytes by entering an ampersand (&) in the first position of the field, followed by a blank. INLPMTLEN is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, *CNAME, *PNAME, *GENERIC, *CMDSTR, *HEX, *X, or *CMD. If FULL(*YES), RSTD(*YES), or CONSTANT are specified, INLPMTLEN(*CALC) must be specified or defaulted.

*CALC

The prompter will determine the length of the prompt field based on the type and length of the parameter.

*PWD If the current value of system value QPWDVLV is '0' or '1', the prompt field will be 10 bytes long. Otherwise, the length of the prompt field will be determined by the length of the parameter. INLPMTLEN(*PWD) is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, *PNAME, or *CNAME.

initial-prompt-length

Specify the initial length in bytes. Valid values are 1-12, 17, 25, 32, 50, 80, 132, 256, and 512.

Top

Prompt specifications (PROMPT)

Specifies what prompt text is used for the parameter. The prompt text gives a short description of the parameter which appears next to the parameter keyword and input field when the command is prompted. Prompt text cannot be specified if *ZEROELEM or *NULL is specified for the **Type of value (TYPE)** parameter, or if a constant value is specified in the **Constant value (CONSTANT)** parameter.

Single values

*NONE

No prompt text is shown for the parameter defined by this PARM statement. This parameter is still prompted by its keyword name, but no prompt text is shown beside the keyword name.

Element 1: Prompt text or message ID

message-identifier

Specify the message identifier that specifies the message containing the prompt text of up to 30 bytes that is shown when the parameter is prompted. If a message having the specified identifier cannot be found in the message file specified on the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

'prompt-text'

Specify the prompt text that is shown when the parameter is prompted. The text must be a character string of no more than 30 bytes, enclosed in apostrophes.

Element 2: Order prompt is displayed

relative-prompt-number

A relative prompt number may be specified for the parameter. The relative prompt number specifies the order in which parameter keywords are prompted. This order affects only the order of prompting, not the order in which the parameters are passed to the command processing program. Parameters having prompt numbers are prompted before parameters having no prompt numbers.

Top

Examples

Example 1: Define a Numeric Parameter

```
PARM KWD(X) TYPE(*DEC) LEN(2) MIN(1) REL(*GT 5)
```

The value for the parameter named X, a 2-digit decimal number, must be entered. The value must be greater than 5.

Example 2: Define a Parameter with Restricted Values

```
PARM KWD(CLASS) TYPE(*CHAR) LEN(1) DFT(A) +  
VALUES(A B C) RSTD(*YES)
```

The value of the parameter named CLASS must be A, B, or C, if specified. If CLASS is not specified, the default value passed to the command processing program will be A.

Example 3: Define a Parameter with Range of Valid Values

```
PARM KWD(MAXREC) TYPE(*DEC) LEN(3 0) MIN(1) +  
      RANGE(&MINREC 500)
```

The value of the MAXREC parameter must be entered as a decimal number of 3 digits or less, with no digits to the right of the decimal point. The value must be greater than or equal to the value entered (or defaulted) for parameter MINREC and also must be less than or equal to 500.

Example 4: Define a Simple List Parameter

```
PARM KWD(FILE) TYPE(*NAME) MIN(2) MAX(5)
```

The FILES parameter is a homogeneous list that contains a minimum of two names and a maximum of five names.

Example 5: Define a List Parameter with Restricted Values

```
PARM KWD(INVFNAME) TYPE(*NAME) DFT(*ALL) +  
      SNGVAL((*ALL XXX)) VALUES(DEPT1 DEPT2 DEPT3) +  
      FILE(*UPD) MIN(0) MAX(3) RSTD(*YES) +  
      PROMPT(USR0002 1)
```

The value of the parameter named INVFNNAME can be a list of up to three file names of which DEPT1, DEPT2, DEPT3, and *ALL are the valid choices. If *ALL is entered, no other values can be entered for the parameter. If this parameter is omitted, file name XXX is passed to the command processing program. If this parameter is entered through a command prompter, the prompt text for this parameter will be retrieved from message identifier USR0002 of the message file specified for the PMTFILE parameter on the Create Command (CRTCMD) command when the command is created. *ALL will be shown as the default parameter value.

Top

Error messages

None

Top

Program (PGM)

Where allowed to run:

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

Parameters
Examples
Error messages

Threadsafe: Yes

The Program (PGM) command is used in a CL source file to identify the start of a CL program or ILE CL procedure that is to be compiled and to specify the parameters that are to be received by the procedure after it is compiled. If a PGM command is used, it must be the first command in the source file; if a PGM command is not used, a PGM command without parameters is assumed. The name of the CL program or ILE CL procedure is specified on the CL command used to compile the CL source file.

The PGM command also specifies the parameters to be passed to the CL procedure, if any, when it is called for processing by another program. For information about how constants are passed, see the PARM parameter description for the Call (CALL) command.

If the CL program or ILE CL procedure source file is compiled to create a program (*PGM) object, the program can be called by a Call (CALL) or Transfer Control (TFRCTL) command, or by a routing entry in a subsystem description. When the program is called by a CALL or TFRCTL command, the specified parameters can be passed to it.

Parameters defined in this command must be passed when the procedure is called. The parameters passed must be of the type, length, and order specified in this command. Each of the parameter values can be a character string, a numeric value, or a CL variable. When received, each value is given a different CL variable name. Each CL variable name must be defined in the CL source file by a separate DCL (Declare) command before the procedure is compiled. Up to 255 parameters can be passed.

ILE programs and procedures will not detect parameter mismatches between the calling program or procedure and the called program or procedure. If the calling procedure passes more parameters than the called procedure expects, the called procedure will ignore the extra parameters. If the calling procedure passes fewer parameters than are specified on the called procedures PGM command, the results may be unpredictable.

Restrictions: This command is valid only in a CL program or ILE CL procedure.

Top

Parameters

Keyword	Description	Choices	Notes
PARM	Parameter CL variable names	Values (up to 255 repetitions): <i>CL variable name</i>	Optional, Positional 1

Top

Parameter CL variable names (PARM)

Specifies one or more CL variables that are to receive the parameter values passed to this procedure. Specify a CL variable name for each of the values to be received; the name must start with an ampersand (&).

Null values, *N, cannot be specified for any parameter. The parameter values are associated with the variables in the PARM parameter in the order in which they were specified on the CALL or TFRCTL commands. The type and length of each value passed must have matching attributes in the calling and receiving programs. However, for character constants, the receiving program can specify a shorter length; when this is done, the character string passed is truncated to the length declared in the receiving program. For information on how each data type is passed, see the description of the PARM parameter in the CALL command.

Note: If a parameter value is to be changed by a CL program or ILE CL procedure or specified as a variable on a CL command, it must be in writeable storage. For example, in C or C++, strings may be read only. If a read-only string is passed as a parameter to a CL program or ILE CL procedure, and the CL program or ILE CL procedure attempts to change the value of the variable or uses the variable on a CL command, the CL program or ILE CL procedure will fail.

CL-variable-name

Specify the name of the CL variable to receive the value passed from the calling program. A maximum of 255 variables can be specified.

Top

Examples

Example 1: CL Procedure Containing No Parameters

```
PGM
:
ENDPGM
```

This PGM command is the first command in a CL source file for a CL program or ILE CL procedure that contains no parameters.

Example 2: CL Procedure Containing Two Parameters

```
PGM PARM(&X &Y)
```

This is the first command in a CL source file for a CL program or ILE CL procedure that contains two parameters, &X and &Y, that have values passed to them from the calling program or procedure.

Example 3: CL Procedure Containing Two Parameters in Positional Form

```
PGM (&PARAM1 &PARAM2)
```

This is the first command in a CL source file for a CL program or ILE CL procedure that specifies two parameters in positional form, &PARAM1 and &PARAM2. When this procedure is called, the calling program or procedure passes the parameter values to be used for &PARAM1 and &PARAM2.

Top

Error messages

None

[Top](#)

Verify TCP/IP Connection (PING)

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

Parameters
Examples
Error messages

The Verify TCP/IP Connection (VFYTCPCNN) command, also known as PING, tests the connectivity between a system and the remote system specified by the remote system parameter.

Notes:

- The VFYTCPCNN (PING) command cannot be used to verify IP over SNA connections.
- The local domain name is used by many applications including PING. PING appends the local domain to a host name if a domain is not specified or if a period (.) does not appear at the end of the specified host name.

Top

Parameters

Keyword	Description	Choices	Notes
RMTSYS	Remote system	<i>Character value</i> , *INTNETADR	Required, Positional 1
INTNETADR	Remote internet address	<i>Character value</i>	Optional
ADRVERFMT	Address version format	* <u>CALC</u> , *IP4, *IP6	Optional
MSGMODE	Message mode	<i>Element list</i>	Optional
	Element 1: Response message detail	* <u>VERBOSE</u> , *QUIET	
	Element 2: Summary, if response errors	* <u>COMP</u> , *ESCAPE	
PKTLEN	Packet length (in bytes)	8-512, <u>256</u>	Optional
NBRPKT	Number of packets	1-999, <u>5</u>	Optional
WAITTIME	Wait time (in seconds)	1-120, <u>1</u>	Optional
LCLINTNETA	Local internet address	<i>Character value</i> , * <u>ANY</u>	Optional
TOS	Type of service	*MINDELAY, *MAXTHRPUT, *MAXRLB, *MINCOST, * <u>NORMAL</u>	Optional
IPTTL	IP time to live (hop limit)	1-255, * <u>DFT</u>	Optional

Top

Remote system (RMTSYS)

Specifies the remote system name of the host with which the Verify TCP/IP operation takes place. To be successful, the name must be valid, and the remote system must be able to communicate with the local system. You can assign names to an internet address by using either of the following:

- Work with Host Table menu, which is an option on the Configure TCP/IP menu.
- Remote name server to map a remote system name to an internet address.

Host name resolution will depend on the value specified for the **Address version format (ADRVERFMT)** parameter.

***INTNETADR**

The remote system is identified by the value specified for the **Remote internet address (INTNETADR)** parameter.

character-value

Specify the remote system name to be verified.

Top

Remote internet address (INTNETADR)

Specifies the remote internet address. Either a valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

character-value

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

Top

Address version format (ADRVERFMT)

Specifies how the host name specified for the **Remote system (RMTSYS)** parameter is to be resolved.

***CALC**

The host name resolution method will be 'calculated' (determined) based on the host name entered in the RMTSYS parameter. IP Version 6 host name resolution will be performed if the system has at least one IP Version 6 address configured. If an IP Version 6 address is not found, IP Version 4 host name resolution will be performed if the system has at least one IP Version 4 address configured. The loopback address is not considered in this case as a configured address.

***IP4** Use the IP Version 4 host name resolution method.

***IP6** Use the IP Version 6 host name resolution method.

Top

Message mode (MSGMODE)

Specifies the amount of information to be displayed.

Element 1: Response message detail

***VERBOSE**

Display messages as each PING response arrives.

***QUIET**

Display only the initial PING (VFYTCPCNN) message and the summary messages.

Element 2: Summary, if response errors

***COMP**

If the PING (CFYTCPCNN) request is successful, the summary message returned is a completion message.

*ESCAPE

A monitorable escape message is returned. This is useful if you have written a program to issue the PING request and wish to monitor the PING request for errors. See the error messages section of the PING (VFYTCPCNN) command help for a list of possible escape messages.

Top

Packet length (in bytes) (PKTLEN)

Specifies the length (in bytes) of the packets that are sent to the remote system.

256 The packet length is 256 bytes.

8-512 Specify the number of bytes in each packet.

Top

Number of packets (NBRPKT)

Specifies the number of packets that are sent to the remote system.

5 Five packets are sent.

1-999 Specify the number of packets that are sent to the remote system.

Top

Wait time (in seconds) (WAITTIME)

Specifies the number of seconds to wait for the return (echo) packet before declaring this packet transfer a failure.

1 The system waits 1 second.

1-120 Specify the number of second to wait.

Top

Local internet address (LCLINTNETA)

Specifies the local internet address of the interface that the outbound packets are to use. Any valid IP Version 4 or IP Version 6 address will be accepted. An IP Version 4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

*ANY Use any interface's local internet address.

character-value

Specify the local internet address.

Top

Type of service (TOS)

Specifies the type of service to be used. The type of service defines how the internet hosts and routers should make trade-offs between throughput, delay, reliability, and cost.

Note: This parameter is not used if IP Version 6 address resolution is used for verifying connectivity to a remote system.

***NORMAL**

Normal service is used for delivery of data.

***MINDELAY**

Minimize delay means that prompt delivery is important for data on this connection.

***MAXTHRPUT**

Maximize throughput means that a high data rate is important for data on this connection.

***MAXRLB**

Maximize reliability means that a higher level of effort to ensure delivery is important for data on this connection.

***MINCOST**

Minimize monetary cost means that lower cost is important for data on this connection.

Top

IP time to live (hop limit) (IPTTL)

Specifies the IP datagram (packet) time-to-live value. The datagram is valid only for the number of router hops specified by this parameter. The time-to-live value acts as a "hop counter". The counter is decremented each time the datagram passes through a router or gateway. Limiting the validity of the datagram by the number of hops helps to prevent internet routing loops.

Note: IP Version 6 refers to this parameter as the **hop limit**.

***DFT** Use the default time-to-live value.

The default time-to-live value for multicast addresses is 1. The default time-to-live value for all other addresses is specified by the IPTTL parameter of the Change TCP/IP Attributes (CHGTCPA) command.

1-255 Specify an IP datagram (packet) time-to-live value.

Top

Examples

Example 1: Verify TCP/IP Connection with a Specified Host Name

```
VFYTCPCNN  RMTSYS(IPHOST)  PKTLEN(100)  NBRPKT(10)
           WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system (known to the TCP/IP configuration as IPHOST) over a TCP/IP link. Each packet transfer must take place within 15 seconds or it fails.

Example 2: Verify TCP/IP Connection with an IP Address

```
VFYTCPCNN  RMTSYS(*INTNETADR)  INTNETADR('128.1.1.10')
           PKTLEN(100)  NBRPKT(10)  WAITTIME(15)
```

This command attempts to send 10 packets of 100 bytes each to a remote system over a TCP/IP interface. The user represents the RMTSYS with its internet address 128.1.1.10, rather than with an assigned system name. Each packet transfer that takes more than 15 seconds fails.

Example 3: Verify TCP/IP Connection with Host Name and Using a Specific Local Interface Address

```
VFYTCPCNN  RMTSYS(IPHOST)  MSGMODE(*QUIET)
           LCLINTNETA('9.2.2.3')
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system over a specific TCP/IP interface that has the local address 9.2.2.3.

Because MSGMODE(*QUIET) is specified, only the primary output messages are displayed. The interface parameter is useful on multi-homed hosts to verify network connectivity through a specific physical interface.

Example 4: Verify TCP/IP Connection with an IP Version 6 Address

```
VFYTCPCNN  RMTSYS(*INTNETADR)
           INTNETADR('1:2:3:4:5:6:7:8')
```

This command attempts to verify the TCP/IP connection of a remote system that has the local address of **1:2:3:4:5:6:7:8**.

Example 5: Verify TCP/IP Connection with a Specified IP Version 6 Defined Host Name

```
VFYTCPCNN  RMTSYS(IPV6HOST)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

The default "Address version format" is *CALC. Host name resolution may return multiple IP addresses for a given host name. But, in the case (*CALC), the first IP address (IP Version 4 or IP Version 6) resolved will be the address used when attempting to verify its connection over a TCP/IP link.

Example 6: Verify TCP/IP Connection and Explicitly Use IP Version 6 Host Name Resolution

```
VFYTCPCNN  RMTSYS(IPV6HOST)  ADRVERFMT(*IP6)
```

This command attempts to send 5 packets (default) of 256 bytes each (default) to a remote system (known to the IP Version 6 TCP/IP configuration as IPV6HOST) over a TCP/IP link.

This example differs from example 5 in that only a valid IP version 6 resolved address, for IPV6HOST, will be used when attempting to verify its connection over a TCP/IP link.

Top

Error messages

None

*ESCAPE Messages

TCP3210

Connection verification statistics: &1 of &2 successful (&3 %).

TCP3219

Address &1 does not match address version format &2.

[Top](#)

Configure PM Agent Line (PMLINMON)

Where allowed to run:

- Interactive job (*INTERACT)

Threadsafe: No

Parameters
Examples
Error messages

The Configure PM Agent Line (CFGPMLIN) command allows PM Agent to vary off a line that is in 'Connect pending' state, transmit the PM Agent performance data, and then put the line back in the 'Connect pending' state.

When you use this command, you change the PM Agent transmission task (Q1PCM1) to check for line status and vary off the appropriate line (Q1PMOFF). Once the transmission is complete, the same line is placed in a 'Connect pending' state (Q1PMON).

1. Read the warning that is shown on the first display and then press Enter.
2. Use the prompt Do you want PM Agent automatic line control active? as a master control switch for the function. If you specify YES, the PM Agent function is active. If you specify NO, the function is disabled.

If you specify NO, you do not need to define the Line Control list again when YES is specified.

You can vary off and on a line by specifying the line only. You can vary off and on a line, controller, and device by specifying all three descriptions.

3. Verify the line, controller, and device that you defined. Press Enter to see a summary of your choices.
4. Press Enter to confirm your choices or press F12 to return to the previous display to change your entries. When you press F3, you are taken out of the line control panel.

Top

Parameters

None

Top

Examples

CFGPMLIN

This command will show a warning panel and, if Enter is pressed, will show the PM Agent line control panel.

Top

Error messages

None

Top

Prompt Control Definition (PMTCTL)

Parameters
Examples
Error messages

The Prompt Control (PMTCTL) statement specifies a condition that is tested to determine whether prompting is done for the parameters whose PARM statement referred to this PMTCTL statement. The PMTCTL statement must have a statement label that matches the label referred to in the **Prompt control (PMTCTL)** parameter of one or more PARM statements in the command definition source.

Top

Parameters

Keyword	Description	Choices	Notes
CTL	Controlling keyword	<i>Name</i>	Required, Positional 1
COND	Controlling conditions	Values (up to 50 repetitions): <i>Element list</i>	Required, Positional 2
	Element 1: Relational operator	*GT, *EQ, *GE, *NL, *LT, *NE, *LE, *NG, *SPCFD, *UNSPCFD	
	Element 2: Controlling keyword value	<i>Character value</i>	
NBRTRUE	Number of true conditions	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Relational operator	*GT, *EQ, *GE, *NL, *LT, *NE, *LE, *NG	
	Element 2: Number of true conditions	0-25	
LGLREL	Logical relation	*AND , *OR	Optional

Top

Controlling keyword (CTL)

Specifies the name of the parameter that controls the prompting. The value of the parameter specified here is compared to the value specified in the **Controlling conditions (COND)** parameter. If the **Prompt control program** prompt (PMTCTLPGM parameter) of the PARM statement is coded for the parameter specified here, the value returned by the program specified in that PMTCTLPGM parameter is compared to the values specified on the **Controlling conditions (COND)** parameter. If the parameter specified here is a list or qualified name, only the first list item or qualifier is compared.

Top

Controlling conditions (COND)

Specifies the condition against which the parameter specified on the **Controlling keyword (CTL)** parameter is tested. Up to 50 conditions can be specified.

*SPCFD

The condition is true, including the default value, if it is specified for the control parameter.

***UNSPCFD**

The condition is true only if the control parameter is not specified. It is not true if the default value is specified.

relational-operator-value

Specify the relational operator and value used to compare the value of the control parameter to the value specified in the **Controlling conditions (COND)** parameter. Valid values are *GT, *EQ, *NL, *LT, *NE, *LE, and *NG.

Top

Number of true conditions (NBRTRUE)

Specifies the number of conditions specified on the **Controlling conditions (COND)** parameter that must be true if the parameter is prompted for.

***ALL** All the conditions must be true.

relational-operator-value

Specify the relational operator and number used to compare the number of conditions that are true to the number specified in the **Number of true conditions (NBRTRUE)** parameter. Valid values are *GT, *EQ, *GE, *NL, *LT, *NE, *LE, and *NG.

Top

Logical relation (LGLREL)

Specifies, when PMTCTL statements are in a group, the logical relationship between this PMTCTL statement and the preceding PMTCTL statements in the group. This allows conditional prompting using more than one controlling parameter.

***AND** Performs a logical AND operation of the resulting condition for this PMTCTL statement with the previous PMTCTL statement.

***OR** Performs a logical OR operation of the resulting condition for this PMTCTL statement with the previous PMTCTL statement.

Top

Examples

Example 1: Selective Prompting with One Control Parameter

```
A:   PMTCTL  CTL(TYPE) COND((*EQ *) (*EQ *LIST)) +
      NBRTRUE(*EQ 1)
```

If either TYPE(*) or TYPE(*LIST) is specified, the parameters which reference this PMTCTL statement are selected for prompting.

Example 2: Selective Prompting with Multiple Control Parameters Using Multiple PMTCTL Statements

```
B:   PMTCTL  CTL(P1)  COND((*EQ *ALL))
      PMTCTL  CTL(P1)  COND((*EQ *SOME))  LGLREL(*OR)
      PMTCTL  CTL(P2)  COND((*EQ *ALL))  LGLREL(*AND)
      PMTCTL  CTL(P1)  COND((*EQ *NONE))  LGLREL(*OR)
      PMTCTL  CTL(P2)  COND((*NE *ALL))  LGLREL(*AND)
```

The parameters which refers to this group of PMTCTL statements are selected for prompting if any of the following conditions exist:

- *ALL is specified for P1.
- *SOME is specified for P1 and *ALL is specified for P2.
- *NONE is specified for P1 and *ALL is not specified for P2.

[Top](#)

Error messages

None

[Top](#)

Position Data Base File (POSDBF)

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

Parameters
Examples
Error messages

The Position Database File (POSDBF) command allows you to set the position of a database file to either the beginning or end of an open file.

Top

Parameters

Keyword	Description	Choices	Notes
OPNID	Open file identifier	<i>Name</i>	Required, Positional 1
POSITION	File position	*START, *END	Required, Positional 2

Top

Open file identifier (OPNID)

Identifies the opened file to reposition. This file must be opened by either the Open Database File (OPNDBF) or Open Query File (OPNQRYF) command.

This is a required parameter.

name Specify the open file identifier.

Top

File position (POSITION)

Specifies the starting or ending position of the database file.

This is a required parameter.

*START

The position of the database file is set to the start position of the member currently open. After the start position is set, a read next operation gets the first record in the member. A previous read operation gets the last record in the previous member, if *ALL is specified for the **Overriding member (MBR)** parameter of the Override with Database File (OVRDBF) command. Otherwise, a *get past start of file* exception occurs.

***END** The position of the database file is set to the end of the member currently open. After the end position is set, a read next operation gets the first record in the next member, if *ALL is specified for the MBR parameter of the Override with Database File (OVRDBF) command. Otherwise, a *get past end of file* exception occurs. A read previous operation gets the last record in the member.

Top

Examples

POSDBF OPNID(XXX) POSITION(*START)

This command sets the record position of the database file that is opened with OPNID(XXX) to the starting position of the database file member that is currently open.

[Top](#)

Error messages

*ESCAPE Messages

CPF5213

Positioning of member &3 failed.

CPF5230

No file open with OPNID(&4).

[Top](#)

Print Adopting Objects (PRTADPOBJ)

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

Parameters
Examples
Error messages

The Print Adopting Objects (PRTADPOBJ) command allows you to print a report of the objects that adopt the special and private authorities of the specified user profile. This is a way to check for security exposures associated with program adoption.

Restrictions:

1. You must have *ALLOBJ or *AUDIT special authority to use this command.
2. The user profile specified on the command is locked while the command is running. The lock prevents such things as objects having their owner changed to this profile. If this profile owns a lot of objects, the profile could be locked for an extended period of time.

This command will print two reports for a user profile. The first report (Full Report) will contain all of the objects that adopt the authorities of the user profile. The second report (Changed Report) will contain the objects that now adopt the authorities of the user profile that did not adopt the authorities of the user profile when the PRTADPOBJ command was previously run for the user profile. If the PRTADPOBJ command was not previously run for the user profile, there will be no 'Changed Report'. If the command has been previously run for the user profile but no additional objects adopt the authorities of the user profile, then the 'Changed Report' will be printed but there will be no objects listed.

The reports will contain the following information:

- The name of the user profile.
- The special authorities that the user profile has.
- The date and time the report was last run (shown on Changed Report only).
- An entry for each object that adopts the user profile's authority. Each entry contains the following information:
 - The name of the object.
 - The type of object.
 - The object's *PUBLIC authority. If the object or the object's library is locked at the time the report is created, the value is set to *LOCKED.
 - The name of the library the object is in.
 - The library's *PUBLIC authority. If the library is locked at the time the report is created, the value is set to *LOCKED.
 - An indication of whether there are any private authorities on the object ('Y' or 'N'). If the object or the object's library is locked at the time the report is created, the value is left blank.

Note: If there are no objects that adopt the authority of a user profile, no reports will be printed for that user. If none of the user profiles specified on the command have objects that adopt the authority of the user profiles, then there will be no reports generated.

The file QSECADPOLD in library QUSRSYS contains information from the last time the PRTADPOBJ command was run for a user profile. There is a member within the file, with the same name as the user profile, for each profile that has been previously specified on the command. System file QADPGMAD in library QSYS with format name of QSYPGMAD is the model file for the QSECADPOLD file.

Top

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	<i>Generic name, name, *ALL</i>	Required, Positional 1
CHGRPTONLY	Changed report only	<u>*NO</u> , *YES	Optional, Positional 2

Top

User profile (USRPRF)

This is a required parameter.

The name of the user profile whose adopted object information will be printed.

***ALL** The adopted information will be printed for all user profiles.

name The name of the user profile to print the adopted information for.

generic-name

The generic name of the user profile to print the adopted information for. A generic name is a character string of one or more characters followed by an asterisk (*).

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Top

Examples

```
PRTADPOBJ  USRPRF(OURSECOFR)
```

This command prints both full and changed reports for the objects that adopt the special and private authorities of the user profile OURSECOFR.

Top

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

Print AFP Data (PRTAFPDTA)

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

Parameters
Examples
Error messages

The Print Advanced Function Printer Data (PRTAFPDTA) command prints output received from a System/370 host. This command allows the user to specify the file being printed and the parameters used to control the print operation.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name, *FIRST</i>	Optional
DEV	Print device	<i>Name, *JOB, *SYSVAL</i>	Optional
FORMDF	Form definition	Single values: *DEVDF, *INLINE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Form definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
COPIES	Number of copies	1-255, <u>1</u>	Optional
STRPAGE	Starting page	<i>Integer, <u>1</u></i>	Optional
ENDPAGE	Ending page	<i>Integer, *END</i>	Optional
FIDELITY	Print fidelity	<i>*ABSOLUTE, *CONTENT</i>	Optional

Top

File (FILE)

Specifies the Advanced Function Printing Data Stream (AFPDS) file to be printed. Only physical files are supported for this command. If you use the Override with Printer File (OVRPRTF) command with PRTAFPDTA, do not override the device type (DEVTYPE parameter).

This is a required parameter.

Qualifier 1: File

name Specify the name of the AFPDS to be printed.

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 file. 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

Member (MBR)

Specifies the member that contains the data to be printed.

*FIRST

The first member in the database file is used.

name Specify the name of the file member that contains the data to be printed.

Top

Print device (DEV)

Specifies the printer that prints the file.

*JOB The printer device specified in the job description is used.

*SYSVAL

The value specified in the system value QPRTDEV is used.

name Specify the name of the printer device.

Top

Form definition (FORMDF)

Specifies the form definition to use when printing the file. A form definition is a resource object that defines the characteristics of the form such as: overlays, position of page data on the form, number of copies of pages, and modification to pages. The form definition is located inline with the file being printed, or in a library.

Single values

*DEVDF

The device description obtains the name of the form definition being used. If no value is specified, *DEVDF is assumed.

*INLINE

The form definition that is inline with the printer file is used.

Qualifier 1: Form definition

name Specify the name of the form definition that must exist in the library named. A maximum of 8 characters can be used.

Qualifier 2: Library

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

*CURLIB

The current library 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

Number of copies (COPIES)

Specifies, for spooled files, the number of copies being printed.

1 One copy of the output is printed.

1-255 Specify the number of copies that are to be printed.

Top

Starting page (STRPAGE)

Specifies the page on which printing starts. This parameter is used for partial printing of a file.

1 Printing starts on page 1. If the start page is not specified, 1 is assumed.

integer

Specify the page number on which printing starts.

Top

Ending page (ENDPAGE)

Specifies the page on which printing ends. This parameter is used for partial printing of a file ending at a specified page number. If both the start page and the end page are specified, the end page must be greater than or equal to the start page. Specifying an end page beyond the end of the actual file does not create an error condition.

*END Printing concludes at the end of the file.

integer

Specify the page number on which printing ends.

Top

Print fidelity (FIDELITY)

Specifies the degree of exactness required when printing the file.

*ABSOLUTE

The job is printed only if the file can be printed exactly as specified by the data stream and external controls.

*CONTENT

Prints the file using all available exception handling.

Top

Examples

Example 1: Printing Specific Pages

```
PRTAFPD TA FILE(MYLIB/MYFILE) STRPAGE(2) ENDPAGE(6)
```

This command prints the first member in file MYFILE in library MYLIB starting with page 2 and ending on page 6.

Example 2: Printing Using All Available Exception Handling

```
PRTAFPDTA FILE(MYLIB/MYFILE) FORMDF(F10101) FIDELITY(*CONTENT)
```

This command prints the first member in file MYFILE in library MYLIB using a form definition of F10101 and all available exception handling.

[Top](#)

Error messages

*ESCAPE Messages

CPF511B

Data stream not correct for record &2 in file &1.

PQT4001

Data stream not valid in structured field &2 in file &1.

PQT4003

Form definition &2 not found in library.

PQT4004

Starting page number &1 greater than ending page number &2.

PQT4006

Unable to process file &1 because of variable length fields.

PQT4007

Data stream not valid in file &1.

[Top](#)

Print Command Usage (PRTCMDUSG)

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

Parameters
Examples
Error messages

The Print Command Usage (PRTCMDUSG) command creates a cross-referenced listing of a specified group of CL commands that are used in a specified group of CL programs. The report shows, program by program, which of the specified commands are used in each program. The report can be used to identify which programs need to be recompiled because of changes that have been made to the command definition objects of commands specified on the PRTCMDUSG command. Note that this command can take a long time to run and can make a lot of printed output.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	Values (up to 50 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Command	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL	
PGM	Program	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Program	<i>Generic name, name</i> , *ALL	
	Qualifier 2: Library	<i>Name</i> , *USRLIBL , *CURLIB , *ALLUSR	

Top

Command (CMD)

Specifies the names of up to fifty CL commands for which specified programs are searched and printed in a report. The system searches the specified programs for every occurrence of each command you specify.

Note: PRTCMDUSG cannot be used to print the command usage for ILE CL programs and modules.

This is a required parameter.

Qualifier 1: Command

name Specify the name of a command.

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 that contains the CL commands whose usage in CL programs is to be reported.

Top

Program (PGM)

Specifies one or more CL programs that are searched for the specified commands. Only the programs and libraries for which you have some (any) authority are included in the report. This parameter also can specify that all (*ALL) programs in the specified library or libraries (*USRLIBL/*ALL, for example) are searched.

Qualifier 1: Program

***ALL** All CL programs in the specified library for which the user has some authority are searched to locate the specified CL commands.

generic-name

Specify the generic name of several programs in the specified library qualifier that are searched for the specified commands. A generic name can be specified as a character string that contains one or more characters followed by an asterisk (*).

name Specify the name of the CL program that is to be searched for the specified CL commands.

Qualifier 2: Library

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

***CURLIB**

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

***ALLUSR**

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

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

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

name Specify the name of the library to be searched.

Examples

```
PRTCMDUSG  CMD(CPYF)  PGM(PAYROLL/*ALL)
```

This commands searches all CL programs in the library PAYROLL for the Copy File (CPYF) commands and prints the names of both the commands and the program.

Error messages

*ESCAPE Messages

CPF0593

PRTCMDUSG command ended by controlled end.

CPF0595

PRTCMDUSG command ended.

CPF0596

PRTCMDUSG command ended. Cannot open print file.

Print Communications Security (PRTCMNSEC)

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

Parameters
Examples
Error messages

The Print Communications Security (PRTCMNSEC) command allows you to print a report containing the security attributes of the *DEV, *CTLD and *LIND objects currently on the system. This command provides a way to check the security of your communications configuration on the system.

The Print Communications Security command will create two spooled output files containing communications security information. The first spooled output file will contain a report generated by the Display Configuration List (DSPCFGL) CL command. This report will contain the entries currently in the APPN remote configuration list QAPPNRMT. If the QAPPNRMT configuration list does not exist on the system then no report will be printed. The second spooled output file contains the security attributes of the *DEV, *CTLD and *LIND objects on the system.

Restriction: You must have *ALLOBJ and *IOSYSCFG, or *AUDIT special authority to use this command.

The spooled output file containing the *DEV, *CTLD and *LIND objects will contain two reports. The first report (Full Report) will contain all of the communications objects and will print the security attributes of each object. The second report (Changed Report) will contain the communications objects that have changed since the PRTCMNSEC command was last run. If the PRTCMNSEC command was not previously run, there will be no 'Changed Report'. If the command has been previously run but no communication object information has changed then the 'Changed Report' will be printed but there will be no objects listed.

The first report will contain the entries from the APPN remote configuration list object QAPPNRMT. If the QAPPNRMT configuration list does not exist then no report will be printed.

The second report will contain the information listed below. The report lists *DEV, *CTLD and *LIND object types. Some fields may be blank or set to zero if the field does not apply to the type of object listed in the report.

*DEV object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *DEV object on the system.
 - The name of the communications object.
 - The object type of the communications object.
 - The device category of the communications object.
 - The secure location value of the communications object.
 - An indication if there is a location password for the communications object.
 - The APPN capable value of the communications object.
 - The single session value of the communications object.
 - The pre-establish session value of the communications object.
 - The SNUF program start value of the communications object.

*CTLD object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *CTLD object on the system.
 - The name of the communications object.
 - The object type of the communications object.
 - The controller category of the communications object.
 - The auto create value of the communications object.
 - The switched controller value of the communications object.
 - The call direction value of the communications object.
 - The APPN capable value of the communications object.
 - The CP sessions value of the communications object.
 - The disconnect timer value of the communications object.
 - The auto delete minutes value of the communications object.
 - The device name value of the communications object.

*LIND object types:

The reports will contain the following information:

- The object type being reported.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each *LIND object on the system.
 - The name of the communications object.
 - The object type of the communications object.
 - The line category of the communications object.
 - The auto create value of the communications object.
 - The auto delete minutes value of the communications object.
 - The auto answer value of the communications object.
 - The auto dial value of the communications object.

The file QSECCMNOLD in library QUSRSYS contains information from the last time the PRTCMNSEC command was run for a library. System file QASECCMN in library QSYS with format name of QSECCMN is the model file for the QSECCMNOLD file.

Top

Parameters

Keyword	Description	Choices	Notes
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 1

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO The full and changed reports will be printed.

*YES Only the changed report will be printed.

Top

Examples

PRTCMNSEC

This command prints both full and change report for the communication security information.

Top

Error messages

*ESCAPE Messages

CPFB307

Command &1 in use in another job.

Top

Print Communications Trace (PRTCMNTRC)

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

Parameters
 Examples
 Error messages

The Print Communications Trace (PRTCMNTRC) command transfers the communications trace data for the specified line, network interface description, or network server description to a spooled file or an output file.

The PRTCMNTRC command can also be used to format communications trace data that was previously dumped to a stream file using the Dump Communications Trace (DMPCMNTRC) command.

Restrictions:

- You must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- The trace data for network server description traces can only be transferred to a spooled file. The trace data cannot be transferred to an output file. There are no formatting options available.
- The following user profiles have authority to this command:
 - QSECOFR
 - QSRV

Top

Parameters

Keyword	Description	Choices	Notes
CFGOBJ	Configuration object	<i>Name</i>	Optional, Positional 1
FROMSTMF	From stream file	<i>Path name</i>	Optional
CFGTYPE	Type	*LIN, *NWI, *NWS	Optional, Positional 2
OUTPUT	Output	*PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	
CODE	Character code	*EBCDIC, *ASCII, *CALC	Optional
SLTLIND	Line description	<i>Name</i> , *ALL	Optional
SLTCTLD	Controller description	<i>Name</i> , *ALL	Optional
FMTSNA	Format SNA data only	*NO, *YES	Optional
FMTRR	Format RR, RNR commands	*NO, *YES	Optional

Keyword	Description	Choices	Notes
FMTTCP	Format TCP/IP data	<u>*LINTYPE</u> , *YES, *NO	Optional
FMTLCP	Format LCP data	<u>*YES</u> , *NO	Optional
FMTNCP	Format NCP data	<u>*YES</u> , *NO	Optional
TCPIPADR	Format TCP/IP data by address	<i>Element list</i>	Optional
	Element 1: Source/destination IP address	<i>Character value, *<u>ALL</u></i>	
	Element 2: Source/destination IP address	<i>Character value, *<u>ALL</u></i>	
SLTPORT	IP port number	<i>Decimal number, *<u>ALL</u></i>	Optional
FMTUI	Format UI data only	<u>*NO</u> , *YES	Optional
FMTMAC	Format MAC or SMT data only	<u>*NO</u> , *YES	Optional
FMTETH	Format Ethernet data only	*NO, <u>*YES</u>	Optional
FMTCCD	Format call control data	<u>*NO</u> , *YES	Optional
FMTBCD	Format broadcast data	*NO, <u>*YES</u>	Optional
EXCLMI	Exclude LMI data	<u>*NO</u> , *YES	Optional
FMTLMI	Format LMI data only	<u>*NO</u> , *YES	Optional
FMTHPRIP	Format HPR over IP data only	<u>*NO</u> , *YES	Optional
FMTLDLCIP	Format LDLC over IP data only	<u>*NO</u> , *YES	Optional

Top

Configuration object (CFGOBJ)

Specifies the configuration object being traced. The object must be a line description, a network interface description, or a network server description.

Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

name Specify the name of the configuration description object.

Top

From stream file (FROMSTMF)

Specifies the path name of the stream file from which communications trace data is formatted. This file must have been created by running the Dump Communications Trace (DMPCMNTRC) CL command. Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be specified.

path-name

Specify the path name of the stream file created by the DMPCMNTRC command.

Top

Type (CFGTYPE)

Specifies the type of configuration description that was traced.

Either the CFGOBJ and CFGTYPE parameters or the FROMSTMF parameter must be 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

Output (OUTPUT)

Specifies whether the output from the command is printed with the job's spooled output or sent to a database file.

Note: For network server description traces, *PRINT must be specified for this parameter.

*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 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

Character code (CODE)

Specifies whether the extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII) character code is used on the line.

*CALC

The system determines whether to format the user data in EBCDIC or ASCII, based on the type of controller that is used.

***ASCII**

The ASCII character code is used.

***EBCDIC**

The EBCDIC character code is used.

Top

Line description (SLTLIND)

Specifies whether to format data for all lines or a specific line communicating on the network during a trace.

***ALL** Formats the data for all lines.

name Specify the name of the line for which trace data is formatted.

Top

Controller description (SLTCTLD)

Specifies whether to format data for all controllers or a specific controller communicating on the network during a trace.

***ALL** Formats data for all controllers.

controller-name

Specify the name of the controller for which trace data is formatted.

Top

Format SNA data only (FMTSNA)

Specifies whether line protocol data or Systems Network Architecture (SNA) data is formatted. Line protocol data includes SDLC, X.25, Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Ethernet DIX V2, DDI, wireless, and IBM Token-Ring Network (TRLAN).

***NO** Only line protocol data is formatted.

***YES** Only SNA data is formatted.

Top

Format RR, RNR commands (FMTRR)

Specifies whether receiver ready (RR) and receiver not ready (RNR) commands are formatted with other data.

***NO** RR and RNR commands are not formatted with other data.

***YES** RR and RNR commands are formatted with other data.

Top

Format TCP/IP data (FMTTCP)

Specifies whether line protocol data or Transmission Control Protocol/Internet Protocol (TCP/IP) data is formatted.

Note: If the trace data is being formatted from a stream file (FROMSTMF parameter), this parameter is ignored and TCP/IP data is formatted.

***LINTYPE**

For X.25, Ethernet, DDI, wireless, Token-Ring, and Frame Relay lines, only line protocol data is formatted. For all other lines supporting TCP/IP, TCP/IP data is formatted.

***YES** TCP/IP data is formatted.

***NO** TCP/IP data is not formatted.

Top

Format LCP data (FMTLCP)

Specifies whether Link Control Protocol (LCP) data is included in the formatted communications trace.

Note: If FMTLCP, FMTNCP, and FMTTCP are all specified *NO when formatting data for a Point-to-Point Protocol (PPP) line, then asynchronous and unrecognized data will be placed in the spooled file. This is also the case if all are specified *YES (or *LINTYPE for FMTTCP). In all other cases asynchronous and unrecognized data will be omitted.

***YES** LCP data is formatted.

*NO LCP data is not formatted.

Top

Format NCP data (FMTNCP)

Specifies whether Network Control Protocol (NCP) data is included in the formatted communications trace.

*YES NCP data is formatted.

*NO NCP data is not formatted.

Top

Format TCP/IP data by address (TCPIPADR)

Specifies an internet address pair for which TCP/IP data is formatted. Any values that are valid for IP address 1 are also valid for IP address 2.

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.

For IPv6 (IP version 6) addresses, the form is *xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx*, where *x* is any valid hexadecimal digit 0 through F.

Note: IPv6 addresses are only valid when formatting trace data from a stream file.

Element 1: Source/destination IP address

*ALL The communications between the systems specified for element 2 and all other systems are printed.

character-value

Specify the address of the system for which communications between this system and the systems specified for element 2 are printed.

Element 2: Source/destination IP address

*ALL The communications between the systems specified for element 1 and all other systems are printed.

character-value

Specify the address of the system for which communications between this system and the systems specified for element 1 are printed.

Top

IP port number (SLTPORT)

Specifies whether data for all internet protocol (IP) ports or only a single IP port is formatted.

Note: This parameter is valid only if FMTTCP(*YES) is specified.

*ALL Data for all IP ports is formatted.

decimal-number

Specify the IP port number (1 to 65535) whose data is to be formatted.

Top

Format UI data only (FMTUI)

Specifies whether line protocol data or unnumbered information (UI) data is formatted.

*NO All line protocol data is formatted.

*YES Only UI data is formatted.

Top

Format MAC or SMT data only (FMTMAC)

Specifies whether line protocol data or medium access control (MAC) or Station Management (SMT) data is formatted.

*NO The line protocol data (TRLAN or Ethernet) is formatted.

*YES Only MAC or SMT data is formatted.

Top

Format Ethernet data only (FMTETH)

Specifies whether IEEE 802.3 data or Ethernet V2 data is formatted.

*YES Both IEEE 802.3 data and Ethernet V2 data are formatted.

*NO Only IEEE 802.3 data is formatted.

Top

Format call control data (FMTCCD)

Specifies whether all network interface data or only Integrated Services Digital Network (ISDN) signalling data is formatted.

*NO All network interface data is formatted.

*YES Only ISDN signaling data is formatted.

Top

Format broadcast data (FMTBCD)

Specifies whether broadcast data and data received containing destination MAC addresses is formatted.

*YES Broadcast data is formatted.

*NO Broadcast data is not formatted.

Top

Exclude LMI data (EXCLMI)

Specifies whether to exclude local management interface (LMI) data from the formatted output.

***NO** LMI data is not excluded from the formatted output.

***YES** LMI data is excluded from the formatted output.

Note: You cannot specify *YES for both the EXCLMI and FMTLMI parameters.

Top

Format LMI data only (FMTLMI)

Specifies whether local management interface (LMI) data is formatted.

***NO** LMI data is not formatted.

***YES** LMI data is formatted.

Note: You cannot specify *YES for both the EXCLMI and FMTLMI parameters.

Top

Format HPR over IP data only (FMTHPRIP)

Specifies whether High Performance Routing Protocol (HPR) over IP data is included in the formatted communications trace.

***NO** HPR over IP data is not formatted.

***YES** Only HPR over IP data is formatted.

Top

Format LDLC over IP data only (FMTLDLCIP)

Specifies whether Logical Data Link Control (LDLC) over IP data is included in the formatted communications trace.

***NO** LDLC over IP data is not formatted.

***YES** Only LDLC over IP data is formatted.

Top

Examples

```
PRTC MNTRC  CFGOBJ(*QESLINE)  CFGTYPE(*LIN)
```

This command prints communications trace data for line description QESLINE.

Top

Error messages

*ESCAPE Messages

360 System i: Programming i5/OS commands Starting with INZPCS (Initialize Client Access/400)

CPF2634
Not authorized to object &1.

CPF39AF
Trace is ending - please wait

CPF39A7
Trace storage not available in communications processor

CPF39A8
Not authorized to communications trace service tool

CPF39A9
Error occurred during communications trace function

CPF39BA
Formatting options selected not valid

CPF39BB
Communications trace data not printed

CPF39BC
Communications trace print request cannot be completed

CPF39B0
No communications traces exist.

CPF39B1
Trace &1 type &2 does not exist

CPF39B3
Trace &1 type &2 contains no data

CPF39B4
Trace data for &1 type &2 cannot be printed

CPF39B5
Communications trace data not printed

CPF39B6
Communications trace function cannot be performed

CPF39B7
Trace data for &1 type &2 cannot be printed

CPF39B8
No SNA data found in trace &1 type &2

CPF39B9
No trace records found for printing trace &1 type &2

CPF39C4
IP address not valid.

CPF3CF2
Error(s) occurred during running of &1 API.

CPF9803
Cannot allocate object &2 in library &3.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9860

Error occurred during output file processing.

CPF9872

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

CPF98A2

Not authorized to &1 command or API.

CPFA0D4

File system error occurred. Error number &1.

Top

Print Device Addresses (PRTDEVADR)

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

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

The Print Device Addresses (PRTDEVADR) command provides a printed list of addresses and related information for devices attached to a local or remote work station controller. For each device attached to the local work station controller named in the controller description (CTLD parameter), the output shows the device's name, its port and switch setting, its type and model number, its shared session number (valid only if device type is 3486 or 3487), and whether the device is a display station or printer.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
CTLD	Controller description	<i>Name</i>	Required, Positional 1

[Top](#)

Controller description (CTLD)

Specifies the name of the local or remote work station controller for which device address information is printed.

This is a required parameter.

[Top](#)

Examples

```
PRTDEVADR CTLD(CTL01)
```

This command prints device address information for the devices that are attached to the CTL01 work station controller.

[Top](#)

Error messages

*ESCAPE Messages

CPF2602

Controller &1 not found.

CPF2625

Not able to allocate object &1.

CPF2628

Device description previously deleted.

CPF263B

Controller &1 not a work station controller.

CPF2634

Not authorized to object &1.

CPF2778

Controller description &1 damaged.

CPF9846

Error while processing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

[Top](#)

Print Directory Information (PRTDIRINF)

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

Parameters
 Examples
 Error messages

The Print Directory Information (PRTDIRINF) command is used to print directory information for objects in the Integrated File System that was collected by the Retrieve Directory Information (RTVDIRINF) command. A spooled file with file name QPEZDIR goes to the spool queue associated with the job using this command.

Restrictions:

- You must have all object (*ALLOBJ) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
RPTTYPE	Type of report	*DIR, *OBJ, *OWN	Required, Positional 1
INFFILEPFX	Information file prefix	Simple name, <u>*LAST</u>	Optional
INFLIB	Information library	Name, <u>*LAST</u>	Optional
OWNER	Owners	Generic name, name, <u>*ALL</u>	Optional
OBJ	Objects	Path name, <u>*ALL</u> , *NONE	Optional
OBJTYPE	Object types	Single values: <u>*ALL</u> Other values (up to 60 repetitions): *ALRTBL, *AUTL, *BLKSF, *BNDDIR, *CFGL, *CHTFMT, *CHRSF, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DDIR, *DEVD, *DIR, *DOC, *DSTMF, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FIFO, *FILE, *FLR, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *MBR, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *NTBD, *NWID, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SOCKET, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *STMF, *SVRSTG, *SYMLNK, *S36, *TBL, *TIMZON, *USRIDX, *USRPRE, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
SUBTREE	Directory subtree	<u>*ALL</u> , *NONE	Optional
MINSIZE	Smallest size	0-99999, <u>0</u>	Optional
SORT	Sort by	<u>*SIZE</u> , *OWNER, *LSTCHG, *NAME	Optional
INCPATH	Include path names	<u>*ALL</u> , *NONE	Optional

Top

Type of report (RPTTYPE)

Specifies the type of report to print.

This is a required parameter.

- *DIR** A report of space being used by each directory is printed. Information will include a total size for the outermost directory and all nested subdirectories. Information for each subdirectory will include the total of space for just the subdirectory and will not include space for objects in any directories contained in the subdirectory.
- *OBJ** A report of object information for files and directories is printed.
- *OWN**
A report of the user profile (owner) information for files and directories is printed.

Top

Information file prefix (INFFILEPFX)

Specifies the file name prefix of the database files that were created by the Retrieve Directory Information (RTVDIRINF) command to store the retrieved directory information.

*LAST

The database files created by the most recent invocation of the RTVDIRINF command will be used. Informational message CPI1E31 will be sent to the job log and will contain the name and library of the files used.

information-file-prefix

Specify the same file prefix as was specified on a previous invocation of the RTVDIRINF command. The RTVDIRINF command created multiple database files to store the retrieved directory information. If a file prefix is specified, a value other than *LAST must be specified for the INFLIB parameter.

Top

Information library (INFLIB)

Specifies the library that contains the database files that were created by the Retrieve Directory Information (RTVDIRINF) command to store the retrieved directory information.

*LAST

The library used by the most recent invocation of the RTVDIRINF command will be used to find the database files which contain the retrieved directory information. Informational message CPI1E31 will be sent to the job log and will contain the name and library of the files used.

library-name

Specify the name of the library that contains the database files created by a previous invocation of the RTVDIRINF command. If a library name is specified, a value other than *LAST must be specified for the INFFILEPFX parameter.

Top

Owners (OWNER)

Specify the names of the owners (user profiles) of the objects to print information about.

- *ALL** The report contains information about objects owned by any user profile.

owner-name

Specify the user profile that owns the objects to print information about.

generic-name

Specify the generic user profile that owns the objects to print information about. 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 name specifies all user profiles with names that begin with the generic prefix.

Top

Objects (OBJ)

Specifies the names of the objects to print information about.

***ALL** If you specify an owner (OWNER parameter), all objects owned by the specified owner are included. If OWNER(*ALL) is specified, all objects are included in the report.

***NONE**

No detail object information is included in the report, just a total size of owned objects, if *NONE is specified for Object (OBJ) parameter and *OWN is specified for the Type of report (RPTTYPE) parameter.

object-name

The object information is included only for the objects specified by the given name. If an owner is specified (OWNER parameter), only the objects meeting the owner criteria and that match the given name are included.

generic-name

The object information is included only for the objects that match the specified generic name. 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 name specifies all objects with names that begin with the generic prefix. If an owner is specified (OWNER parameter), only the objects meeting the owner criteria and that match the given generic name are included.

Top

Object types (OBJTYPE)

Specifies the object types to print information about. If the OWNER or OBJ parameters were specified with a value other than *ALL, the owner and object name criteria must also be met.

Single values

***ALL** Information about objects of all types is printed.

Other values (up to 60 repetitions)

object-type

Information about objects of the type(s) specified is printed.

Top

Directory subtree (SUBTREE)

Whether to print subdirectories statistics in the report or not.

***ALL** Statistics are included for all subdirectories within the directory processed by the Retrieve Directory Information (RTVDIRINF) command.

***NONE**

Statistics are included only for objects in the directory processed by the Retrieve Directory Information (RTVDIRINF) command.

Top

Smallest size (MINSIZE)

Specifies the size of the smallest object to include.

0 All objects are included regardless of size.

size Specify size in number of kilobytes.

Top

Sort by (SORT)

Specifies the order in which the information should be sorted.

***SIZE** Information is sorted by object size, from largest to smallest.

***OWNER**

The information is sorted in alphabetical order by owner name.

***LSTCHG**

The information is sorted by last-change date with the oldest information first.

***NAME**

Information is sorted by object name, names are listed in alphabetical order.

Top

Include path names (INCPATH)

Specifies if the report will include the path where the objects reside. If *NONE is specified for Object (OBJ) parameter and *OWN is specified for the Type of report (RPTTYPE) parameter, this parameter will be ignored.

***ALL** Paths are included in the report.

***NONE**

Paths are not included in the report.

Top

Examples

Example 1: Print Information, Grouped by Owner, for Most Recent RTVDIRINF

```
PRTDIRINF RPTTYPE(*OWN) INFFILEPFX(*LAST) INFLIB(*LAST)
          OWNER(*ALL) OBJ(*ALL) OBJTYPE(*ALL) SORT(*SIZE)
```


This command prints an owner report from the database file created by the most recent invocation of the Retrieve Directory Information (RTVDIRINF) command. Information in the report will be included for all objects, grouped by their owner. The information is sorted by object size and sent to the printer file QPEZDIR.

Example 2: Print a Directory Report

```
PRTDIRINF  RPTTYPE(*DIR)  INFFILEPFX(MYROOTDIR)
           INFLIB(QUSRSYS)  SUBTREE(*ALL)
```

This command prints a directory report from database file MYROOTDIRO and MYROOTDIRD created by a prior invocation of the Retrieve Directory Information (RTVDIRINF) command. If information was collected for subdirectories by the RTVDIRINF command, that information will be included in the directory report. The information is sent to the printer file QPEZDIR.

Top

Error messages

*ESCAPE Messages

CPF2110

Library &1 not found.

CPF1ED2

File &1 is in use and cannot be accessed.

CPF1ED5

File prefix &1 or library &2 not found in QAEZDBFILE.

CPF1ED6

File &1 in library &2 not found.

CPF1ED9

Retrieved directory information not complete.

CPF1EEC

Not authorized to file &1.

CPF1E99

Unexpected error occurred.

Top

Print Document (PRTDOC)

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

Parameters
 Examples
 Error messages

The Print Document (PRTDOC) command permits the user to print a document using the word processing function of OfficeVision.

This command also permits the user to override all print option values that are currently stored with a document. When a document is created, a set of default print options is associated with that document. If the user wants to override one or more of the parameters in this print command, the user must select OPTIONS(*YES) so that the print options appear on the display. When the print options appear, any of the print parameters can be changed. The user can override one or all of the print option parameters with this command.

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

Top

Parameters

Keyword	Description	Choices	Notes
DOC	Document	Character value, <u>*PRV</u> , *ALL	Optional, Key, Positional 1
FLR	Folder	Character value, <u>*PRV</u>	Optional, Key, Positional 2
OPTIONS	Display print options	<u>*NO</u> , *YES, *PRTFILE, *OUTFILE	Optional, Key
PRTFILE	Printer file	Qualified object name	Optional
	Qualifier 1: Printer file	Name, <u>QSYSPRT</u>	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
OUTFILE	File to receive output	Single values: <u>*PRV</u> Other values: Qualified object name	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, <u>*FIRST</u> , *PRV	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
CURSTS	Current status value	Character value, <u>*PRV</u> , *NOCHK	Optional
NEWSTS	New status value	Character value, <u>*PRV</u> , *NOCHG	Optional
OUTDTATYP	Type of data for output	<u>*PRV</u> , *ALL, *IDP	Optional
DLTDOC	Delete document	<u>*NO</u> , *YES	Optional
OUTPUT	Output device	<u>*SAME</u> , *PRINT, *	Optional
DEV	Print device	Name, <u>*SAME</u> , *USRPRF, *SYSVAL, *WRKSTN	Optional

Keyword	Description	Choices	Notes
OUTQ	Output queue	Single values: *SAME, *FILE, *DEV, *WRKSTN Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Output queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL	
SPLFILE	Output file	<i>Name</i> , *SAME, *DOC, *FILE	Optional
FORMTYPE	Form type	<i>Character value</i> , *SAME, *STD	Optional
COVERPAGE	Print separator page	*SAME, *YES, *NO	Optional
PRTQLTY	Print quality	*SAME, *LETTER, *TEXT, *DRAFT	Optional
COPIES	Number of copies	1-99, *SAME	Optional
DUPLEX	Print on both sides	*SAME, *YES, *NO, *TUMBLE	Optional
AUTOBIND	Automatic page binding	*SAME, *YES, *NO	Optional
HOLD	Delay printing	*SAME, *YES, *NO	Optional
PRTERLOG	Print document error log	*PRV, *YES, *NO	Optional
ERRFORM	Error log form type	<i>Character value</i> , *SAME, *STD	Optional
LARGEPRINT	Large print	*SAME, *YES, *NO	Optional
MRGTYPE	Merge type	*SAME, *QRY, *DOC, *FILE, *BLANK	Optional
QRYDFN	Query	Single values: *SAME Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Query	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL	
DTADOC	Data document	<i>Character value</i> , *SAME	Optional
DTAFLR	Data folder	<i>Character value</i> , *SAME	Optional
DTAFILE	Data file	Single values: *SAME Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Data file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DTAMBR	Data member	<i>Name</i> , *SAME, *FIRST, *FILE, *LAST	Optional
MLTLINRPT	Multiple line report	*SAME, *YES, *NO	Optional
ADJLINES	Adjust line endings	*SAME, *YES, *NO	Optional
ADJPAGES	Adjust page endings	*SAME, *YES, *NO	Optional
ALWWIDOW	Allow widow lines	*SAME, *YES, *NO	Optional
RENUMBER	Renumber system page numbers	*SAME, *YES, *NO	Optional
PRTCHGSYM	Print change symbols	*SAME, *YES, *NO	Optional
SYMBOLS	Change symbols to print	<i>Character value</i> , *SAME	Optional
DRAFTSPACE	Draft spacing	*SAME, *YES, *NO	Optional
LINNBR	Print line numbers	*SAME, *YES, *NO	Optional
RESOLVE	Resolve instructions	*SAME, *YES, *NO	Optional
LEFTSPACES	Additional spaces to left	0-99, *SAME	Optional
CHRID	Character identifier	<i>Element list</i>	Optional
	Element 1: Graphic character set	1-9999, *SAME, *BLANK	
	Element 2: Code page	1-9999	
SAVOUTPUT	Save resolved output	*SAME, *YES, *NO	Optional
SAVDOC	Resolved output document	<i>Character value</i> , *SAME, *BLANK	Optional
SAVFLR	Resolved output folder	<i>Character value</i> , *SAME, *BLANK	Optional
JOBQ	Place on job queue	*SAME, *YES, *NO	Optional

Keyword	Description	Choices	Notes
JOBDB	Job description	Single values: *SAME Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Job description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i>	
SNDMSG	Send completion message	*SAME, *YES, *NO	Optional
CNLERR	Cancel on error	*SAME, *YES, *NO	Optional
STRPAGE	Start page	0.01-9999.99, *FIRST, *LAST, *PAGERANGE, *SAME	Optional
ENDPAGE	End page	0.01-9999.99, *FIRST, *LAST, *STRPAGE, *PAGERANGE, *SAME	Optional
PAGERANGE	Page ranges	Single values: *SAME Other values (up to 7 repetitions): <i>Element list</i>	Optional
	Element 1: Start page	0.01-9999.99, *FIRST, *LAST	
	Element 2: End page	0.01-9999.99, *FIRST, *LAST, *STRPAGE	
LBLACROSS	Number of labels across page	1-99, *SAME	Optional
LBLWIDTH	Width of labels	2-198, *SAME	Optional
SHEETFEED	Sheet feed labels	*SAME, *YES, *NO	Optional
LBLDOWN	Number of rows per sheet	1-99, *SAME	Optional
SHFLEFTMAR	Shift left margin	*SAME, *YES, *NO	Optional

Top

Document (DOC)

Specifies the name of the document that is printed.

***PRV** The name used in the previous session is used.

***ALL** All documents to which the user is authorized are printed to a database file. This is valid only when the output is directed to an OUTFILE.

document-name

Specify the name of the document to be printed.

Top

Folder (FLR)

Specifies the name of the folder that contains the document that is printed.

***PRV** The name used in the previous session is used.

folder-name

Specify the name of the folder that contains the document being printed.

Top

Display print options (OPTIONS)

Specifies whether the print options for this document are displayed before the document is printed.

***NO** The print options are not displayed before the document is printed.

***YES** The print options are displayed before the document is printed.

***PRTFILE**

The print options specified on the **Printer file** prompt (PRTFILE parameter) are used.

***OUTFILE**

The document is printed to the database file specified on the **File to receive output** prompt (OUTFILE parameter).

Top

Printer file (PRTFILE)

Specifies the printer file to use for the print options. This parameter is valid only if *PRTFILE is also specified on the **Display print options** prompt (OPTIONS parameter).

When *PRTFILE is specified, the following Print Options are overridden by the appropriate values in the printer file:

- DEV (Printer ID)
- PRTQLTY
- OUTQ
- FORM
- COPIES
- HOLD
- DUPLEX
- OUTPUT FILE

QSYSPRT

The document is printed using the system printer device file. This value overrides the printer name specified in the print options associated with the document.

printer-device-file-name

Specify the name and library of the printer device file that is used for the print document request. This value overrides the printer file name specified in the print options associated with the document.

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 name of the library where the printer device file is located.

Top

File to receive output (OUTFILE)

Specifies the name of the database file in which the displayed information is stored. If the specified file does not exist, this command creates a database file and file member. If the file is created, the public authority for the file is the same as the 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. Output to OUTFILE is supported only if *OUTFILE is specified in the **Display print options** prompt (OPTIONS parameter).

***PRV** The library and database file used in the previous (last) PRTDOC request for this user is used.

data-base-file-name

Specify the qualified name of the database file in which the resolved document information is stored. If no file is found by that name, a file and member by that name are created and stored in the specified library, or in *CURLIB, if no library is 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 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 file is located.

Top

Output member options (OUTMBR)

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

The possible **member to receive output** values are:

***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.

member-name

Specify the name of the file member that is to receive the output. If a file member is specified that does not exist, the system creates it.

***PRV** The member used in the previous (last) PRTDOC request for this user is used for this request.

The possible **replace or add records** values are:

***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

Current status value (CURSTS)

Specifies the value the document Interchange Document Profile (IDP) status field must have before the document may be printed to the database file. This field is 20 characters long and is valid only if OUTFILE output is requested.

***PRV** The value used in the previous (last) PRTDOC request for this user is used.

***NOCHK**

The status field is not checked before printing this document to the database file.

value Specify the value that the status field must equal before the document is printed to the database file.

New status value (NEWSTS)

Specifies the value the document Interchange Document Profile (IDP) status field is set to after the document has been printed to the database file. If a value is specified on the **New status value** prompt (NEWSTS parameter), you must have at least *CHANGE authorization to the document. This field is 20 characters long and is valid only if OUTFILE output is requested.

***PRV** The value used in the previous (last) PRTDOC request for this user is used.

***NOCHG**

The status field is not changed after printing this document to the database file.

value Specify the value to which the status field is set after the document is printed to the database file.

Top

Type of data for output (OUTDTATYP)

Specifies whether the entire document, or only the Interchange Document Profile (IDP) information, is printed to the database file.

***PRV** The value used in the previous (last) PRTDOC request for this user is used.

***ALL** The entire document is printed to a database file.

***IDP** Only the Interchange Document Profile (IDP) is printed to a database file.

Top

Delete document (DLTDOC)

Specifies whether the document is deleted after it has been printed to the database file.

***NO** The document is not deleted after being printed to the database file.

***YES** The document is deleted after being printed to the database file.

Note: You must be the owner of the document or have *ALL authority to delete it.

Top

Output device (OUTPUT)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

***SAME**

The output device does not change.

***PRINT**

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

***** Your resolved document is shown on the display. A resolved document is a document with the text instructions processed.

Top

Print device (DEV)

Specifies the name of the printer.

*SAME

The printer does not change.

*USRPRF

The printer ID specified in your user profile is used to print the document.

*SYSVAL

The system printer is used to print the document.

*WRKSTN

The printer assigned to the user's work station is used to print the document.

printer-name

Specify the name of the printer you want to use to print the document.

Top

Output queue (OUTQ)

Specifies the name of the output queue. It must already exist.

*SAME

The output queue does not change.

*DEV The output queue associated with the printer specified on the **Print device** prompt (DEV parameter) is used.

*FILE The output queue and output queue library values are taken from one of the following:

1. If the **Printer file** prompt (PRTFILE parameter). is specified, values from the specified printer device file are used.
2. If the **Printer file** prompt (PRTFILE parameter). is not specified, values from the Printer File Prompt on the document print options are used.

*WRKSTN

The output queue assigned to the user's work station is used.

output-queue-name

Specify the name and library of the output queue that holds your output until it is ready to print.

The possible library values are:

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

library-name

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

Top

Output file (SPLFILE)

Specifies the name of the output file.

*SAME

The output file does not change.

*FILE The output file name is the name of the printer file used.

***DOC** The document name is used for the spooled file name. However, if the document name is longer than 10 characters, or contains a period, then the spooled file name is QSYSPRT.

spool-file-name

Specify the name of the file in which you want the output stored while it is on the output queue.

Top

Form type (FORMTYPE)

Specifies the type of forms used in the printer.

***SAME**

The forms type does not change.

***STD** The standard printer form for your computer system is used.

form-type

Specify the forms type for the type of paper on which the output is printed.

***BLANK**

No special form type is specified.

Top

Print separator page (COVERPAGE)

Specifies whether a cover page is printed that includes such things as the document name, folder name, document description, subject, reference, and author name.

***SAME**

The cover page value does not change.

***YES** A cover page is printed.

***NO** A cover page is not printed.

Top

Print quality (PRTQLTY)

Specifies the type of print quality that is used to print your document.

***SAME**

The print quality value does not change.

***LETTER**

Your document is printed in letter quality type.

***TEXT**

Your document is printed in text quality type. This is better quality than *DRAFT but not as good as *LETTER.

***DRAFT**

Your document is printed in draft quality type.

Top

Number of copies (COPIES)

Specifies the number of copies of your document you want to print.

***SAME**

The copies value does not change.

value Specify a number, ranging from 1 through 99, for the number of copies of your document you want to print.

Top

Print on both sides (DUPLEX)

Specifies whether output is printed on one side or two sides of the paper.

***SAME**

The value does not change.

***YES** The output is printed on both sides of the paper, with the top of each printed page at the same end of the sheet of paper.

***TUMBLE**

The output is printed on both sides of the paper, with the top of one printed page at the opposite end from the top of the other printed page.

***NO** The output is printed on one side of the paper.

Top

Automatic page binding (AUTOBIND)

Specifies whether the left and right margins of alternating pages are adjusted to allow for page binding.

***SAME**

The autobind option does not change.

***YES** The margins are adjusted to allow for page binding.

***NO** The margins are not adjusted to allow for page binding.

Top

Delay printing (HOLD)

Specifies whether the printing of your documents is put on hold. The documents are held on the output queue, where you can release them to print, or delete them if you do not want them to print. You can print a group of documents together by putting them on the output queue before releasing them to print.

***SAME**

The hold value does not change.

***YES** The printing is delayed for the specified documents.

***NO** Your documents begin printing when the printer is ready.

Top

Print document error log (PRTERLOG)

Specifies whether to include the document error log as part of the information printed with the document.

- *PRV The value used in the previous (last) PRTDOC request for this user is used for this request.
- *YES The error log is printed to the output device.
- *NO The error log is not printed to the output device.

Top

Error log form type (ERRFORM)

Specifies the forms type for the type of paper on which the error log is printed.

*SAME

The error form value does not change.

- *STD The error log is printed on the paper specified in the printer file for the printer you selected.

error-form-name

Specify the name of the forms on which the error log is printed.

*BLANK

No special error log form type is specified.

Top

Large print (LARGEPRINT)

Specifies whether your document is printed using large print.

*SAME

The large print value does not change.

- *YES Your document is printed using large print.
- *NO Your document is not printed using large print.

Top

Merge type (MRGTYPE)

Specifies where data being merged is stored.

*SAME

The merge source does not change.

- *QRY Data requested in a query is merged. A query is a request to select and copy one or more records from a file based on defined conditions.
- *DOC Data stored in a document is merged.
- *FILE Data stored in a file is merged.
- *BLANK
No data will be merged.

Top

Query (QRYDFN)

Specifies the name of the query that is run to pass the data being merged. A query is a request to select and copy one or more records from a file based on defined conditions. *QRY must be specified on the **Merge type** prompt (MRGTYPE parameter)..

*SAME

The query name does not change.

query-definition-name

Specify the name of the query that is used to move the data being merged.

The possible library values are:

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

library-name

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

Top

Data document (DTADOC)

Specifies the name of the document that contains the data being merged. *DOC must be specified on the **Merge type** prompt (MRGTYPE parameter).

*SAME

The document name does not change.

document-name

Specify the name of the document using 1 to 12 alphanumeric characters. If you use more than 8 characters, the ninth character must be a period (.) followed by a 1- to 3-character extension.

Top

Data folder (DTAFLR)

Specifies the name of the folder that contains the document being merged. *DOC must be specified on the **Merge type** prompt (MRGTYPE parameter).

*SAME

The folder name does not change.

folder-name

Specify the name of the folder that contains the document being merged.

Top

Data file (DTAFILE)

Specifies the name of the file in which the member that contains data to be merged is located. *FILE must be specified on the **Merge type** prompt (MRGTYPE parameter).

*SAME

The data file name does not change.

file-name

Specify the name of the file that contains the data to be merged.

The possible library values are:

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

library-name

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

Top

Data member (DTAMBR)

Specifies the name of the file member that contains the data to be merged. This parameter is valid only when MRGTYPE(*FILE) is specified.

***SAME**

The file member does not change.

***FIRST**

The first member in the file contains the data to be merged.

***FILE** The member with the same name as the file contains the data to be merged.

***LAST**

The last member in the file contains the data to be merged.

member-name

Specify the name of the file member that contains the data being merged. A member is a set of data within a file.

Top

Multiple line report (MLTLINRPT)

Specifies whether a multiple line report is created. A multiple line report is created by merging data field instructions. This creates a report in which each record of data produces several lines of output.

***SAME**

The multiple line report option does not change.

***YES** A multiple line report is created.

***NO** A multiple line report is not created.

Top

Adjust line endings (ADJLINES)

Specifies whether the line endings in the printed document are adjusted. The lines are adjusted according to what is specified on the Line Spacing/Justification display. This is useful when you print a document that has data merged into it, has instructions, has display attributes that do not print as spaces, or uses a proportionally spaced font.

***SAME**

The line endings values do not change.

***YES** Adjusts line endings in the printed document.

***NO** Does not adjust the line endings in the printed document. This is useful if you have typed text exactly as you want it printed.

Top

Adjust page endings (ADJPAGES)

Specifies whether the page endings in the printed document are adjusted. The pages are determined by what is specified for the first typing line and last typing line prompts on the Page Layout/Paper Options display.

*SAME

The page endings value does not change.

***YES** Page endings in the printed document are adjusted.

***NO** Page endings in the printed document are not adjusted.

Top

Allow widow lines (ALWWIDOW)

Specifies whether the page endings are determined by the exact number of lines per page specified on the Page Layout/Paper Options display.

*SAME

The allow widow lines value specified in the print document options does not change.

***YES** Page endings are determined by the exact number of lines per page.

***NO** Page endings are not determined by the exact number of lines per page.

Top

Renumber system page numbers (RENUMBER)

Specifies whether the page numbers are renumbered when the document is printed.

*SAME

The value does not change.

***YES** The page numbers are renumbered when the document is printed.

***NO.** The page numbers are not renumbered when the document is printed.

Top

Print change symbols (PRTCHGSYM)

Specifies whether change symbols are printed in the left margin on your document. Change symbols are used to indicate lines that have been revised.

*SAME

Print change symbol value does not change.

***YES** Change symbols are printed in the left margin of your document.

***NO** The change symbols are not printed in the left margin of your document.

Top

Change symbols to print (SYMBOLS)

Specifies that up to 5 change symbol characters may appear in the left margin of the printed document. If your document contains more than one change symbol character and you do not select which change symbol characters you want to print, all change symbol characters specified in your document are printed.

***SAME**

The change symbol value does not change.

value Specify up to 5 change symbol characters to appear in the left margin of the printed document.

Top

Draft spacing (DRAFTSPACE)

Specifies whether the spacing value can be adjusted for your document. For example, if the Line spacing prompt is 3 (Triple), then the doubled spacing value is 6, and five blank lines are printed between each line of text in your document. The document is still paginated using the value in the Line spacing prompt; so, depending on the amount of text on a page, one page may print over two pages.

***SAME**

The draft spacing value does not change.

***YES** The spacing value for your document is doubled.

***NO** The spacing value that exists in the Line spacing prompt on the Line Spacing/Justification display is used.

Top

Print line numbers (LINNBR)

Specifies whether line numbers are printed in your document. The line numbers begin with 1 on the first page of your document. Line numbers are not printed in headers or footers.

***SAME**

The line numbers value does not change.

***YES** Line numbers are printed in your document.

***NO** Line numbers are not printed in your document.

Top

Resolve instructions (RESOLVE)

Specifies whether the instructions that you have placed in your document are processed. For example, the Date instruction (.date) is resolved to the actual date (04/03/62).

***SAME**

The resolve value does not change.

***YES** The instructions you have placed in your document are processed.

***NO** The instructions you have placed in your document are not processed. For example, the Date instruction (.date) is printed as *date.

Top

Additional spaces to left (LEFTSPACES)

Specifies whether the left margin is increased.

*SAME

The left spaces value does not change.

value Specify a number, ranging from 0 through 99, for the number of spaces that are added to the left margin in your printed document.

Top

Character identifier (CHRID)

Specifies the graphic character set ID that is used to print your job. A graphic character set ID is an identifier that is used to specify a set of graphic characters in a code page. The graphic character set ID selected overrides the automatic value set by the system for the specific printer.

*SAME

The character set code page value does not change.

character-set

Specify the character set that is used to print your job. Up to 4 digits can be specified for the character set.

code-page

A code page is a particular assignment of hexadecimal identifiers to graphic characters. Up to 4 digits can be specified for the code page.

*BLANK

No special character set code page is used.

Top

Save resolved output (SAVOUTPUT)

Specifies whether the document you are printing is also saved as a final form document.

*SAME

The save resolved output value does not change.

*YES The printed document is saved as a final form document.

*NO The printed document is not saved as a final form document.

Top

Resolved output document (SAVDOC)

Specifies the name of the document that contains the final form document.

*SAME

The save document name does not change.

document-name

Specify the name of the document that contains the final form document. The document name ranges from 1 through 12 alphanumeric characters. If you use more than 8 characters, the ninth character must be a period (.) followed by a 1- to 3-character extension. If the document name you specify does not already exist, the document is created for you.

***BLANK**

A resolved output document is not specified.

Top

Resolved output folder (SAVFLR)

Specifies the name of the folder that contains the document being saved in final form.

***SAME**

The save folder value does not change.

folder-name

Specify the name of the folder that contains the document.

***BLANK**

A resolved output folder is not specified.

Top

Place on job queue (JOBQ)

Specifies whether the print request is put on the job queue.

***SAME**

The job queue value does not change.

***YES** The printing of the document is placed on the job queue.

***NO** The printing of the document is not placed on the job queue.

Top

Job description (JOBDD)

Specifies the name of the job description that describes how the job is run.

***SAME**

The job description value does not change.

job-description-name

Specify the name of the job description that describes how the job is run.

The possible library values are:

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

library-name

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

Top

Send completion message (SNDMSG)

Specifies whether you are putting your print job on the job queue and want a message sent to you when the job has completed.

***SAME**

The send message value does not change.

- *YES A message is sent to you when the print job has completed.
- *NO A message is not sent to you when the print job has completed.

Top

Cancel on error (CNLERR)

Specifies whether printing is stopped on your document if an error is detected.

*SAME

The cancel error value does not change.

- *YES Printing is stopped on your document if an error is detected. The error is listed in the error log with an error message stating that the job is canceled.
- *NO Printing continues on your document even if an error is detected.

Top

Start page (STRPAGE)

Specifies the page number on which you want printing to start.

Note: If the STRPAGE(*page-number*) value specified is larger than the ENDPAGE(*page-number*) value specified, the entire document is printed.

*PAGERANGE

The pages specified on the PAGERANGE parameter are printed.

*SAME

The start page value does not change.

*FIRST

Printing is started on the first page of the document.

*LAST

Printing is started on the last page of the document.

page-number

Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.

Top

End page (ENDPAGE)

Specifies the page number on which you want printing to stop.

*PAGERANGE

The pages specified on the PAGERANGE parameter are printed.

*SAME

The end page value does not change.

*FIRST

Printing is ended after the first page of the document.

*LAST

Printing is ended after the last page of the document.

***STRPAGE**

The end page value is the same as the start page value. Only one page is printed.

page-number

Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.

Top

Page ranges (PAGERANGE)

Specifies the page ranges to print. A maximum of 7 ranges can be specified.

Note: If the STRPAGE(*page-number*) value specified is larger than the ENDPAGE(*page-number*) value specified, the entire document is printed.

***SAME**

The page range specified on the document print options is printed.

The possible **start page** values are:

***FIRST**

Printing is started on the first page of the document.

***LAST**

Printing is started on the last page of the document.

page-number

Specify the page on which to begin printing. Valid values range from 0.01 through 9999.99.

The possible **end page** values are:

***FIRST**

Printing is ended after the first page of the document.

***LAST**

Printing is ended after the last page of the document.

***STRPAGE**

The end page value is the same as the start page value. Only one page is printed.

page-number

Specify the page on which to stop printing. Valid values range from 0.01 through 9999.99.

Top

Number of labels across page (LBLACROSS)

Specifies the number of labels that are printed across a page.

***SAME**

The label across page value does not change.

value Specify the number of labels you want printed across a page. Valid values range from 1 through 99.

Top

Width of labels (LBLWIDTH)

Specifies the width (in number of characters) of the label. The width of a label is the number of characters from the left edge of the first label to the left edge of the next label, including the blank spaces between the labels. If the width you specify is larger than the margins for your document, the margins are used as the width.

*SAME

The label width value does not change.

value Specify the width (in number of characters) that you want the label to be. Valid values range from 2 through 198.

Top

Sheet feed labels (SHEETFEED)

Specifies, if you are sheet feed printing, whether you want more than one row of labels on a page. If you are using sheet feed paper, there is no other way to print more than one row of labels on a page.

*SAME

The sheet feed value does not change.

***YES** You are sheet feed printing and want more than one row of labels on a page.

***NO** You are not sheet feed printing, or you only want to print one row of labels on a page.

Top

Number of rows per sheet (LBLDOWN)

Specifies, if ***YES** was selected for the **Sheet feed labels** prompt (SHEETFEED parameter), the number of rows of labels that you want printed on a page.

*SAME

The label down value does not change.

value Specify the number of rows of labels that you want printed on a page. Valid values range from 1 through 99.

Top

Shift left margin (SHFLEFTMAR)

Specifies whether to shift the left margin to prevent text from being truncated.

*SAME

The SHFLEFTMAR value does not change.

***YES** When the text exceeds the right margin or the paper edge, the left margin is shifted so that as much text as possible is printed. If the text does not exceed the right margin or the paper edge, the text is not shifted.

***NO** The left margin is not shifted when text exceeds the right margin. Any text exceeding the right margin is truncated.

Top

Examples

Example 1: Printing to a File

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*OUTFILE)
        OUTFILE(MYFILE/MYLIB)  OUTMBR(MYMBR *REPLACE)
        CURSTS(*PRV)  NEWSTS(*PRV)
        OUTDTATYP(*PRV)  PRTERLOG(*PRV)  DLTD(*NO)
```

This command prints the document MYDOC in folder MYFLR to the database file MYFILE in library MYLIB in the database file member MYMBR. If the member already exists, it is replaced by the contents of MYDOC. The CURSTS, NEWSTS, OUTDTATYP, and PRTERLOG are taken from the last PRTDOC request. The document is not deleted after it is printed to the database file MYFILE.

Example 2: Printing a Document

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)
        DEV(MYPRNTR)  OUTQ(*DEV)
```

This command prints the document MYDOC in the folder MYFLR on a printer called MYPRNTR.

Example 3: Printing Document Error Log

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  PRTERLOG(*YES)
```

This command prints the document with a document error log attached to it.

Example 4: Increasing Margin

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  LEFTSPACES(10)
```

This command prints the document and has 10 extra spaces inserted in the left margin.

Example 5: Printing a Cover Page

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)  OPTIONS(*NO)  COVERPAGE(*YES)
```

This command prints the document with a cover page.

Example 6: Printing One Page to a File

```
PRTDOC  DOC(MYDOC)  FLR(MYFLR)
        OPTIONS(*OUTFILE)  OUTFILE(MYLIB/MYFILE)
        OUTMBR(*FIRST)  PAGERANGE((5 5))
```

This command prints page 5 of the document to the database file MYFILE in library MYLIB in the first member.

Top

Error messages

*ESCAPE Messages

CPF6C01

Error occurred during data stream transformation.

CPF6C03

Error occurred during document conversion.

CPF9012

Start of document interchange session not successful for &1.

CPF9801

Object &2 in library &3 not found.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

OFCFFFC

User storage capacity exceeded.

OFCFFFD

Damaged object found.

OFC8EA3

OfficeVision for AS/400 editor is not available to resolve to a display.

OFC8E01

Printer ID selected is not correct.

OFC8E1C

Cannot delay output when spooling is not active.

OFC8E1D

Printer for large print is not correct.

OFC8E2A

Output file member is in use.

OFC8E2B

Not authorized to output disk file or library.

OFC8E2C

Output disk file member could not be opened.

OFC8E30

Incorrect character set ID specified.

OFC8E38

Member is not a document output file member.

OFC8E4D

Output file name &9 is incorrect.

OFC8E50

Job has been canceled because of error.

OFC8E6B

Not authorized to output disk file member.

OFC8E6D

Could not access the output disk file member.

- OFC80B5**
OfficeVision for OS/400 editor is not available on the system.
- OFC800A**
Folder is in use.
- OFC800B**
Document &1 is in use.
- OFC800E**
&1 already exists as document or folder.
- OFC800F**
Display does not support text.
- OFC8006**
Folder not found.
- OFC8007**
Document &1 not found in folder.
- OFC8008**
Request not allowed with folder.
- OFC8009**
Request not allowed with document &1.
- OFC801A**
Document has been saved to diskette, tape or save file.
- OFC8010**
Document &1 cannot be processed.
- OFC8011**
Document &1 needs to be recovered.
- OFC8016**
Document &1 is checked out.
- OFC8018**
Document &1 is empty.
- OFC802C**
Label option specified with non-label document.
- OFC802D**
Option not allowed for PC editor.
- OFC8029**
Cannot save resolved output when printing a resolved document
- OFC820D**
Library &4 was not found.
- OFC820F**
Member &3 is in use.
- OFC947E**
Fill-in document &1 could not be opened.
- OFC9486**
Printer file or printer file library was not found.
- OFC960A**
&1 key was pressed by the user to end the PRTDOC function.

- OFC9609**
&1 is the resolved output file name for the print options function.
- OFC980B**
&9 documents printed, &10 documents not processed.
- OFC980C**
Error printing document &1 to a file.
- OFC980D**
Error converting document &1.
- OFC980E**
Error converting document &1.
- OFC980F**
Could not delete document &1 from folder.
- OFC9801**
Document &1 could not be opened.
- OFC9802**
Folder could not be opened.
- OFC9806**
No documents were selected for printing.
- OFC9808**
Document &1 does not have selected status.
- OFC9809**
Error log incorrect with document descriptions only.
- OFC9810**
Could not update status for document &1.
- OFC9811**
Folder needs to be reclaimed.

Top

Print Disk Information (PRTDSKINF)

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

Parameters
 Examples
 Error messages

The Print Disk Information (PRTDSKINF) command is used to print disk space information that was stored in database file QAEZDISK or QAEZDnnnnn by the Retrieve Disk Information (RTVDSKINF) command, where 'nnnnn' is the ASP number of the independent ASP for which disk space information was retrieved. The output with file name QPEZDISK goes to the spool queue associated with the job using this command.

Top

Parameters

Keyword	Description	Choices	Notes
RPTTYPE	Type of report	*LIB, *FLR, *OWN, *OBJ, *SYS	Required, Positional 1
ASPDEV	ASP device	Name, <u>*SYSBAS</u>	Optional
LIB	Libraries	Generic name, name, <u>*ALL</u>	Optional
OWNER	Owners	Generic name, name, <u>*ALL</u>	Optional
FLR	Folders	Generic name, name, <u>*ALL</u>	Optional
DOC	Documents	Generic name, name, *ALL, <u>*NONE</u>	Optional
OBJ	Objects	Generic name, name, *ALL, <u>*NONE</u>	Optional
OBJTYPE	Object types	Single values: <u>*ALL</u> Other values (up to 60 repetitions): *ALRTBL, *AUTL, *BLKSF, *BNDDIR, *CFGL, *CHTFMT, *CHRSF, *CLD, *CLS, *CMD, *CNL, *COSD, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DEACR, *DEADI, *DEADS, *DIR, *DIRCR, *DIRDS, *DOC, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FIFO, *FILE, *FLR, *FNTRSC, *FORMDF, *FTR, *GSS, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODL, *NODGRP, *NWID, *NWSCFG, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SRVPGM, *SSND, *STMF, *SVRSTG, *SOCKET, *SYMLNK, *S36, *TBL, *TIMZON, *USRIDX, *USRPRE, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
MINSIZE	Smallest size	Decimal number, <u>0</u>	Optional
SORT	Sort by	<u>*SIZE</u> , *OWNER, *LSTCHG, *LSTUSE, *NAME	Optional

Top

Type of report (RPTTYPE)

Specifies the type of report to print. The report information is taken from member QCURRENT in QAEZDISK or QAEZDnnnnn, where 'nnnnn' is the ASP number of the independent ASP for which disk space information was retrieved. If QCURRENT does not contain any data, an error message is sent.

This is a required parameter.

Restrictions:

- If option *FLR is specified for the RPTTYPE parameter, *SYSBAS must be specified for the ASPDEV parameter. Folders are not allowed on auxiliary storage pool (ASP) devices, they are only allowed on the system ASP and basic ASPs.

*LIB A report of the library information contained in the file is printed.

*FLR A report of the folder information contained in the file is printed.

*OWN

A report of the user profile (owner) information contained in the file is printed.

*OBJ A report of object information contained in the file is printed.

*SYS A report of only the system information contained in the file is printed.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device for which disk space information is to be printed.

*SYSBAS

Disk information for the system ASP and all basic ASPs is printed. File QAEZDISK in library QUSRSYS contains the disk space information that is to be printed.

name Specify the name of the ASP device for which disk space information is to be printed. File QAEZDnnnnn in library QUSRSYS contains the disk space information that is to be printed, where 'nnnnn' is the ASP number of the specified ASP device.

Top

Libraries (LIB)

Specifies the names of the libraries to print information about.

*ALL The report has information on all user libraries on the system.

name Specify the user library.

generic-name

Specify the generic library name.

Top

Owners (OWNER)

Specifies the names of the owners (user profiles) to print information about.

*ALL The report contains information on all user profiles on the system.

name Specify the name of a user profile.

generic-name

Specify the generic user profile name.

Top

Folders (FLR)

Specifies the names of the folders to print information about.

***ALL** The report has information on all user folders on the system.

name Specify the folder name.

generic-name

Specify the generic folder name.

Top

Documents (DOC)

Specifies the names of the documents to print information about.

***ALL** The report contains information on all documents in the specified folder.

name Specify the document by the given name within the specified folder.

generic-name

Specify the documents specified by the generic qualification.

Top

Objects (OBJ)

Specifies the names of the objects to print information about.

***ALL** If you specify a library or owner, then the object information is all objects within the library or those controlled by the owner.

***NONE**

No library or owner is specified.

name Specify a library or owner, then the object information is the object specified by the given name within the library or controlled by the owner.

generic-name

Specify a library or owner, then the object information are the objects that meet the specified generic qualification within the library or controlled by the owner.

Top

Object types (OBJTYPE)

Specifies the object types to print information about.

Single values

***ALL** If you specify a library or owner, information is printed on all the specified object types within the library or controlled by the owner. If an object name is specified, information on all object types with that name, within the library, or controlled by the owner is printed. If a library or

owner is not specified, the report has information on all object types on the system. If an object name is specified, information only on object types with that name is printed.

Other values (up to 60 repetitions)

object-type

Specify a library or owner, then the object type information is the object type specified within the library or controlled by the owner. If an object is specified, the report has information on the objects with the specified object type within the library or controlled by the owner.

Top

Smallest size (MINSIZE)

Specifies the size of the smallest piece of information to include. For example, if a library report is requested without objects, then this size would be the size of the smallest library to include. If objects within the library are requested, then this would be the size of the smallest object within the library to include.

0 All objects are included regardless of size.

size Specify size in thousands of bytes.

Top

Sort by (SORT)

Specifies the order in which the information should be sorted.

*SIZE Information is sorted from large to small.

*OWNER

The information is sorted in alphabetical order by owner name.

*LSTCHG

The information is sorted by last-change date with the oldest information first.

*LSTUSE

The information is sorted by last-use date with the oldest information first.

*NAME

Information is sorted in alphabetical order according to the report type.

Top

Examples

```
PRTDSKINF ASPDEV(*SYSBAS) RPTTYPE(*LIB) LIB(*ALL) OBJ(*ALL)
          SORT(*SIZE)
```

This command prints a library report from database file QAEZDISK in library QUSRSYS in member QCURRENT, containing information about all libraries, objects, and object types in the libraries. The information is sorted by size and sent to the printer file QPEZDISK.

Top

Error messages

*ESCAPE Messages

CPF1ED0

Current collection of disk space information not found.

CPF1ED1

Not authorized to collect disk space information.

CPF1ED2

File &1 is in use and cannot be accessed.

CPF1EEC

Not authorized to file &1.

CPF1E99

Unexpected error occurred.

[Top](#)

Print Error Log (PRERRLOG)

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

Parameters
 Examples
 Error messages

The Print Error Log (PRERRLOG) command is used primarily for problem analysis tasks. It places a formatted printer file of the data in the machine error log in a spooled printer device file named QPCSMPT or in a specified output file.

Restrictions:

- The following user profiles have private authorities to use the command:
 - QPGMR
 - QSYSOPR
 - QSRV
 - QSRVBAS

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of log data to list	<u>*ALL</u> , *ALLSUM, *ANZLOG, *MCH, *DEV, *ERRLOGID, *VOLSTAT	Optional, Positional 1
DEV	Logical device	Single values: <u>*ALL</u> Other values (up to 10 repetitions): <i>Name</i>	Optional
RSRCNAME	Resource name	Values (up to 10 repetitions): <i>Name</i>	Optional
ERRLOGID	Error log identifier	Values (up to 10 repetitions): <i>Hexadecimal value</i>	Optional
OUTPUT	Output	<u>*PRINT</u> , *OUTFILE	Optional
PERIOD	Time period for log output	<i>Element list</i>	Optional
	Element 1:	<i>Element list</i>	
	Element 1: Beginning time	<i>Time</i> , <u>*AVAIL</u>	
	Element 2: Beginning date	<i>Date</i> , <u>*CURRENT</u>	
	Element 2:	<i>Element list</i>	
	Element 1: Ending time	<i>Time</i> , <u>*AVAIL</u>	
	Element 2: Ending date	<i>Date</i> , <u>*CURRENT</u>	
PRTFMT	Print format	<u>*CHAR</u> , *HEX	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
VOLTYPE	Volume type	<i>Character value</i>	Optional
MODEL	Model number	<i>Character value</i>	Optional

Keyword	Description	Choices	Notes
VOL	Volume	Single values: *ALL Other values (up to 10 repetitions): <i>Character value</i>	Optional
VOLSTAT	Volume statistical data	*KEEP , *DLT	Optional
VOLSTATYP	Volume statistics type	*LIFETIME , *SESSION	Optional
SELECT	Error log entries to select	*ALL , *PRC , *MEDIA , *LWS , *CMN , *PWR , *LPP , *LIC	Optional
SORT	Sort by	*DATETIME , *TIME , *DEVADR , *ERRTYPE , *RSRCNAME	Optional

Top

Type of log data to list (TYPE)

Specifies the type of error log data from the machine error log to print in the spooled printer file.

***ALL** All the error codes in the machine's error log are printed. In addition, the error codes for each subsystem (for example, diskette units, printers) are printed in summary form.

***ALLSUM**

All the data in the error log is printed in summary form.

***ANZLOG**

A one-line summary is created for each entry in the error log.

***MCH** Only the error data produced by machine checks is printed.

***DEV** Only the error data produced by the devices specified in the following parameters is printed:

- **Logical device (DEV)** parameter
- **Resource name (RSRCNAME)** parameter

***ERRLOGID**

Only the error data with the specified error log record is printed. ***ERRLOGID** can only be specified if the **Error log identifier (ERRLOGID)** parameter is also specified. It is ignored for other request types.

***VOLSTAT**

Only the tape or diskette volume statistical data records are printed.

Note: If you specify ***PRINT** on the **Output (OUTPUT)** parameter and ***VOLSTAT** on this parameter, lifetime statistics are printed. If you specify ***OUTFILE** on the **OUTPUT** parameter and ***VOLSTAT** on this parameter, session statistics are directed to the output file. If the name of the volume is reported as '*****', it means that this volume is not displayable.

Top

Logical device (DEV)

Specifies the device names for which you want the error log data to be printed. This parameter is valid only if ***DEV** is specified for the **Type of log data to list (TYPE)** parameter. This parameter cannot be specified if a value is specified for the **Resource name (RSRCNAME)** parameter.

Single values

***ALL** The error log data for all device names is printed.

Other values

name Specify one or more device names whose error log data you want to print. A maximum of ten device names can be specified.

Top

Resource name (RSRCNAME)

Specifies the resource names for which error log entries are to be printed. This parameter is valid only if *DEV is specified for the **Type of log data to list (TYPE)** parameter. This parameter cannot be specified if a value is specified for the **Logical device (DEV)** parameter.

name Specify one or more resource names whose error log data you want to print. A maximum of ten resource names can be specified.

Note: If you specify a storage controller input/output processor (IOP) as the resource name, no error log entries are printed for the resource.

Top

Error log identifier (ERRLOGID)

Specifies that error log entries with the specified error log identifier are printed. This parameter is valid only if *ERRLOGID is specified for the **Type of log data to list (TYPE)** parameter. A maximum of ten error log identifiers can be specified.

hexadecimal-value

Specify the error log identifier of an error log entry to be printed.

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

Time period for log output (PERIOD)

Specifies the period of time for which the error log data is printed. The following values can be coded in this parameter, which contains two sets of two values each.

Note: This parameter is not valid when TYPE(*VOLSTAT) and VOLSTATTYP(*LIFETIME) are specified.

Element 1:

Element 1: Beginning time

*AVAIL

The error data that is available for the specified start or end date is printed.

- time* Specify the start time of the specified start date for which the error data is printed. The time is specified in 24-hour format with or without a time separator as follows:
- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
 - Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Element 2: Beginning date

*CURRENT

The error data that is available for the current day and between the specified start and end times (if specified) is printed.

- date* Specify the start date for which error data is printed. The date must be specified in the job date format.

Element 2:

Element 1: Ending time

*AVAIL

The error data that is available for the specified start or end date is printed.

- time* Specify the end time for the specified end date that specifies the error data to be printed. See the **Beginning time** description on this parameter for details about how time must be specified.

Element 2: Ending date

*CURRENT

The error data that is available for the current day and between the specified start and end times (if specified) is printed.

- date* Specify the end date for which error data is printed. The date must be specified in the job date format.

Top

Print format (PRTFMT parameter)

Specifies whether the indicated report prints any hexadecimal data in character format. This parameter cannot be specified if *VOLSTAT is specified for the **Type of log data to list (TYPE)** parameter, or if a value is specified for the **File to receive output (OUTFILE)** parameter.

*CHAR

The report is formatted so that hexadecimal data prints as character data.

*HEX No formatting is done for the report. Hexadecimal data prints as hexadecimal.

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 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

Volume type (VOLTYPE)

Specifies the volume type of the specified volume identifier. Valid entries are 4-digit device type numbers for cartridge tape, reel tape, or diskette. This parameter returns information about all the volumes that use the same technology as the tape device type that was specified. For example, if 6380 is the specified value for this parameter, information about all 1/4 inch tape cartridges on the system is returned.

character-value

Specify the volume type.

Top

Model number (MODEL)

Specifies the model number of the specified model type. This parameter is required if the device type is 9331 and TYPE(*VOLSTAT) is specified.

character-value

Specify the model number.

Top

Volume (VOL)

Specifies the name of the volume for which you want statistics processed.

Single values

***ALL** Volume statistics are processed for all volumes.

Other values

character-value

Specify the name of the volume for which statistics are processed. A maximum of ten volume names can be specified.

Top

Volume statistical data (VOLSTAT)

Specifies whether the volume statistical data records are kept or deleted from the machine error log after they are printed. This parameter is valid only if *VOLSTAT is specified on the **Type of data (TYPE)** parameter.

Note: ENDOPT(*UNLOAD) must be specified during the SAVE operation to generate volume statistics at the completion of the tape operation.

***KEEP**

The volume statistical data records are kept in the error log after they are printed.

***DLT** The volume statistical data records are deleted from the error log for volumes that are not active after they are printed.

Notes:

1. You cannot specify *DLT on this parameter if *OUTFILE is specified on the **Output (OUTPUT)** parameter.

2. The length of time it takes to run this command when VOLSTAT(*DLT) is specified is dependent on the number of volume IDs being deleted.

Top

Volume statistics type (VOLSTATTYP)

Specifies the type of volume statistics printed or directed to an output file. This parameter is valid only if *VOLSTAT is specified on the **Type of log data to list (TYPE)** parameter.

*LIFETIME

Lifetime statistics are printed. Lifetime statistics cannot be placed in an output file.

*SESSION

Session statistics are directed to the output file specified on the **File to receive output (OUTFILE)** parameter. Session statistics cannot be printed.

Top

Error log entries to select (SELECT)

Specifies which type of error log entries are included on the report.

*ALL All error log entries are included on the report.

*PRC The processor error log entries are included on the report.

*MEDIA

The error log entries for disk, tape, and diskette devices are included on the report.

*LWS The error log entries for local workstations are included on the report.

*CMN The error log entries for communications are included on the report. These include entries for communications I/O processors, I/O adapters, ports, lines, controllers, and devices connected with SDLC, ASYNC, BSC, X.25, IDLC, ISDN, and local area network line protocols.

*PWR The error log entries for system power control network (SPCN) are included on the report.

*LPP The error log entries for licensed programs are included on the report.

*LIC The error log entries for Licensed Internal Code are included on the report.

Top

Sort by (SORT)

Specifies the order in which the entries appear on the report.

*DATETIME

The entries are sorted by date and time. The summary entries are for each day.

*TIME

The entries are sorted by the time of day only. The summary entries are for each hour.

*DEVADR

The entries are sorted by the address of the device. The summary entries are divided into three levels: those for which the first two digits of the address are the same, those for which the first four digits of the address are the same, and those for which the first eight digits of the address are the same.

*ERRTYPE

The entries are sorted by the severity of the type of error. The more severe types of errors report at the top of the list. The summary entries are divided into two levels: those that have a common error type, and those that have a common error type and system reference code.

*RSRCNAME

The entries are sorted by the resource name of the device.

Top

Examples

Example 1: Printing Error Log Data

```
PRTERLOG
```

This command gets the error data in the machine error log that occurred for all device types and puts it in a spooled file. The entire error log is printed and any hexadecimal data is in character format.

Example 2: Using the System Resource Manager Database

```
PRTERLOG TYPE(*DEV) RSRCNAME(TAPE000001) PRTFMT(*HEX)
```

This command uses the system resource manager database to determine the device type, model, and serial number for the resource TAPE000001. The print request is based on that information. The report is put in the spooled file and contains all records that pertain to that device type, model, and serial number. Any hexadecimal data in the file is converted to hexadecimal format.

Example 3: Processing Error Log Entries

```
PRTERLOG TYPE(*DEV) DEV(DISKLU1) OUTPUT(*OUTFILE)  
OUTFILE(MYLIB/MYDBD) OUTMBR(ELOG)
```

This command processes all the error log entries for the logical device named DISKLU1. They are put in the file MYDBD, in the library MYLIB, and in the member ELOG. No spooled files are created.

Top

Error messages

*ESCAPE Messages

CPD36CA

OUTPUT(*OUTFILE) cannot be specified with DEV(*ALL).

CPD36C2

DEV and RSRCNAME cannot be used together.

CPD36C3

PRTFMT parameter not valid with TYPE(*VOLSTAT).

CPD36C4

OUTFILE not valid with PRTFMT parameter.

CPD36C5

RSRCNAME parameter can only be used with TYPE(*DEV) parameter.

CPD36C7

ERRLOGID valid only with TYPE(*ERRLOGID).

CPD36C9

PERIOD not valid for specified TYPE and VOLSTATYP.

CPF3535

Error log not available for printing.

CPF3541

No error log entries were found.

CPF3593

PERIOD parameter start time exceeds end time.

CPF3693

Service function ended because error occurred.

CPI36CA

Resource name &1 not found.

CPI36CC

No error log entries were found for &1 &2.

Top

Print Internal Data (PRTINTDTA)

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

Parameters
 Examples
 Error messages

The Print Internal Data (PRTINTDTA) command is used primarily for problem analysis tasks. It writes the machine internal data to a spooled printer file. The data is used to service the system.

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.
2. To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service dump function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_DUMP, can also be used to change the list of users that are allowed to perform dump operations.
3. The command must be issued from within the job with internal data being printed, or the issuer of the command must be running under a user profile which is the same as the job user identity of the job with internal data being printed, or the issuer of the command must be running under a user profile which has a job control (*JOBCTL) special authority. The job user identity is the name of the user profile by which a job is known to other jobs. It is described in more detail in the Work Management Guide.
4. This command is intended to be used only at the direction of your service representative.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of data	*DMP, *INTCFG, *NOTES, *JOB	Required, Positional 1
DMPID	Dump identifier	Character value, *NONE, *ALL	Optional
PERIOD	Time period for internal data	Element list	Optional
	Element 1: Start time and date	Element list	
	Element 1: Beginning time	Time, *AVAIL	
	Element 2: Beginning date	Date, *CURRENT	
	Element 2: End time and date	Element list	
	Element 1: Ending time	Time, *AVAIL	
JOB	Job name	Single values: *, *SVRTYPE Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	Name	
	Qualifier 2: User	Name	
	Qualifier 3: Number	000000-999999	
SLTTHD	Thread ID to include	Single values: *ALL, *SELECT Other values (up to 20 repetitions): <i>Hexadecimal value</i>	Optional

Keyword	Description	Choices	Notes
SVRTYPE	Server type	Single values: *NONE Other values (up to 5 repetitions): <i>Generic name, name</i>	Optional

Top

Type of data (TYPE)

Specifies the type of data to be printed.

This is a required parameter.

***DMP** The data to print was dumped by a previously issued Dump Job Internal (DMPJOBINT) command or by the machine when it was processing a device error or object damage. The dump identifier of the data or *ALL must be specified for the **Dump identifier (DMPID)** parameter.

***INTCFG**

The machine internal configuration and resource information is printed.

***NOTES**

The notes portion of the machine internal data, for the period specified for the **Time period for internal data (PERIOD)** parameter, is printed.

***JOB** The data to be printed is for the job specified for the JOB parameter.

Top

Dump identifier (DMPID)

Specifies, for internal dumps only, the dump identifiers associated with the machine internal data that is printed. This parameter must be specified **only** if *DMP is specified on the **Type of data (TYPE)** parameter; otherwise, it is ignored.

***NONE**

No dump identifier is specified.

***ALL** The dump portion of the machine internal data, for the period specified on the **Time period for internal data (PERIOD)** parameter, is printed.

character-value

Specify the dump identifier of the dump output that is printed. The identifier specified must contain 8 characters.

Top

Time period for internal data (PERIOD)

Specifies the period of time for which the notes or dump portion of the machine internal data is printed. This parameter is valid only if *NOTES is specified for the **Type of data (TYPE)** parameter or if *DMP is specified on the TYPE parameter and *ALL is specified for the **Dump identifier (DMPID)** parameter; otherwise, it is ignored.

Element 1: Start time and date

Element 1: Beginning time

***AVAIL**

The notes or dump data that are available from the beginning date to the ending date (or for the current day only) are printed.

- time* Specify the beginning time for the specified beginning date for which you want the notes or dump data printed. The time can be specified with or without a time separator:
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
 - With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Element 2: Beginning date

***CURRENT**

The notes or dump data that are available for the current day and between the specified beginning and ending times (if specified) are printed.

- date* Specify the beginning date for which you want the notes or dump data printed. The job date format must be used.

Element 2: End time and date

Element 1: Ending time

***AVAIL**

The notes or dump data available from the beginning date to the ending date (or for the current day only) are printed.

- time* Specify the ending time for the specified ending date for which you want the notes or dump data printed. See the **Beginning time** description on this parameter for details about how time must be specified.

Element 2: Ending date

***CURRENT**

The notes or dump data available for the current day and between the specified starting and ending times (if specified) are printed.

- date* Specify the ending date for which you want the notes or dump data printed. The system date format must be used.

Top

Job name (JOB)

Specifies the qualified name of the job for which the data will be dumped. This parameter must be specified **only** if *JOB is specified for the **Type of data (TYPE)** parameter; otherwise, it is ignored.

Single values

*
_ The job that issued this command is the job that will be dumped.

***SVRTYPE**

All jobs whose server type matches the server type attribute specified for the **Server type (SVRTYPE)** parameter will be dumped.

Qualifier 1: Job name

name Specify the name of the job to be dumped.

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 dumped.

Top

Thread ID to include (SLTTHD)

Specifies a list of up to twenty threads whose information is to be included. This parameter must be specified **only** if *JOB is specified for the **Type of data (TYPE)** parameter; otherwise, it is ignored.

Single values

***ALL** All threads are dumped.

***SELECT**

A list of thread identifiers is shown from which the user can select up to twenty to be included.

*SELECT is only valid if the PRTINTDTA command is run in an interactive session; otherwise, an error message is sent.

Other values

thread-identifier

Specify the identifiers of up to twenty threads whose information is to be included. A thread identifier is a string of eight hexadecimal characters.

Top

Server type (SVRTYPE)

Specifies the server type attribute to identify the job to be dumped. This parameter must be specified **only** if *SVRTYPE is specified for the **Job name (JOB)** parameter; otherwise, it is ignored. All jobs whose server types matches this value will be dumped. For a list of possible server types, see Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

Single values

***NONE**

No server types are dumped.

Other values

generic-name

Specifies the generic server type used to identify the job to be dumped. 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. All jobs whose server type matches the specified generic prefix will be dumped.

name Specify the server type used to identify the job to be dumped. A list of up to five server types can be specified.

Top

Examples

Example 1: Dump by Dump Identifier

```
PRTINTDTA TYPE(*DMP) DMPID(0102FA3C)
```

This command prints the job internal dump output that has a dump identifier of 0102FA3C.

Example 2: Dump by Job Identifier

```
PRTINTDTA TYPE(*JOB) JOB(201230/ALMATM/QPADEV0008)
          SLTTHD(*ALL)
```

This command prints the job internal dump output for the selected job including all threads information.

Example 3: Dump a Job by Specifying Server Type

```
PRTINTDTA TYPE(*JOB) JOB(*SVRTYPE) SVRTYPE(QIBM_FTP)
```

This command prints the dump output for the job with the server type set to QIBM_FTP.

Top

Error messages

*ESCAPE Messages

CPF3517

Cannot specify *SELECT for the thread ID to include.

CPF3519

Cannot start service function.

CPF6801

Command prompting ended when user pressed &1.

CPF98A2

Not authorized to &1

Top

Print IP over SNA (PRTIPSCFG)

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

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

The Print IP over SNA Configuration (PRTIPSCFG) command prints information about the current AF_INET Sockets over SNA configuration. The spooled file created by this CL command is named QSYSPRT. It is sent to the job default output queue. The user data value of the spooled file is PRTIPSCFG.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

PRTIPSCFG

This command prints the current AF_INET sockets over SNA configuration data.

[Top](#)

Error messages

*ESCAPE Messages

CPFA116

&1 configuration not printed.

[Top](#)

Print JOB Authority (PRTJOBDAUT)

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

Parameters
Examples
Error messages

The Print Job Description Authority (PRTJOBDAUT) command allows you to print a report of the job descriptions in a library that do not have public authority of *EXCLUDE, and a user name is specified in the job description. This is a way to check for job descriptions that every user on the system is authorized to use that allow the user to run as another user profile.

This command will print two reports for a library. The first report (Full Report) will contain all of the job descriptions that do not have public authority of *EXCLUDE and have a user name specified. The second report (Changed Report) will contain the job descriptions that **now** do not have public authority of *EXCLUDE or have a user name specified that either did have public authority of *EXCLUDE, did not have a user name specified, or did not exist when the PRTJOBDAUT command was previously run for the library. If the PRTJOBDAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional job descriptions do not have public authority of *EXCLUDE and a user name specified, then the 'Changed Report' will be printed but there will be no job descriptions listed. Changes to user profile special authorities will not cause a 'Changed Report' to be generated.

The reports will contain the following information:

- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each job description that does not have *PUBLIC authority of *EXCLUDE and has a user name specified. Each entry contains the following information:
 - The name of the library the job description is in.
 - The name of the job description.
 - The owner of the job description.
 - The name of the user profile specified in the job description.
 - The special authorities associated with the user profile. The special authorities that are shown are all of the special authorities that would be available when that job description is used. The special authorities shown are the special authorities that the user has, plus the special authorities that the user's group profiles have (if the user has any groups).

The file QSECJBDOLD in library QUSRSYS contains information from the last time the PRTJOBDAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the '*' will be replaced with a 'Q' in the member name. System file QAOBJAUT in library QSYS with format name of QSYDSAUT is the model file for the QSECJBDOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR, *ALLAVL, *ALLUSRAVL	Required, Positional 1
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2

Top

Library (LIB)

This is a required parameter.

The name of the library to search for job descriptions with public authority that is not *EXCLUDE and a user name is specified.

.

***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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

*ALLUSR

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

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

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

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

```
QDSNX      QRCLxxxxx   QUSRDIRDB   QUSRVI
QGPL       QSRVAGT    QUSR IJS    QUSRVRxMx
QGPL38     QSYS2      QUSRINFSKR
QMGTC      QSYS2xxxxx QUSRNOTES
QMGTC2     QS36F      QUSROND
QMPGDATA   QUSER38    QUSRPOSGS
```

QMOMDATA	QUSRADSM	QUSRPOSSA
QMOMPROC	QUSRBRM	QUSRPYMSVR
QPFRDATA	QUSRDIRCF	QUSRDRARS
QRCL	QUSRDIRCL	QUSRSYS

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALLAVL**

All libraries in all available ASPs are searched.

***ALLUSRAVL**

All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name Specify the name of the library to be searched.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

***NO** The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Top

Examples

```
PRTJOBDAUT LIB(QGPL)
```

This command prints both full and changed report for the job descriptions in the library QGPL.

Top

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

Top

Print JVM Job (PRTJVMJOB)

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

Parameters
Examples
Error messages

The Print JVM Job (PRTJVMJOB) command allows the user to print Java Virtual Machines (JVMs) running in active jobs.

Restrictions:

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 worked with, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority.

The job user identity is the name of the user profile by which a job is known to other jobs. More information about the job user identity is in the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter>.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	<i>Qualified job name</i>	Required, Positional 1
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
OPTION	Option	Single values: *ALL Other values (up to 9 repetitions): *CRTOPT, *ENVVAR, *PASEENVVAR, *LCK, *GC, *THREAD, *INLPROP, *CURPROP, *GCTABLE	Optional

Top

Job name (JOB)

Specifies the name of the job where the JVM is running. All three of the qualifiers for the job parameter are required.

This is a required parameter.

Qualifier 1: Job

name Specify the name of the job to be printed.

Qualifier 2: User

name Specify the name that identifies the user profile under which the job was started.

Qualifier 3: Number

000000-999999

Specify the job number assigned by the system.

Top

Option (OPTION)

Specifies which information is being worked with.

Single values

***ALL** Print all of the JVM information.

Other values (up to 9 repetitions)

***CRTOPT**

Print JVM creation options

***ENVVAR**

Print environment variables

***PASEENVVAR**

Print PASE environment variables

***LCK** Print Java lock information

***GC** Print garbage collection information

***THREAD**

Print threads

***INLPROP**

Print initial Java system properties.

***CURPROP**

Print current Java system properties.

***GCTABLE**

Print the GC table information

Top

Examples

```
PRTJVMJOB JOB(032072/USERNAME/QP0ZSPWP) OPTION(*THREAD)
```

This command will print the Java thread information for the job 032072/USERNAME/QP0ZSPWP.

Top

Error messages

*ESCAPE Messages

CPE3440

Operation not supported.

CPF9871

Error occurred while processing.

424 System i: Programming i5/OS commands Starting with INZPCS (Initialize Client Access/400)

JVAB321

Java Virtual Machine job &3/&2/&1 not found or not active.

JVAB322

Not authorized to perform function on Java Virtual Machine.

JVAB323

Operation not supported with a Classic Java Virtual Machine.

JVAB328

Machine or ASP storage limit reached.

JVAB333

PRTJVMJOB failed.

[Top](#)

Print PEX Report (PRTPEXRPT)

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

Parameters
 Examples
 Error messages

The Print PEX Report (PRTPEXRPT) command prints a formatted listing of the data that was collected by the performance explorer and saved across a set of physical files in a particular library.

Restrictions:

- This command is shipped with PUBLIC *EXCLUDE authority.
- You must have read (*READ) and execute (*EXECUTE) authority to the specified library.
- To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of the operating system through i5/OS Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) CL command or 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.

Top

Parameters

Keyword	Description	Choices	Notes
MBR	Member	<i>Name</i>	Required, Positional 1
LIB	Library	<i>Name</i> , <u>QPEXDATA</u>	Optional, Positional 2
TYPE	Type	<u>*STATS</u> , *TRACE, *PROFILE, *BASIC	Optional, Positional 3
OUTPUT	Output	<u>*PRINT</u> , *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
TRACEOPT	Trace options	<i>Element list</i>	Optional
	Element 1: Sort by	<u>*TIMESTAMP</u> , *TASK	
	Element 2: Omit completion records	<u>*NO</u> , *YES	
	Element 3: Omit Category	Single values: <u>*NONE</u> Other values (up to 17 repetitions): *PGM, *LICPGM, *ASM, *BASE, *DISK, *DSKSVR, *FAULT, *JOB, *LOCK, *SAR, *MIBRKT, *LICBRKT, *DASD, *DASDSRVR, *PAGEFLT, *RMPR, *RMSL	

Keyword	Description	Choices	Notes
TRCTYPE	Select trace type	Single values: *ALL Other values (up to 11 repetitions): *CALLRTN, *BASIC, *DSKIO1, *DSKIO2, *DSKSVR, *DSKSTG, *VRTADR, *PGMACT, *FILEOPEN, *PRFDTA, *TASKSWT	Optional
PERIOD	Time period for report	<i>Element list</i>	Optional
	Element 1: Start time and date	<i>Element list</i>	
	Element 1: Starting time	<i>Time, *AVAIL</i>	
	Element 2: Starting date	<i>Date, *BEGIN, *CURRENT</i>	
	Element 2: End time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time, *AVAIL</i>	
	Element 2: Ending date	<i>Date, *END, *CURRENT</i>	
SLTJOB	Select jobs	Single values: *ALL Other values (up to 10 repetitions): <i>Qualified job name</i>	Optional
	Qualifier 1: Select jobs	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</i>	
	Qualifier 3: Number	000001-999999, *ALL	
OMTJOB	Omit jobs	Single values: *NONE Other values (up to 10 repetitions): <i>Qualified job name</i>	Optional
	Qualifier 1: Omit jobs	<i>Generic name, name</i>	
	Qualifier 2: User	<i>Generic name, name, *ALL</i>	
	Qualifier 3: Number	000001-999999, *ALL	
STATSOPT	Stats options	<i>Element list</i>	Optional
	Element 1: Sort by	<i>Integer, *CPU, *PGMNAME, *INVCNT, *DBSYNPIO, *DBASYNPIO, *NDBSYNPIO, *NDBASYNPIO, *MICALLS, *MIINST, *CUMLCPU, *CUMLDBSYNPIO, *CUMLDBASYNPIO, *CUMLNDBSYNPIO, *CUMLNDBASYNPIO</i>	
	Element 2: Summarize by	*PROGRAM , *MODULE, *BLANK	
PROFILEOPT	Profile options	<i>Element list</i>	Optional
	Element 1: Sort by	*SAMPLECOUNT , *ADDRESS	
	Element 2: Summarize by	*PROGRAM , *MODULE, *PROCEDURE, *STATEMENT, *BLANK	
	Element 3: Filter percentage	0-99, 0	
ORDER	Order	*DESCENDING , *ASCENDING	Optional
NBRTHD	Number of threads	1-64, 1 , *CALC	Optional
TASKINF	Task information	*ALL , *NONE	Optional

Top

Member (MBR)

Specifies where the data is located for the report. This is the value that was specified for the SSNID or DTAMBR parameter when the data was saved using the End Performance Explorer (ENDPEX) command. Each database file used by performance explorer when it saved the collected performance data should have a member with the name specified.

This is a required parameter.

name Specify the member name.

Library (LIB)

Specifies the library where the data will be found.

QPEXDATA

The collected data exists in database files in library QPEXDATA.

name Specify the library name which contains the database files that hold the collected performance data.

Type (TYPE)

Specifies the type of report to produce. The type of report requested must match the type of data that was collected. If there is a mismatch, an error message is issued. The type of performance data collected is determined by the performance explorer definition that was specified on the Start Performance Explorer (STRPEX) command. Refer to the Add PEX Definition (ADDPEXDFN) command for more information.

Note: An exception to the matching of types occurs when you collect data with a definition of TYPE(*TRACE) INTERVAL(nn) BASEVT(*PMCO). When you collect this trace data, you are allowed to specify a report type of *PROFILE. This type of report is known as a *TRACE collection and a *PROFILE report.

*STATS

A statistics report is produced.

Note: This parameter is valid only for data collected by *STATS mode definitions.

*TRACE

A trace report is produced.

Note: This parameter is valid only for data collected by *TRACE mode or *PROFILE PRFTYPE(*JOB) definitions.

*PROFILE

A profile report is produced.

Note: This parameter is valid only for data collected by *PROFILE mode definitions or *TRACE TRCTYPE(*PROFILE) definitions.

*BASIC

A basic report is produced that includes the definition, run, and task information sections.

Note: This parameter is valid for data collected by any definition.

Output (OUTPUT)

Specifies whether the output from the command is printed with the job's spooled output or directed to a database file.

Note: This parameter is valid only if TYPE(*TRACE) is specified.

*PRINT

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

*OUTFILE

The output is directed to the database file specified in the **File to receive output (OUTFILE)** parameter.

Top

File to receive output (OUTFILE)

Specifies the name of the database file to which the output of the command is directed. If this file does not exist, this command creates a database file in the specified library. The public authority is the same as the create authority specified for the library in which the file is created.

Notes:

1. The file specified here cannot be a DDM file
2. The model file QAVPETRCI resides in library QSYS.

Qualifier 1: File to receive output

name Specify the name of the output file that receives the output of the command.

Qualifier 2: Library

*LIBL The library list is used to locate the output file. If the output file is not found, one is created in the current library. If no current library exists, the output file is created in the QGPL library.

*CURLIB

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

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

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed.

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 on the **File to receive output (OUTFILE)** parameter. If the member already exists, the user has 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 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 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.

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

Trace options (TRACEOPT)

Specifies how to organize a trace (*TRACE) report. Records are ordered based on the value specified for the **Order (ORDER)** parameter.

Element 1: Sort by

*TIMESTAMP

The records are listed in time stamp order.

*TASK

The records are listed in time stamp order within each job/task.

Element 2: Omit completion records

*NO All records associated with this performance data collection session are included in the report.

*YES All completion records are excluded from the report. This is helpful if there is a large amount of data to review.

Element 3: Omit Category

Single values

*NONE

No categories are omitted.

Other values (up to 17 repetitions)

*PGM Exclude the category for the program call flow events.

*LICPGM

Exclude the category for the Licensed Internal Code call flow events.

*ASM Exclude the category for the auxiliary storage management events.

*BASE

Exclude the category for the base events, which includes tasking events.

*DISK

Exclude the category for the direct access storage device events.

*DSKSVR

Exclude the category for the disk server events.

*FAULT

Exclude the category for the page fault events.

*JOB Exclude the category for the job or process management events.

*LOCK

Exclude the category for the seize lock events.

*SAR Exclude the category for the segment address register events.

*MIBRKT

Exclude the category for the machine interface program bracketing events.

*LICBRKT

Exclude the category for the Licensed Internal Code bracketing events.

***DASD**

Exclude the category for the direct access storage device events.

***DASDSRVR**

Exclude the category for the disk server events.

***PAGEFLT**

Exclude the category for the page fault events.

***RMPR**

Exclude the category for the resource management process management events.

***RMSL**

Exclude the category for the resource management seize lock events.

Top

Select trace type (TRCTYPE)

Specifies which trace events to include in the output. The options possible are the same options found on the Add PEX Definition (ADDPEXDFN) command.

Single values

***ALL** Include all trace events in the output.

Other values (up to 11 repetitions)

***CALLRTN**

Specifies that call return events are included in the output. Call return events occur when a program is entered and exited as well as when certain machine instructions are started and completed.

***BASIC**

Specifies that events relative to general performance analysis are included in the output.

***DSKIO1**

Specifies that events associated with disk input/output operations are included in the output.

***DSKIO2**

Specifies that events associated with the disk input/output operations plus higher level requests to do input/output operations are included in the output.

***DSKSVR**

Specifies that events associated with disk server operations are included in the output.

***DSKSTG**

Specifies that events associated with disk storage consumption are included in the output.

***VRTADR**

Specifies that events associated with virtual address assignment are included in the output.

***PGMACT**

Specifies that events associated with program activations and deactivations are included in the output.

***FILEOPEN**

Specifies that events associated with file opens are included in the output.

***PRFDTA**

Specifies that events associated with CPU instruction profiling are included in the output.

Note: The *PFRDTA value provides you with a detailed list of files. To receive a list in a summary format, as an alternative, you can specify PRTPEXRPT TYPE(*PROFILE).

***TASKSWT**

Specifies that events associated with tasking are included in the output.

Top

Time period for report (PERIOD)

Specifies the period of time on which to report. The parameter consists of two lists of two elements each. Data collected prior to the starting time on the starting date and after the ending time on the ending date is not included in the report.

Element 1: Start time and date

Element 1: Starting time

***AVAIL**

The recorded data that is available for the specified starting date is shown.

- time* Specify the starting time on the specified starting date that indicates the recorded data to be shown. The time is specified in 24-hour format with or without a time separator:
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
 - With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

All time and date entries must be 2-digits in length, meaning zeros must be included.

Element 2: Starting date

***BEGIN**

The recorded data from the beginning of the log is shown.

***CURRENT**

The recorded data for the current day and between the specified starting and ending times (if specified) is shown.

- date* Specify the date printed. The date must be entered in the format specified by the system value QDATFMT, and if separators are used, as specified by the system value QDATSEP.

Element 2: End time and date

Element 1: Ending time

***AVAIL**

The recorded data that is available for the specified ending date is shown.

- time* Specify the ending time for the specified ending date that determines the recorded date that is printed.

Element 2: Ending date

***END** The last day on which data was logged is shown. If PERIOD(*END) is specified, a time value other than *AVAIL for end time is ignored.

*CURRENT

The current day is the last day for which recorded data is shown.

date Specify the ending date for which recorded data is to be printed. The date must be entered in the format specified by the system value QDATFMT, and if separators are used, as specified by the system value QDATSEP.

Top

Select jobs (SLTJOB)

Specifies which jobs to include from the report. This allows the user to narrow the scope of the performance explorer report by selecting specific jobs.

The SLTJOB and OMTJOB parameters are mutually exclusive.

Single values

***ALL** All jobs in the performance explorer database are included.

Other values (up to 10 repetitions)

Qualifier 1: Select jobs

name Specify the name of the job to be included in the performance explorer report.

generic-name

Specify the generic name of the job to be included in the performance explorer report.

Note: A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic name, the system assumes it to be the complete object name.

Qualifier 2: User

***ALL** All jobs that match the specified job name are included.

name Specify the name of the user of the job to be included.

generic-name

Specify the generic user name of the jobs to be included.

Qualifier 3: Number

***ALL** All jobs that match the specified job name and user name are included.

000001-999999

Specify the job number to further qualify the job name and user name.

Top

Omit jobs (OMTJOB)

Specifies which jobs are omitted from the report. This allows the user to narrow the scope of the performance explorer report by omitting specific jobs.

The SLTJOB and OMTJOB parameters are mutually exclusive. You must use the default for one of these parameters.

Single values

*NONE

No jobs in the performance explorer database are omitted.

Other values (up to 10 repetitions)

Qualifier 1: Omit jobs

name Specify the name of the job to be omitted in the performance explorer report.

generic-name

Specify the generic name of the job to be omitted in the performance explorer report.

Note: A generic name is a character string that contains one or more characters followed by an asterisk(*), for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic name, the system assumes it to be the complete object name.

Qualifier 2: User

*ALL All jobs that match the specified job name will be omitted.

name Specify the name of the user of the job to be omitted.

generic-name

Specify the generic user name of the jobs to be omitted.

Qualifier 3: Number

*ALL All jobs that match the specified job name and user name will be omitted.

000001-999999

Specify the job number to further qualify the job name and user name.

Top

Stats options (STATSOPT)

Specifies how to organize a statistics (*STATS) report. Records are ordered based on the value specified for the **Order (ORDER)** parameter.

Note: This parameter is ignored if, on the Add PEX Definition (ADDPEXDFN) command, you specified TYPE(*STATS) and DTAORG(*HIER). The parameter is ignored to retain the parent-child relationship that was collected for this definition.

Element 1: Sort by

*CPU Arrange the output by amount of CPU time.

*PGMNAME

Arrange the output by program name.

***INVCNT**

Arrange the output by number of times program or procedure is called.

***DBSYNCIO**

Arrange the output by amount of physical database synchronous I/O.

***DBASYNCIO**

Arrange the output by amount of physical database asynchronous I/O.

***NDBSYNCIO**

Arrange the output by amount of physical non-database synchronous I/O.

***NDBASYNCIO**

Arrange the output by amount of physical non-database asynchronous I/O.

***MICALLS**

Arrange the output by number of MI calls.

***MIINST**

Arrange the output by MI instruction name.

***CUMLCPU**

Arrange the output by cumulative CPU value.

***CUMLDBSYNCIO**

Arrange the output by cumulative amount of physical database synchronous I/O.

***CUMLDBASYNCIO**

Arrange the output by cumulative amount of physical database asynchronous I/O.

***CUMLNDBSYNCIO**

Arrange the output by cumulative amount of physical non-database synchronous I/O.

***CUMLNDBASYNCIO**

Arrange the output by cumulative amount of physical non-database asynchronous I/O.

Element 2: Summarize by

***PROGRAM**

The data is summarized at the program level.

***MODULE**

The data is summarized at the module level.

***BLANK**

The data is not summarized.

Top

Profile options (PROFILEOPT)

Specifies how to organize a profile (*PROFILE) report. Records are ordered based on the value specified for the **Order (ORDER)** parameter.

Element 1: Sort by

***SAMPLECOUNT**

Arrange the output relative to the sample count.

***ADDRESS**

Arrange the output relative to the sampled address.

Element 2: Summarize by

***PROGRAM**

Summarize the data at the program level.

***MODULE**

Summarize the data at the module level.

***PROCEDURE**

Summarize the data at the procedure level.

***STATEMENT**

Summarize the data at the statement level.

***BLANK**

No summary records are provided.

Element 3: Filter percentage

0 No records are omitted from the report.

0-99 Specify a number in the range of 0 to 100.

Note: This provides a filter to eliminate the insignificant records. For example, an entry of 10 would omit all the records that contain less than 10% of the samples taken during the collection.

Top

Order (ORDER)

Specifies how the data should be ordered in the report.

***DESCENDING**

The data records are ordered in descending order. If records are sorted by a numeric field, records are ordered from largest to smallest. If records are sorted by a name field, records are in reverse alphabetical order, for example, from Z to A.

***ASCENDING**

The data records are in ascending order. If records are sorted by a numeric field, records will be ordered from smallest to largest. If records are sorted by a name field, records are in alphabetical order, for example, from A to Z).

Top

Number of threads (NBRTHD)

Specifies the number of concurrent threads that the PRTPEXRPT command uses to print the data. Specifying a number greater than 1 allows the PRTPEXRPT command to take advantage of available CPU cycles, especially on a multi-processor system. While this may speed up the command processing, it may also degrade the performance of other jobs on the system. You can minimize this impact by changing the priority of the job that runs the PRTPEXRPT command to a higher number. You should also verify that the disk subsystem can handle the additional threads. Typically, the PRTPEXRPT command requires one disk arm for each active thread.

Note: If you specify OUTPUT (*PRINT), the number of spooled files is equal to NBRTHD (one spooled file per thread).

1 Only one thread for the PRTPEXRPT command is used to process the collected data.

***CALC**

The system calculates a reasonable number of threads to do the command processing which does

not use excessive system resources. Usually this is one or two threads for each available processor. If this command is run in an interactive job, *CALC uses only one thread.

1-64 Specify the number of threads for the PRTPEXRPT command to use to process the collected data.

Top

Task information (TASKINF)

Specifies whether the task information section is to be printed or not.

***ALL** The task information section will be printed and will include details for all jobs and tasks available in the collected data. If ADDPEXDFN LSTALLJOB(*YES) was specified in the definition used to collect this data, then details for all jobs and tasks on the system during the time of the collection will be printed.

***NONE**
The task information section will not be printed.

Top

Examples

Example 1: Printing a Statistics Report

```
PRTPEXRPT  MBR(SAMPLE)  LIBRARY(SAMPLELIB)
           TYPE(*STATS)  STATSOPT(*INVCNT *MODULE)
```

This command prints a report based on data members named SAMPLE in library SAMPLELIB. The data is arranged in descending order based on invocation counts and is summarized at the module level.

Example 2: Printing a Profile Report

```
PRTPEXRPT  MBR(SAMPLE2)  TYPE(*PROFILE)
           PROFILEOPT(*SAMPLECOUNT *PROGRAM)
           ORDER(*DESCENDING)
```

This command prints a report based on data members named SAMPLE2 in the default library, QPEXDATA. The data is arranged in descending order based on the sample count and is summarized at the program level.

Top

Error messages

*ESCAPE Messages

CPFAF0A
Collection &1 was not found

CPFAF09
Library &1 not found.

CPF4102
File &2 in library &3 with member &4 not found.

CPFAF14

The data in the member specified does not match the type of report requested.

[Top](#)

Print Profile Internals (PRTPRFINT)

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

Parameters
Examples
Error messages

The Print Profile Internals (PRTPRFINT) command allows you to print a report containing information on the number of entries contained in a user profile (*USRPRF) object. The number of entries in the user profile determines the size of the user profile.

Four types of entries can be found in a user profile:

Owned objects

For every object a user profile owns, an "owned objects" entry exists in that user's profile (*USRPRF).

Private authorities

For every private authority a user is granted, a "private authority" entry exists in that user's profile (*USRPRF).

Authorized objects

For every user that is granted a private authority to an object a profile owns, an "authorized object" entry exists in the object owner's profile (*USRPRF).

Primary group authorities

For every object for which a user is the primary group, a "primary group" entry exists in that user's profile (*USRPRF).

Each entry in the user profile causes the user profile (*USRPRF) object to grow. The combination of all of the entries determines the size of the user profile. A user profile (*USRPRF) can contain approximately 10 million entries for objects in the system auxiliary storage pool plus an additional 10 million entries for each independent auxiliary storage pool which is varied on that the profile owns objects on, has private authority to objects on, or is the primary group profile of objects on it. A user profile can not exceed 10 million entries for the objects on any single auxiliary storage pool.

You can also think of the total number of entries as determining how "full" a user profile is. The report produced by this command shows how full the user profile is by giving a percentage (rather than giving the actual number of entries in the profile.) The report also gives a percentage for each of the four types of entries in the *USRPRF. Note: In the report, the total percentage full for a profile can be greater than 100% due to rounding.

You can choose to run this report for all user profiles, a subset of profiles, a specific profile or all profiles that are at least a specified percentage full. For example, you can run the report for the CJW profile or you can run the report for all profiles that are at least 99.90% full.

Note: If your system has any independent auxiliary storage pools varied on, then the percentages produced by this report may not be what you expect. The percentage is computed for each profile based on the total number of entries used divided by the total number of entries available to that specific profile. The total number of entries available to each profile may not be the same depending on whether the profile has entries for any object on a varied on independent auxiliary storage pool. For example, if the system has two varied on independent auxiliary storage pools and profile TESTUSER1 only has entries for objects on one of them, then the total number of entries available for TESTUSER1 is 20 million. If profile TESTUSER2 has entries for objects on both of the independent auxiliary pools, then the total

number of entries available for TESTUSER2 is 30 million. However, if both of the independent auxiliary storage pools are varied off, then the total number of entries available for profiles TESTUSER1 and TESTUSER2 is 10 million.

Recommendations to avoid profiles becoming full:

- Do not have one profile own everything on your system. For example, have each application be owned by its own profile.
- Do not use IBM-supplied profiles, such as QSECOFR and QPGMR, as owners of your application. As shipped from IBM, they already own many objects and can become full when they also own user (non-IBM) objects.
- If you are granting private authorities to many objects for several users, you should consider using an authorization list to secure the objects. Authorization lists will cause one private authority entry for the authorization list in the user's profile rather than one private authority entry for each object. In the object owner's profile, authorization lists will cause an authorized object entry for every user granted authority to the authorization list rather than an authorized object entry for every object multiplied by the number of users granted the private authority.

Authorization lists are especially useful if you are granting private authorities to files. Files are complex objects. For complex objects, you get an entry for each piece of the object. For example, in a file owner's profile, you have an ownership entry for each piece of the file, including an entry or two for each member. (Physical files have two entries per member.) If you grant a private authority to ten users and the file has 50 members, the result will be 100 authorized object entries in the file owner's profile. With an authorization list, the ownership entries will remain the same, but the authorized object entries will be reduced to one for each user granted authority to the authorization list securing the file.

Do not confuse the percentage full of a user profile with the maximum storage (MAXSTG) that a user profile can own. They are two different concepts.

Restriction: You must have all object (*ALLOBJ) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
SELECT	Select by	*USRPRF, *PCTFULL	Optional
USRPRF	User profile	<i>Qualifier list</i>	Optional
	Qualifier 1: User profile	<i>Generic name, name, *ALL</i>	
PCTFULL	Percent full	0.01-100.0, <u>99.90</u>	Optional

Top

Select by (SELECT)

Specifies what criteria is used to select the user profiles to include in the report.

*USRPRF

User profiles are selected for the report based on the profile name specified for the USRPRF parameter.

*PCTFULL

User profiles are selected for the report based on the value specified for the PCTFULL parameter.

User profile (USRPRF)

If *USRPRF was specified for the **Select by (SELECT)** parameter, you must specify the user profiles to be included in the report.

***ALL** All user profiles will be included in the report.

user-name

The name of the user profile to be included in the report.

generic-user-name

The generic name of the user profile to be included in the report. A generic name is a character string of one or more characters followed by an asterisk (*).

Percent full (PCTFULL)

If *PCTFULL was specified for the **Select by (SELECT)** parameter, you must enter a value which will be used as the percentage full. User profiles that are at least as full as the percentage specified on this parameter will be included in the report. The value specified must be between 0.01 and 100.00.

99.90 User profiles that are at least 99.9 percent filled with entries will be included in the report.

percent-full

A value, ranging from 0.01 through 100.00, for the percent full selection value.

Examples

```
PRTPRFINT  SELECT(*PCTFULL)  PCTFULL(99.00)
```

This command prints a report of user profile internal information for all of the user profiles that are at least 99 percent full.

Error messages

***ESCAPE Messages**

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

Print Publicly Auth Objects (PRTPUBAUT)

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

Parameters
Examples
Error messages

The Print Publicly Authorized Objects (PRTPUBAUT) command allows you to print a report of the specified objects that do not have public authority of *EXCLUDE. For *PGM objects, only the programs that do not have public authority of *EXCLUDE that a user can call (the program is either user domain or the system security level (QSECURITY system value) is 30 or below) will be included in the report. This is a way to check for objects that every user on the system is authorized to access.

This command will print two reports. The first report (Full Report) will contain all of the specified objects that do not have public authority of *EXCLUDE. The second report (Changed Report) will contain the objects that **now** do not have public authority of *EXCLUDE that did have public authority of *EXCLUDE or did not exist when the PRTPUBAUT command was previously run. If the PRTPUBAUT command was not previously run for the specified objects and library or folder, there will be no 'Changed Report'. If the command has been previously run, but no additional objects do not have public authority of *EXCLUDE, then the 'Changed Report' will be printed but there will be no objects listed.

The reports will contain the following information:

- The object type specified on the command (if object type is not *DOC or *FLR).
- The name of the library specified on the command (if object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
- The name of the folder the documents are in (if object type is *DOC).
- The name of the directory objects are in (if object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK).
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each object that does not have *PUBLIC authority of *EXCLUDE. Each entry contains the following information:
 - The name of the library the object is in (if object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
 - The name of the folder the folder is in (if object type is *FLR).
 - The name of the object.
 - The owner of the object.
 - The authorization list securing the object.
 - The special value for the *PUBLIC authority (e.g. *ALL or *CHANGE).
 - The sensitivity level of the document or folder (if object type is *DOC or *FLR).
 - An indicator for the individual authorities that *PUBLIC has to the program ('X' or ' ') (if object type is not *DOC or *FLR).

The file QPBXXXXXXX (where 'XXXXXXX' is the object type specified on the command) in library QUSRSYS contains information from the last time the PRTPUBAUT command was run. If object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK there is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the '*' will be replaced with a 'Q' in the member name. For object types that don't require a library to be specified (e.g. *USRPRF), the library name is QSYS. System file QAOBJAUT in library QSYS with format name of QSYDSAUT is the model file for the file.

If the object type is *FLR, the first member will contain the information from the previous time *FLR was specified on the command. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *DOC, there is a member within the file for each folder that has been previously specified on the command. The member name will be the same as the system name of the folder. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK, there is a member within the file for each directory that has previously been specified in the **Directory (DIR)** parameter. The member names are based on the order the directories are processed. The member naming convention is x000000001, x000000002, and so on. The first character in the member name will either be N or Y. This character indicates if the subdirectories were searched when the data was gathered. N indicates the subdirectories were not searched, Y indicates they were searched. Once a member name has been assigned to a directory, the numeric portion with the appropriate prefix is used for all of the object types listed above. The system file QASECDIR in library QSYS with format name of QSECDIR is the model file for the file.

Note: The file QASECGFIPB in library QUSRSYS contains the file ID values of every directory that has been processed and the Nxxxxxxx member name that has been assigned to it. The system file QASECGFI in library QSYS with format name of QSECGFI is the model file for QASECGFIPB.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJTYPE	Object type	*ALRTBL, *AUTL, *BLKSF, *BNDDIR, *CFGL, *CHRSE, *CHTFMT, *CLD, *CLS, *CMD, *CNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVLD, *DIR, *DOC, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FIFO, *FILE, *FLR, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDL0D, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SOCKET, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *STMF, *SVRSTG, *SYMLNK, *S36, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 1
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2
LIB	Library	Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR	Optional
FILAUT	Print file authority	*NO, *YES	Optional
CMDAUT	Print command authority	*NO, *YES	Optional
PGMAUT	Print program authority	*NO, *YES	Optional
JOBDAUT	Print JOBQ authority	*NO, *YES	Optional
FLR	Folder	Character value	Optional
DIR	Directory	Path name	Optional
SCHSUBDIR	Search subdirectory	*NO, *YES	Optional

Object type (OBJTYPE)

This is a required parameter.

The type of object to search for. For a complete list of object types, press the F4 key when prompting this parameter.

object-type

The type of object to be processed.

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

***NO** The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Library (LIB)

This is a required parameter for all object types except *AUTL, *BLKSF, *CFGL, *CNNL, *COSD, *CTLD, *DEVD, *DIR, *DOC, *FLR, *LIB, *LIND, *MODD, *NWID, *NWSD, *SOCKET, *STMF, *SYMLNK, and *USRPRF.

The name of the library to search for objects with public authority that is not *EXCLUDE.

***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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

***ALLUSR**

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

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

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

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

QDSNX	QRCLxxxxx	QUSRDIRB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

name Specify the name of the library to be searched.

Top

Print file authority (FILAUT)

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *FILE objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified for the **Object type (OBJTYPE)** parameter.

Note: This parameter is only used when OBJTYPE is *LIB.

***NO** The PRTPUBAUT command will not be run for *FILE objects for each of the libraries that does not have public authority of *EXCLUDE.

***YES** The PRTPUBAUT command will be run for *FILE objects for each of the libraries that does not have public authority of *EXCLUDE.

Top

Print command authority (CMDAUT)

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *CMD objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified for the **Object type (OBJTYPE)** parameter.

Note: This parameter is only used when OBJTYPE is *LIB.

***NO** The PRTPUBAUT command will not be run for *CMD objects for each of the libraries that does not have public authority of *EXCLUDE.

***YES** The PRTPUBAUT command will be run for *CMD objects for each of the libraries that does not have public authority of *EXCLUDE.

Top

Print program authority (PGMAUT)

Specifies whether the Print Publicly Authorized Objects (PRTPUBAUT) command will be run for *PGM objects for each of the libraries that do not have public authority of *EXCLUDE, when *LIB is specified for the **Object type (OBJTYPE)** parameter.

Note: This parameter is only used when OBJTYPE is *LIB.

- *NO** The PRTPUBAUT command will not be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.
- *YES** The PRTPUBAUT command will be run for *PGM objects for each of the libraries that does not have public authority of *EXCLUDE.

Top

Print JOB Description Authority (JOBDAUT)

Specifies whether the Print Job Description Authority (PRTJOBDAUT) command will be run for each of the libraries that does not have public authority of *EXCLUDE, when *LIB is specified for the **Object type (OBJTYPE)** parameter. The PRTJOBDAUT command will list all of the job descriptions in the library that do not have public authority of *EXCLUDE and have a user name specified.

Note: This parameter is only used when OBJTYPE is *LIB.

- *NO** The PRTJOBDAUT command will not be run for each of the libraries that does not have public authority of *EXCLUDE.
- *YES** The PRTJOBDAUT command will be run for each of the libraries that does not have public authority of *EXCLUDE.

Top

Folder (FLR)

This is a required parameter if *DOC is specified for the **Object type (OBJTYPE)** parameter.

The name of the folder to search for documents with *PUBLIC authority that is not *EXCLUDE.

folder-name

The name of the folder to be searched.

Top

Directory (DIR)

This is a required parameter if *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK is specified for the **Object type (OBJTYPE)** parameter.

The pathname of the directory to search for objects that do not have public authority of *EXCLUDE. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

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

directory-name

The name of the directory to be searched.

Top

Search subdirectory (SCHSUBDIR)

Specifies whether to search the subdirectories for objects to be included in the public authority report.

Note: This parameter is only used when OBJTYPE is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK.

***NO** The subdirectories are not searched.

***YES** The subdirectories are searched.

Top

Examples

```
PRT PUBAUT OBJTYPE(*FILE) LIB(QSYS)
```

This command prints both full and changed reports for the file objects in the library QSYS.

Top

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

Top

Print Private Authorities (PRTPVTAUT)

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

Parameters
Examples
Error messages

The Print Private Authority (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder or directory. The report will list all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects.

This command will print three reports for the selected objects. The first report (Full Report) will contain all of the private authorities for each of the selected objects.

The second report (Changed Report) will contain additions/changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library or folder. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects will be listed in the 'Changed Report'. If the PRTPVTAUT command was not previously run for the specified objects in the specified library or folder, there will be no 'Changed Report'. If the command has been previously run but no changes have been made to the authorities on the objects, then the 'Changed Report' will be printed but there will be no objects listed.

The third report (Deleted Report) will contain any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users that were removed as privately authorized users will be listed in the 'Deleted Report'. If the PRTPVTAUT command was not previously run, there will be no 'Deleted Report'. If the command has been previously run but no delete operations have been done to the objects, then the 'Deleted Report' will be printed but there will be no objects listed.

The reports will contain the following information:

- The object type specified on the command (if object type is not *AUTL).
- The date and time the report was last run (not shown on the Full Report).
- The name of the library specified on the command (if object type is not *AUTL, *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
- The library's *PUBLIC authority (if object type is not *AUTL, *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK).
- The name of the folder the documents or folder are in (if object type is *DOC or *FLR).
- The name of the directory the objects are in (if object type is *BLKSF, *DIR, *SOCKET, *STMF, *SYMLNK).
- The directory's *PUBLIC authority (if object type is *BLKSF, *DIR, *SOCKET, *STMF, *SYMLNK).
- An entry for each user that has an authority to the objects in the list. Each entry contains the following information:
 - The name of the object (only shown for the first user).
 - The owner of the object (only shown for the first user).
 - The primary group of the object (only shown for the first user).
 - The name of the authorization list securing the object (only shown for the first user if object type is not *AUTL).
 - The sensitivity level of the document or folder (if object type is *DOC or *FLR, only shown for the first user).
 - The name of the user authorized to the object.

- The special value for the user's authority to the object (e.g. *ALL or *CHANGE).
- An indicator for the individual authorities that the user has to the object ('X' or ' ') (if object type is not *DOC or *FLR).

The file QPVXXXXXXXX (where 'XXXXXXXX' is the object type specified on the command) in library QUSRSYS contains information from the last time the PRTPVTAUT command was run. If object type is not *BLKSF, *DIR, *DOC, *FLR, *SOCKET, *STMF, or *SYMLNK there is a member within the file, with the same name as the library, for each library that has been previously specified on the command. For object types that don't require a library to be specified (e.g. *USRPRF), the library name is QSYS. System file QAOBJAUT in library QSYS with format name of QSYDSAUT is the model file for the file.

If the object type is *FLR, the first member will contain the information from the previous time *FLR was specified on the command. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *DOC, there is a member within the file for each folder that has been previously specified on the command. The member name will be the same as the system name of the folder. System file QASECDLO in library QSYS with format name of QSECDLO is the model file for the file.

If the object type is *FILE and the AUTTYPE parameter value is *FIELD or *ALL, the Display Object Authority (DSPOBJAUT) command will be run for each file that has field level authorities associated with it. For each of these files, a spooled file by the name of QPOBJAUT will be created that contains all of the field level authority data for the file. There is no changed report support available for the field level authority data on a file.

If the object type is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK, there is a member within the file for each directory that has previously been specified in the **Directory (DIR)** parameter. The member names are based on the order the directories are processed. The member naming convention is x000000001, x000000002, and so on. The first character in the member name will either be N or Y. This character indicates if the subdirectories were searched when the data was gathered. N indicates the subdirectories were not searched, Y indicates they were searched. Once a member name has been assigned to a directory, the numeric portion with the appropriate prefix is used for all of the object types listed above. The system file QASECDIR in library QSYS with format name of QSECDIR is the model file for the file.

Note: The file QASECGFIPV in library QUSRSYS contains the file ID values of every directory that has been processed and the Nxxxxxxx member name that has been assigned to it. The system file QASECGFI in library QSYS with format name of QSECGFI is the model file for QASECGFIPV.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJTYPE	Object type	*ALRTBL, *AUTL, *BLKSF, *BNDDIR, *CFGL, *CHRSE, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DIR, *DOC, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FIFO, *FILE, *FLR, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SOCKET, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *STMF, *SVRSTG, *SYMLNK, *S36, *TBL, *TIMZON, *USRIDX, *USRPRE, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 1
CHGRPTONLY	Changed report only	* <u>NO</u> , *YES	Optional, Positional 2
LIB	Library	<i>Name</i>	Optional
AUTTYPE	Authority type	* <u>OBJECT</u> , *FIELD, *ALL	Optional
FLR	Folder	<i>Character value</i>	Optional
AUTLOBJ	Print AUTL objects	* <u>NO</u> , *YES	Optional
DIR	Directory	<i>Path name</i>	Optional
SCHSUBDIR	Search subdirectory	* <u>NO</u> , *YES	Optional

Top

Object type (OBJTYPE)

This is a required parameter.

The type of object to search for. For a complete list of object types, press the F4 key when prompting this parameter.

object-type

The type of object to be processed.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed reports should be printed.

*NO The full and changed reports are printed.

*YES Only the changed report and the deleted reports are printed.

Top

Library (LIB)

This is a required parameter for all object types except *AUTL, *BLKSF, *CFGL, *CNNL, *COSD, *CTLD, *DEVD, *DIR, *DOC, *FLR, *LIB, *LIND, *MODD, *NWID, *NWSO, *SOCKET, *STMF, *SYMLNK, and *USRPRF.

The name of the library to search for objects to be included in the private authority report.

Top

Authority type (AUTTYPE)

Specifies whether object level authority, field level authority, or both object level and field level authority reports are generated. Field level authority information only applies to *FILE objects.

*OBJECT

Object level authority reports are generated for the specified objects.

*FIELD

For each data base file that has field level authorities a field level authority report is generated.

This value is only valid if *FILE is specified for the **Object type (OBJTYPE)** parameter.

***ALL** For each data base file that has field level authorities, a field level authority report is generated. Also, the object level authority reports for all the files in the specified library are generated.

This value is only valid if *FILE is specified for the **Object type (OBJTYPE)** parameter.

Top

Folder (FLR)

This is a required parameter if *DOC is specified for the **Object type (OBJTYPE)** parameter.

The name of the folder to search for documents to be included in the private authority report.

folder-name

The name of the folder to be searched.

Top

Print AUTL objects (AUTLOBJ)

Specifies whether the Display Authorization List Objects (DSPAUTLOBJ) command will be run for each of the authorization lists on the system. DSPAUTLOBJ provides a list of all the objects that are secured by a specific authorization list. This parameter is only used if the object type is *AUTL. It is ignored for all other object types.

***NO** The DSPAUTLOBJ command will not be run for each of the authorization lists on the system.

***YES** The DSPAUTLOBJ command will be run for each of the authorization lists on the system. The output for the command will be sent to the same output queue as the authorization list report.

Top

Directory (DIR)

This is a required parameter if *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK is specified for the **Object type (OBJTYPE)** parameter.

The name of the directory to search for objects to be included in the private authority report. Only local objects in the Root, QOpenSys, and User-Defined file systems are supported.

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

directory-name

The name of the directory to be searched.

Top

Search subdirectory (SCHSUBDIR)

Specifies whether to search the subdirectories for objects to be included in the private authority report.

Note: This parameter is only used when OBJTYPE is *BLKSF, *DIR, *SOCKET, *STMF, or *SYMLNK.

***NO** The subdirectories are not searched.

***YES** The subdirectories are searched.

Top

Examples

```
PRTPVTAUT OBJTYPE(*FILE) LIB(PAYROLLLIB)
```

This command prints the full, changed, and deleted reports for all file objects in the library PAYROLLLIB.

Top

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

Top

Print Queue Authority (PRTQAUT)

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

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

The Print Queue Authority (PRTQAUT) command allows you to print a report of the output queue and job queue authority information for the objects in the specified library. This command provides a way to check the authority attributes of the output queue and job queue objects on the system.

This command will print two reports for a library. The first report (Full Report) will contain all of the output queues and job queues in the specified library. The second report (Changed Report) will contain the output queues and job queues that have been created or had the authority attributes changed since the PRTQAUT command was last run for the library. If the PRTQAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional queue information is available then the 'Changed Report' will be printed but there will be no queues listed.

The reports will contain the following information:

- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each output queue and job queue that exists on the system. Each entry contains the following information:
 - The name of the library the queue is in.
 - The name of the queue.
 - The object type of the queue.
 - The owner of the queue.
 - The public authority of the queue.
 - The display data value of the output queue. For job queue objects this field will be set to *NONE.
 - The operator control value of the queue.
 - The authority to check value of the queue.

The file QSECQOLD in library QUSRSYS contains information from the last time the PRTQAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the '*' will be replaced with a 'Q' in the member name. System file QASECQF in library QSYS with format name of QSECQF is the model file for the QSECQOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR	Optional, Positional 1

Keyword	Description	Choices	Notes
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2

Top

Library (LIB)

This is a required parameter.

The name of the library to search for output queue and job queue objects to report.

***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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

*USRLIBL

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

*ALLUSR

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

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

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

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

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

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL

program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

name Specify the name of the library to be searched.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

*NO The full and changed reports will be printed.

*YES Only the changed report will be printed.

Top

Examples

```
PRTQAUT LIB(QUSRSYS)
```

This command prints both full and changed reports for the output queues and job queues in the library QUSRSYS.

Top

Error messages

*ESCAPE Messages

CPFB307

Command &1 in use in another job.

Top

Print Subsystem Description (PRTSBSDAUT)

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

Parameters
Examples
Error messages

The Print Subsystem Description Authority (PRTSBSDAUT) command allows you to print a report of the subsystem descriptions in a library that contain a default user in a subsystem description entry. This command provides a way to check for subsystem descriptions that allow work to be performed on your system while running under a default user profile.

This command will print two reports for a library. The first report (Full Report) will contain all of the subsystem descriptions that contain a default user in a subsystem description entry. The second report (Changed Report) will contain the subsystem descriptions that have been changed to contain a subsystem entry with a default user since the PRTSBSDAUT command was last run for the library. If the PRTSBSDAUT command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional subsystem descriptions contain entries with a default user, then the 'Changed Report' will be printed but there will be no subsystem descriptions listed. Changes to user profile special authorities will not cause a 'Changed Report' to be generated.

The reports will contain the following information:

- The name of the library that was specified on the command.
- The date and time the report was last run (only shown on the Changed Report).
- An entry for each subsystem description that contains a subsystem entry with a default user specified. Each entry contains the following information:
 - The name of the library the subsystem description is in.
 - The name of the subsystem description.
 - The owner of the subsystem description.
 - The name of the default user profile specified in the subsystem entry.
 - The special authorities associated with the user profile. The special authorities that are shown are all of the special authorities that would be available when the subsystem entry is used. The special authorities shown are the special authorities that the user has, plus the special authorities that the user's group profiles have (if the user has any groups).

The file QSECSBDOLD in library QUSRSYS contains information from the last time the PRTSBSDAUT command was run for a library. There is a member within the file, with the same name as the library, for each library that has been previously specified on the command. If a special value is specified for the library name (for example, *USRLIBL), then the '*' will be replaced with a 'Q' in the member name. System file QASECSBF in library QSYS with format name of QSECSBF is the model file for the QSECSBDOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR, *ALLAVL, *ALLUSRAVL	Required, Positional 1
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2

Top

Library (LIB)

This is a required parameter.

The name of the library to search for subsystem descriptions contain a subsystem entry with a default user profile specified.

.

***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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

***ALLUSR**

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

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

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

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

```
QDSNX      QRCLxxxxx   QUSRDIRDB  QUSRVI
QGPL       QSRVAGT    QUSR IJS   QUSRVRxMx
QGPL38     QSYS2      QUSRINFSKR
QMGTC      QSYS2xxxxx QUSRNOTES
QMGTC2     QS36F      QUSROND
QMPGDATA   QUSER38    QUSRPOSGS
```

QMOMDATA	QUSRADSM	QUSRPOSSA
QMOMPROC	QUSRBRM	QUSRPYMSVR
QPFRDATA	QUSRDIRCF	QUSRDRARS
QRCL	QUSRDIRCL	QUSRSYS

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALLAVL**

All libraries in all available ASPs are searched.

***ALLUSRAVL**

All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name Specify the name of the library to be searched.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

***NO** The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Top

Examples

```
PRTSBSDAUT LIB(QSYS)
```

This command prints both full and changed reports for all subsystem descriptions in library QSYS.

Top

Error messages

*ESCAPE Messages

CPFB307

Command &1 in use in another job.

Top

Print SQL Information (PRTSQLINF)

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

Parameters
Examples
Error messages

The Print Structured Query Language Information (PRTSQLINF) command allows you to print information about the SQL statements in a program, SQL package, service program, or job. The information includes the SQL statements, the access plans used during the running of the statement, and a list of the command parameters which are defined either during the precompile of the source member for the object or when SQL statements are run.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	Single values: *JOB Other values: <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OBJTYPE	Object type	*PGM, *SQLPKG, *SRVPGM	Optional, Positional 2

Top

Object (OBJ)

Specifies either the name of the object for which you want SQL information printed or *JOB indicating that the job's SQL information is to be printed. A named object can be a program, an SQL package, or a service program.

Single values

***JOB** The SQL information for the current job is to be printed. The output will only contain information for statements which have been dynamically prepared for the job. It will not contain information for SQL statements in programs, service programs, or SQL packages used by the job.

Qualifier 1: Object

name Specify the name of the program or SQL package for which you want information printed.

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

Object type (OBJTYPE)

Specifies the object type of the object specified for the **Object (OBJ)** parameter.

***PGM** The object is a program.

***SQLPKG**
The object is an SQL package.

***SRVPGM**
The object is a service program.

[Top](#)

Examples

Example 1: Printing SQL Information

```
PRTSQLINF PAYROLL
```

This command will print information about the SQL statements contained in program PAYROLL.

[Top](#)

Error messages

*ESCAPE Messages

SQL9011
Print of SQL information failed.

[Top](#)

Print Stop Word List (PRTSWL)

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

Parameters
Examples
Error messages

The Print Stop Word List (PRTSWL) command is used to print the words from an IBM-supplied or user-created stop word list.

Top

Parameters

Keyword	Description	Choices	Notes
LANGID	Language ID	<i>Character value</i>	Required, Positional 1
TYPE	Stop word list type	*IBM , *USER	Optional

Top

Language ID (LANGID)

Specifies the language identifier (ID) for the stop word list.

This is a required parameter.

Top

Stop word list type (TYPE)

Specifies the type of stop word list to print.

***IBM** The stop word list is IBM-supplied.

***USER**
The stop word list is user-created.

Top

Examples

```
PRTSWL LANGID(ENG) TYPE(*IBM)
```

This command prints the IBM-supplied stop word list with the language ID ENG.

Top

Error messages

*ESCAPE Messages

CPF8725

&1 type stop word list not supported for language.

CPF9899

Error occurred during processing of command.

[Top](#)

Print System Information (PRTSYSINF)

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

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

The Print System Information (PRTSYSINF) command prints system information that should be maintained for disaster recovery and system verification purposes. A record of the contents of your system, such as how your system is customized and what libraries it contains, is important to your upgrade success because the information helps you do the following:

- Plan you upgrade procedures
- Evaluate the success of moving information
- Perform disaster recovery, if necessary

[Top](#)

Parameters

None

[Top](#)

Examples

None

[Top](#)

Error messages

Unknown

[Top](#)

Print System Security Attr (PRTSYSSECA)

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

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

The Print System Security Attributes (PRTSYSSECA) command prints a report of security related system values and network attributes to a spooled file. The report includes the system value or network attribute name, the current value, and the recommended value.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

[Top](#)

Parameters

None

[Top](#)

Examples

PRTSYSSECA

This command prints a report of all security-related system values and network attributes, showing the current value and the recommended value.

[Top](#)

Error messages

*ESCAPE Messages

CPF304

User does not have required special authorities.

[Top](#)

Print Point-to-Point Profile (PRTTCPPTP)

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

Parameters
Examples
Error messages

The Print Point-to-Point Profile (PRTTCPPTP) command is used to print the configuration data for a point-to-point TCP/IP profile. Printer file QPTOCPPP is used to generate the spooled file. The spooled file name will be the same as the point-to-point profile name, and the spooled file user data will be 'PRTTCPPTP'.

Top

Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	<i>Character value</i>	Required, Positional 1

Top

Configuration profile (CFGPRF)

Specifies the point-to-point configuration profile to be printed.

This is a required parameter.

character-value

Specify the name of a valid configuration profile.

Top

Examples

```
PRTTCPPTP  CFGPRF(ANSPROFILE)
```

This command prints the configuration data for point-to-point profile ANSPROFILE. The spooled file name will be ANSPROFILE and the spooled file user data will be 'PRTTCPPTP'.

Top

Error messages

*ESCAPE Messages

TCP83F1

Point-to-point profile &1 not printed.

Top

Print Trace Data (PRTTRC)

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

Parameters
 Examples
 Error messages

The Print Trace (PRTTRC) command formats and writes the trace records to the selected output file. The trace records were written to a set of database files by the ENDTRC (End Trace) command and PRTTRC is used to format these trace records to a spooled output file or to a database output file. If the trace records are written to a spooled output file, printer file QPSRVTRCJ is used. The user data for the spooled file will be the same as the value specified for the DTAMBR (Data member) parameter.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- You must have authority to the library and the database files within that library where the trace data is stored.
- If DLTRC(*YES) is specified, you must have authority to the DLTRC (Delete Trace Data) command.
- The record format of the database output file must match the record format of the IBM-supplied output file QASCTJFL.
- The ENDTRC command that produced the trace data and the PRTTRC command that processes and formats the trace data must be run on the same release of Operating System.

Top

Parameters

Keyword	Description	Choices	Notes
DTAMBR	Data member	Name	Optional, Positional 1
DTALIB	Data library	Name, <u>*CURLIB</u>	Optional
SLTJOB	Select jobs	Single values: <u>*ALLJOBS</u> Other values (up to 10 repetitions): <i>Qualified job name</i>	Optional
	Qualifier 1: Select jobs	Generic name, name, <u>*ALL</u>	
	Qualifier 2: User	Generic name, name, <u>*ALL</u>	
	Qualifier 3: Number	000001-999999, <u>*ALL</u>	
DLTRC	Delete trace	*YES, *NO	Optional
SORT	Sort by	*TIME, <u>*THREAD</u>	Optional
OUTPUT	Output	<u>*PRINT</u> , *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	

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

Top

Data member (DTAMBR)

Specifies the member name for the trace data that you want to print. The member name will be the same as the trace session identifier specified on the Start Trace (STRTRC) and End Trace (ENDTRC) commands. The member name is the same for each of the physical files that contain the trace data.

This is a required parameter.

name Specify the name of the database file member that contains the trace data.

Top

Data library (DTALIB)

Specifies the library that contains the set of database files where the collected trace data is stored.

*CURLIB

The trace data is printed from files in the current library for the job. If no library is specified as the current library for the job, QGPL is used.

name Specify the name of the library that contains the trace data files.

Top

Select jobs (SLTJOB)

Specifies which jobs to include in the trace listing. This allows the user to reduce the size of the trace listing by selecting only a subset of the jobs that were part of the trace. Up to ten qualified job names can be specified.

Single values

*ALLJOBS

All jobs that were part of the trace are included.

Qualifier 1: Select jobs

*ALL All jobs that match the specified job user are included.

generic-name

Specify the generic name of the jobs to be included in the trace listing. 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 included in the trace listing.

Qualifier 2: User

***ALL** All jobs that match the specified job name are included.

generic-name

Specify the generic user name of the jobs to be included.

name Specify the name of the user of the job to be included.

Qualifier 3: Number

***ALL** All jobs that match the specified job name and user name are included.

000001-999999

Specify the job number to further qualify the job name and user name.

Top

Delete trace (DLTTRC)

Specifies whether trace data is deleted after it has been printed.

***YES** The trace data in the database files is deleted after the print has completed.

***NO** The trace data in the database files is saved. The DLTTRC (Delete Trace) command can be used to delete the data when it is no longer needed.

Top

Sort by (SORT)

Specifies how the trace data for each job is sorted in the specified output file.

***THREAD**

The trace data for each job is sorted by thread. If a job has multiple threads, the trace data for each thread is sorted by time.

***TIME**

The trace data for each job is sorted by time. If a job has multiple threads, the trace data for all threads in the job is sorted by time. This can result in the trace output for multiple threads to be intermingled.

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 this command creates the file, the text is "OUTFILE created by PRTRC command" and the public authority is *EXCLUDE.

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

Examples

Example 1: Print and Delete Trace

```
PRTTTC DTAMBR(TRACE8) DTALIB(TRCLIB1) DLTTTC(*YES)
```

This command formats and prints the trace data contained in database file members named TRACE8 in library TRCLIB1. The trace data members are removed after the trace data spooled file has been written. All jobs which were part of the trace will be part of the trace listing.

Example 2: Print Subset Trace

```
PRTTTC DTAMBR(T123456789) DTALIB(QGPL)
        SLTJOB(*ALL/QSYS/QCMN*) DLTTTC(*YES)
```

This command formats and prints the trace data contained in database file members named T123456789 in library QGPL. The trace data members are removed after the trace data spooled file has been written. Only those traced jobs that were started under user profile QSYS and had job names that started with "QCMN" will be part of the trace listing.

Example 3: Print Trace and Sort by Time

```
PRTTTC DTAMBR(MYTRACE) DTALIB(MYTRCLIB)
        DLTTTC(*YES) SORT(*TIME)
```

This command formats and prints the trace data contained in database file members named MYTRACE in library MYTRCLIB. The trace data members are removed after the trace data spooled file has been written. The trace records are sorted by the time the record was collected. If the traced jobs were multithreaded, the trace output is sorted by job, with all threads in that job sorted by time. The resulting output may have trace information for multiple threads intermingled.

Example 4: Print Trace to an Output File

```
PRTTTC DTAMBR(BIGTRACE) DTALIB(TRACELIB) DLTTTC(*YES)
        OUTPUT(*OUTFILE) OUTFILE(MYLIB/MYFILE)
```

This command stores the trace data contained in database file members named BIGTRACE in library TRACELIB into a database output file named MYFILE in library MYLIB. The trace data members named BIGTRACE are removed after the trace data has been written to the database output file.

Top

Error messages

*ESCAPE Messages

CPF39CD

Error occurred during processing of the PRTTTC command.

CPF98A2

Not authorized to &1 command or API.

Print Trigger Programs (PRTRGPGM)

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

Parameters
Examples
Error messages

The Print Trigger Program (PRTRGPGM) command lists the programs which have been defined as a trigger program for the files in the specified library.

This command will print two reports for a library. The first report (Full Report) will contain all of the trigger programs associated with files in the specified library. The second report (Changed Report) will contain the trigger programs that **now** appear in the specified library and were not in the library when the PRTRGPGM command was previously run for the library. If the PRTRGPGM command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional trigger programs are in the specified library, the 'Changed Report' will be printed but there will be no objects listed. Changing the trigger time, trigger event or trigger update condition for a trigger program will not cause a 'Changed Report' to be generated.

The file QSECTRGOLD in library QUSRSYS contains information from the last time the PRTRGPGM command was run for a library. There is a member within the file, with the same name as the library specified, for each library previously specified on the command. System file QAFDTRG in library QSYS with format name of QWHFDTRG is the model file for the QSECTRGOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *LIBL, *USRLIBL, *CURLIB, *ALL, *ALLUSR	Required, Positional 1
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2

Top

Library (LIB)

This is a required parameter.

The name of the library to search for files that have trigger programs.

***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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

***ALLUSR**

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

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

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

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

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRISJS	QUSRVRxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVRxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

name Specify the name of the library to be searched.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

***NO** The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Top

Examples

PRTRGPGM LIB(*ALL)

This command searches all files in all libraries and prints both full and changed trigger program reports.

[Top](#)

Error messages

*ESCAPE Messages

CPF304

User does not have required special authorities.

[Top](#)

Print User Objects (PRTUSROBJ)

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

Parameters
Examples
Error messages

The Print User Objects (PRTUSROBJ) command allows you to print a report of the objects in a library that are not created by IBM. Objects are included in the report if the "Created by user" attribute is not *IBM or QLPINSTALL. Use this command to check for user created objects that are in libraries intended for use only by IBM. For example, you may want to run this program for library QSYS to determine if it contains any non-IBM (user) objects.

Note: Some objects created by IBM will still appear in this report. For example, objects created by a PTF exit program will be included in this report. Objects are excluded from the report only when their "Created by user" attribute is either "*IBM" or "QLPINSTALL."

This command will print two reports for a library. The first report (Full Report) will contain all of the objects that have not been created by IBM. The second report (Changed Report) will contain the object that **now** appear in the specified library and were not in the library when the PRTUSROBJ command was previously run for the library. If the PRTUSROBJ command was not previously run for the library, there will be no 'Changed Report'. If the command has been previously run for the library but no additional objects have been added to the library that were not created by IBM, then the 'Changed Report' will be printed but there will be no objects listed.

The file QSECPUOOLD in library QUSRSYS contains information from the last time the PRTUSROBJ command was run for a library. There is a member within the file, with the same name as the library specified, for each library previously specified on the command. System file QADSPOBJ in library QSYS with format name of QLIDOBJD is the model file for the QSECPUOOLD file.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name	Required, Positional 1
CHGRPTONLY	Changed report only	*NO, *YES	Optional, Positional 2

Top

Library (LIB)

This is a required parameter.

The name of the library to search for objects that were not created by IBM.

Top

Changed report only (CHGRPTONLY)

Specifies whether just the changed report should be printed.

***NO** The full and changed reports will be printed.

***YES** Only the changed report will be printed.

Top

Examples

```
PRTUSROBJ LIB(QSYS) CHGONLY(*NO)
```

This command searches library QSYS for any objects that were not created by IBM and prints both full and changed reports.

Top

Error messages

*ESCAPE Messages

CPFB304

User does not have required special authorities.

Top

Print User Profile (PRTUSRPRF)

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

Parameters
Examples
Error messages

The Print User Profile (PRTUSRPRF) command allows you to print a report containing information for the user profiles on the system. Four different reports can be printed. One contains authority type information, one contains environment type information, one contains password type information, and one contains password level type information.

The authority information report will contain the following information:

- The type of report.
- The criteria used to select user profiles in the report.
- The special authorities being selected on (if 'Select by' is *SPCAUT).
- The user classes being selected on (if 'Select by' is *USRCLS).
- An entry for each user profile that was selected. Each entry contains the following information:
 - The name of the user profile.
 - The names of the user's group profiles.
If the user profile does not have any group profiles, this field will contain '*NONE'. If the user has group profiles, an additional entry will follow the user profile entry for each of the user's groups. The entry will contain the group profile name and the special authorities that the group profile has.
 - An indicator for the special authorities that the user profile has ('X' or ' ').
 - The user class for the user profile.
 - Whether the user profile or its group profile own objects created by this user profile.
 - What authority is given to the user's group profile for newly created objects (if the owner value is *USRPRF).
 - The limited capability value for the user profile.

The environment information report will contain the following information:

- The type of report.
- The criteria used to select user profiles in the report.
- The special authorities being selected on (if 'Select by' is *SPCAUT).
- The user classes being selected on (if 'Select by' is *USRCLS).
- An entry for each user profile that was selected. Each entry contains the following information:
 - The name of the user profile.
 - The name of the user's current library.
 - The name of the user's initial menu, and the library it is in.
 - The name of the user's initial program, and the library it is in.
 - The name of the user's job description, and the library it is in.
 - The name of the user's message queue, and the library it is in.
 - The name of the user's attention program, and the library it is in.

The password information report will contain the following information:

- The type of report.
- The criteria used to select user profiles in the report.

- The special authorities being selected on (if 'Select by' is *SPCAUT).
- The user classes being selected on (if 'Select by' is *USRCLS).
- The value of the QPWDEXPITV system value (for reference if the user's password expiration interval is *SYSVAL).
- An entry for each user profile that was selected. Each entry contains the following information:
 - The name of the user profile.
 - The status of the user profile.
 - The number of sign-on attempts that were not valid.
 - The 'no password' indicator ('X' if the user doesn't have a password, ' ' if it does).
 - Whether the password is managed locally.
 - The user's previous sign-on date.
 - The date the user's password was last changed.
 - The user's password expiration interval.
 - Whether the user's password is set to expired.

The password level information report will contain the following information that can be used to determine if the system is ready to change password levels.

- The type of report.
- The criteria used to select user profiles in the report.
- The special authorities being selected on (if 'Select by' is *SPCAUT).
- The user classes being selected on (if 'Select by' is *USRCLS).
- An entry for each user profile that was selected. Each entry contains the following information:
 - The name of the user profile.
 - The 'password present for level 0 or 1' indicator ('*YES' if the user has a password, '*NO' if the user does not have a password, or '*UNKNOWN' if the password information was not available).
 - The 'password present for level 2 or 3' indicator ('*YES', '*NO', or '*UNKNOWN').
 - The 'password present for NetServer' indicator for Windows 95 and 98 NetServer passwords ('*YES', '*NO', or '*UNKNOWN').

Note: The Display Security Attributes (DSPSECA) command can be used to display the current and pending password level for the system. The password level can be changed by changing the QPWDLVL system value.

Restriction: You must have all object (*ALLOBJ) or audit (*AUDIT) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of information	*ALL, *AUTINFO, *ENVINFO, *PWDINFO, *PWDLVL	Optional
SELECT	Select by	*SPCAUT, *USRCLS, *MISMATCH	Optional
SPCAUT	Special authorities	Single values: *ALL Other values (up to 9 repetitions): *ALLOBJ, *AUDIT, *JOBCTL, *IOSYSCFG, *SAVSYS, *SECADM, *SERVICE, *SPLCTL, *NONE	Optional
USRCLS	User class	Single values: *ALL Other values (up to 5 repetitions): *USER, *SYSOPR, *PGMR, *SECADM, *SECOFR	Optional

Type of information (TYPE)

The type of information that can be printed for the selected user profiles.

***ALL** All of the reports are printed for the selected user profiles.

***AUTINFO**

A report containing the authority type information for the selected user profiles is printed.

***ENVINFO**

A report containing the environment type information for the selected user profiles is printed.

***PWDINFO**

A report containing the password type information for the selected user profiles is printed.

***PWDLVL**

A report containing the password level type information for the selected user profiles is printed. This report can be used to determine which user profiles have passwords that are used at the different password levels.

Select by (SELECT)

Specifies what criteria is used to select the user profiles to include in the report.

***SPCAUT**

User profiles will be selected for the report based on special authorities.

***USRCLS**

User profiles will be selected for the report based on user class.

***MISMATCH**

User profiles will be selected for the report based on their special authorities not being the default values assigned to their user class.

Note: The defaulted special authorities for user classes changed in V3R7. Therefore, when running this report for profiles created prior to V3R7, you may notice a larger than expected number of profiles that do not match the default values.

Special authorities (SPCAUT)

If *SPCAUT was specified for the **Select by (SELECT)** parameter, it specifies which special authorities should be used to select users. User profiles with any of the special authorities specified for this parameter will be included in the report. A maximum of 9 special authorities can be specified.

You can enter multiple values for this parameter.

***ALL** All user profiles will be included in the report.

Or select one or more of the following values (9 maximum):

***ALLOBJ**

User profiles with *ALLOBJ special authority will be included in the report.

***AUDIT**

User profiles with *AUDIT special authority will be included in the report.

***IOSYSCFG**

User profiles with *IOSYSCFG special authority will be included in the report.

***JOBCTL**

User profiles with *JOBCTL special authority will be included in the report.

***SAVSYS**

User profiles with *SAVSYS special authority will be included in the report.

***SECADM**

User profiles with *SECADM special authority will be included in the report.

***SERVICE**

User profiles with *SERVICE special authority will be included in the report.

***SPLCTL**

User profiles with *SPLCTL special authority will be included in the report.

***NONE**

User profiles with no special authorities will be included in the report.

Top

User class (USRCLS)

If *USRCLS was specified for the **Select by (SELECT)** parameter, it specifies that user classes should be used to select users. User profiles with a user class that is specified for this parameter will be included in the report. A maximum of 5 user classes can be specified.

You can enter multiple values for this parameter.

***ALL** All user profiles will be included in the report.

Or select one or more of the following values (5 maximum):

***USER**

User profiles with *USER user class will be included in the report.

***SYSOPR**

User profiles with *SYSOPR user class will be included in the report.

***PGMR**

User profiles with *PGMR user class will be included in the report.

***SECADM**

User profiles with *SECADM user class will be included in the report.

***SECOFR**

User profiles with *SECOFR user class will be included in the report.

Top

Examples

```
PRTUSRPRF TYPE(*ALL) SELECT(*SPCAUT) SPCAUT(*ALLOBJ *SECADM)
```

This command prints all four reports for user profiles that have either *ALLOBJ or *SECADM special authority.

[Top](#)

Error messages

***ESCAPE Messages**

CPFB304

User does not have required special authorities.

CPFB307

Command &1 in use in another job.

[Top](#)

Power Down System (PWRDWNSYS)

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

Parameters
Examples
Error messages

The Power Down System (PWRDWNSYS) command prepares the system for ending and then starts the power-down sequence. All active subsystems are notified that the system is being powered down; no new jobs or routing steps can be started by any subsystem. For example, jobs that are on a job queue will not be started. In addition, jobs that are on a job queue as a result of a Transfer Job (TFRJOB) command will not complete. They are removed from the job queue during the subsequent initial program load (IPL), and their job logs are produced.

When the system is powered down with the *CNTRLD option, a vary off of configuration objects is initiated, but may not complete before the power down completes. When the system is powered down with the *IMMED option, no vary off of configuration objects is performed.

Notes:

1. If network server descriptions are configured on the system, all NWSDs should be varied off **before** the PWRDWNSYS command is issued to ensure the integrity of system and user data associated with each network server.
2. If tape units are installed on the system, all tape reels that are on the device(s) should be unloaded **before** the system is powered down to ensure the integrity of data on the tapes.
3. If the system has a primary partition, powering down the primary partition will cause the other partitions to power down. Ensure the other partitions are ready to be powered down before powering down the primary partition.
4. If independent auxiliary storage pool (ASP) devices are configured on the system, all independent ASPs should be varied off **before** PWRDWNSYS command is issued to ensure the integrity of data associated with each independent ASP.
5. The registered Prepower down system exit point (QIBM_QWC_PWRDWNSYS) has two possible formats. Format PWRD0100 can be used to add a program that is called when the PWRDWNSYS command is used. Format PWRD0200 can be used to add one or more programs that are called when the PWRDWNSYS command is used. These exit programs can perform clean up functions before the system is powered down.
6. When changing the QENDJOBLMT and QPWRDWNLMT system values, specify values so that QPWRDWNLMT is greater than QENDJOBLMT. The values need to allow enough time for system-supplied end-of-job functions such as completing commitment control processing and closing database files.

Restrictions:

1. To run this command, the user must have job control (*JOBCTL) authority.
2. When *IMGCLG is specified on the **IPL source (IPLSRC)** parameter you need the following authorities:
 - execute (*EXECUTE) authority to library QUSRSYS
 - use (*USE) authority to the image catalog specified by the **Image catalog (IMGCLG)** parameter
 - use (*USE) authority to the virtual device description
 - execute (*X) authority to each directory in the image catalog path name
3. If you have a user exit program defined to the Prepower down system exit point (QIBM_QWC_PWRDWNSYS) for format PWRD0100, then the user must have use (*USE) authority to

the user exit program and execute (*EXECUTE) authority to the library that contains that program. If not, then this user exit program will not be called and the system will continue to power down.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	*CNTRLD, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	1-99999, <u>3600</u> , *NOLIMIT	Optional, Positional 2
RESTART	Restart options	<i>Element list</i>	Optional
	Element 1: Restart after power down	*NO, *YES	
	Element 2: Restart type	*IPLA, *SYS, *FULL	
IPLSRC	IPL source	*PANEL, A, B, D, *IMGCLG	Optional
IMGCLG	Image catalog	<i>Name</i>	Optional
ENDSBSOPT	End subsystem option	Single values: *DFT Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL	Optional
TIMOUTOPT	Timeout option	*CONTINUE, *MSD, *SYSREFCDE	Optional
CONFIRM	Confirm	*ENVVAR, *INTERACT, *YES, *NO	Optional

Top

How to end (OPTION)

Specifies whether the system allows the active subsystem to end processing of active jobs in a controlled manner (which lets the application program perform end processing), or whether the system ends the jobs immediately. In either case, the system does perform certain job-cleanup functions.

*CNTRLD

The subsystem, in the time specified by the **Controlled end delay time (DELAY)** parameter ends all active jobs in a controlled manner. During that time, programs running in those jobs are allowed to perform cleanup (end-of-job processing). When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. If the possibility exists that an active job could begin to loop or send an inquiry message to QSYSOPR, you should specify a time delay using the DELAY parameter.

*IMMED

The subsystem ends all active jobs immediately. When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job and the QENDJOBBLMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs running in those jobs are not allowed to perform any cleanup. A minimum amount of time is required when *IMMED is specified. The amount of time allowed for cleanup when *IMMED is specified is controlled by the system values QENDJOBBLMT and QPWRDWNLMT.

Note: The *IMMED value might cause undesirable results if data has been partially updated. This value should be used only after a controlled end has been attempted unsuccessfully.

Note: When *IMMED is specified while the system is operating under auxiliary power, or if the delay time specified in the **Controlled end delay time (DELAY)** parameter ends while the system is under auxiliary power, the system is immediately powered-down without additional job cleanup activity.

Top

Controlled end delay time (DELAY)

Specifies the amount of time, in seconds, that the system allows a controlled end to be performed by the active subsystems. If the end of job routines are not finished in the specified delay time, any remaining jobs are ended immediately.

3600 The amount of time in which to complete a controlled end of processing is limited to 3600 seconds.

*NOLIMIT

The system does not power down until the last job is complete.

Note: If *NOLIMIT is specified, a batch job could begin to loop, and the system does not power down.

1-99999

Specify the maximum amount of delay time, in seconds, in which a controlled end can be performed.

Top

Restart after power down (RESTART)

Specifies whether the system ends and powers down, or whether the system ends and then starts again in unattended mode.

The second element of this parameter specifies the point from which the initial program load (IPL) restarts. Specifying *SYS rather than *FULL can reduce the time required to restart the system.

Element 1: Restart after power down

***NO** The system ends and powers down.

***YES** If the system is on utility power, it undergoes end of processing (but does not power down) and then does an abbreviated IPL. If the system is on auxiliary power, it powers down and an automatic-IPL occurs when utility power is restored (if QPWRRSTIPL system value is set to '1'). When the system starts again or when an automatic-IPL occurs, the IPL proceeds in an unattended mode. In unattended mode, displays such as the IPL options display are not shown.

Element 2: Restart type

***IPLA** The value specified on the Change IPL Attributes (CHGIPLA) command is used. To determine the current setting for this value, use the Display IPL Attributes (DSPIPLA) command.

***SYS** The operating system is restarted. The hardware will only be restarted when required by the system.

***FULL** All portions of the system, including the hardware, are restarted.

Top

IPL source (IPLSRC)

Specifies whether an initial-program-load (IPL) is started from the A-source, B-source or D-source of the system. This parameter allows you to control which Licensed Internal Code (LIC) storage source of the system to IPL. Also, the source of the system determines where LIC program temporary fixes (PTFs) are applied. This parameter also allows the system to be upgraded to a new release from an install image on DASD.

Source Considerations

LIC has three storage areas known as the A-source, the B-source and the D-source. The D-source is the install media. The A- and B-sources are part of the system memory. Initially, the A- and B-sources are identical, but when Licensed Internal Code fixes are performed temporarily (PTF), the temporary fixes are stored on the B-source. When the same fixes become permanent, they are copied from the B-source to the A-source; therefore, the fixes reside on both the A-source and the B-source.

When you want to send temporary fixes to the B-source, you must start the system from the A-source, which causes the fixes to be sent to the B-source.

When you start the system from the A-source, you are running the system from the permanent fixes. When you start the system from the B-source, you are running the system from a mixture of temporary and permanent fixes. When you start the system from the D-source, you are using the Licensed Internal Code loaded from the install media.

It is recommended that you specify RESTART(*YES); otherwise, you cannot be assured which source of the system is actually started. This precaution can save you some time.

*PANEL

The system is started from the source that is currently shown on the operator's panel, the A-source, the B-source, or the D-source.

- A** The system is started from the A-source.
- B** The system is started from the B-source.
- D** The system is started from the D-source, the install media.

*IMGCLG

The system is started from the image catalog specified with the **Image catalog (IMGCLG)** parameter. RESTART(*YES) must be used when this option is selected.

Top

Image catalog (IMGCLG)

Specifies the image catalog used when IPLSRC(*IMGCLG) is selected. After the system is powered down, an install using the specified image catalog is performed. See the Work with Catalog Entries (WRKIMGCLGE) command for more information. RESTART(*YES) must be used when this parameter is specified.

name Specify the name of the image catalog in library QUSRSYS.

Top

End subsystem option (ENDSBSOPT)

Specifies the options to take when ending the active subsystems. In general, specifying these options will improve the performance of the PWRDWNSYS command. Each option has certain side effects that you need to analyze before using that option.

This parameter has no effect on jobs that are already in the ending status.

***DFT** The subsystems will end with no special ending options.

- Joblogs will be produced.
- The run priority will not change.
- The timeslice value will not change.

***NOJOBLOG**

No joblogs will be created for jobs that are ended due to this command being invoked. This includes subsystem monitor jobs and all user jobs in the subsystem. This option can significantly reduce the amount of time necessary to complete the PWRDWNSYS command. However, if a problem occurs in a job, there will be no joblog to record the problem, which may make problem diagnosis difficult or impossible.

Note: If OPTION(*IMMED) is specified, then no joblogs are produced during PWRDWNSYS regardless of the **End subsystem option (ENDSBSOPT)** parameter. However, these joblogs will still be produced on the next IPL of the system unless the *NOJOBLOG option is specified. Therefore, if you specify OPTION(*IMMED) ENDSBSOPT(*NOJOBLOG), the system will not power down more quickly, but the subsequent IPL may be faster.

***CHGPTY**

The CPU priority of jobs that are ending is changed to a higher value (worse priority). The remaining active jobs on the system may have better performance when *CHGPTY is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

***CHGTSL**

The timeslice of jobs that are ending is changed to a lower value. The remaining active jobs on the system may have better performance when *CHGTSL is specified. However, jobs that are ending may take longer to finish. This option is ignored if the system is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Top

Timeout option (TIMOUTOPT)

Specifies the option to take when the system does not end within the time limit specified by the QPWRDWNLMT system value. If this time limit is exceeded, the subsequent IPL will be abnormal regardless of the value specified for this parameter.

***CONTINUE**

The system will ignore the timeout condition and continue powering the system down. If RESTART(*YES) is specified, the system will restart automatically. A minimum of information will be available for service to debug the system.

***MSD** The system will issue a main store dump which can be used by service to debug the system. If the main store dump manager is configured correctly, the system will restart after the dump is finished.

***SYSREFCDE**

The system will display system reference code B900 3F10 and the system will stop. This will allow service to debug the system.

Confirm (CONFIRM)

Specifies whether the request should be confirmed before the system is powered down.

*ENVVAR

The value in environment variable QIBM_PWRDWNSYS_CONFIRM is used to determine whether the request should be confirmed. If the value is set to *INTERACT, *YES, or *NO, the action described below for that value is taken. If the environment variable is not defined or not set to one of these values, then a confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. System initiated power downs do not use the environment variable.

*INTERACT

A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. There is no confirmation when the PWRDWNSYS command is issued in a non-interactive job.

***YES** A confirmation panel is displayed when the PWRDWNSYS command is issued in an interactive job. An inquiry message is sent to QSYSOPR when the PWRDWNSYS command is issued in a non-interactive job.

***NO** There is no confirmation when the PWRDWNSYS command is issued.

Top

Examples

Example 1: Performing An Immediate End

```
PWRDWNSYS  OPTION(*IMMED)
```

This command causes the system to perform an immediate end without allowing any active jobs to perform cleanup routines. Once the system completes its end functions, it starts the power-down sequence.

Example 2: Specifying a Controlled End

```
SBMJOB    JOB(LASTJOB)  JOBQ(QBATCH)  JOBPTY(9)  JOBQ(QBATCH)
          RQSDTA('PWRDWNSYS *CNTRLD 3600')
```

This command submits a low priority batch job that, when run, causes the system to perform a controlled end. The controlled end is allowed one hour (3600 seconds) for completion before any remaining jobs are ended. This method of issuing the PWRDWNSYS command could be used to allow other higher priority jobs on job queue QBATCH (including those that are on the queue as a result of the Transfer Job (TFRJOB) command) to be completed before the PWRDWNSYS command is run. There must be an active subsystem for which the QBATCH job queue is a source of work.

Example 3: Specifying a Controlled End With No Time Limit

```
PWRDWNSYS  OPTION(*CNTRLD)  RESTART(*YES)
```

This command causes the system to perform a controlled end with no time limit. When all jobs in the system have completed, the system prepares for ending and starts an IPL.

After PWRDWN SYS OPTION(*CNTRLD) is entered, and before the delay time ends, this command can be overridden by entering PWRDWN SYS OPTION(*IMMED). In this case, the values specified or defaulted for the RESTART parameter on the second command also override the values specified or defaulted for the first command.

Example 4: Changing the IPL Source After Immediate End

```
PWRDWN SYS  OPTION(*IMMED)  RESTART(*YES)  IPLSRC(A)
```

This command causes the system to end immediately and change the IPL source to A. When the system restarts, it IPLs on the A source.

Example 5: Allowing the Operating System to Determine the Restart Point

```
PWRDWN SYS  OPTION(*IMMED)  RESTART((*YES *SYS))
```

This command causes the IPL to restart at the point determined by the operating system.

Example 6: Changing the Time Out Option.

```
PWRDWN SYS  OPTION(*IMMED)  TIMOUTOPT(*MSD)
```

This command causes the system to end immediately. If the QPWRDWN LMT system value is exceeded, the system will dump the main storage. If the main store dump manager is configured correctly, the system will restart. Otherwise, the B900 3F10 system reference code will be displayed and the system will halt.

Example 7: Installing a New Release of the Operating System.

```
PWRDWN SYS  RESTART(*YES)  IPLSRC(*IMGCLG)  IMGCLG(MYCAT1)
```

This command causes the system to end and then start installing a new release of the operating system from the image catalog MYCAT1.

Top

Error messages

*ESCAPE Messages

CPF1001

Wait time expired for system response.

CPF1036

System powering down with *CNTRLD option.

CPF1037

System powering down with *IMMED option.

CPF1038

No authority to use command.

CPF1091

Function check occurred in system arbiter.

CPF18C7

PWRDWNSYS not allowed to continue.

CPFBC42

Verification for image catalog &1 failed.

[Top](#)

Query Document Library (QRYDOCLIB)

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

Parameters
 Examples
 Error messages

The Query Document Library (QRYDOCLIB) command allows you to search for documents within the document library. A document list is created containing the results of the search. You can also copy information about the documents that satisfy the search request into a database file for processing.

When the QRYDOCLIB command is run, a document list object is created. The document list object is created regardless of whether an output file is produced unless the user specifies *NONE for the DOCL parameter. This document list object can be used by the SAVDLO command.

Restrictions:

- The current user of this command must have the authority to work on behalf of the specified user ID address. To work on behalf of other users, the user must have special permission granted with the Grant User Permission (GRTUSRPMN) command.
- The format of the output file must be the same as OSIQDL of the system file, QSYS/QAOSIQDL.
- If several QRYDOCLIB commands are run at the same time, the document list name (**Document list (DOCL)** parameter) and the output file name or member name (**File to receive output (OUTFILE)** parameter) must be different for each of these QRYDOCLIB commands.

Top

Parameters

Keyword	Description	Choices	Notes
QRYDFN	Query definition	Single values: <u>*NONE</u> Other values: <i>Element list</i>	Optional
	Element 1: Start query	*IF	
	Element 2: Search criteria	Values (up to 49 repetitions): <i>Element list</i>	
	Element 1: Profile	*ACTDATE, *ALWRPL, *ASP, *AUTHOR, *CHGDATE, *CMPDATE, *CPYLST, *CRTDATE, *DOCCLS, *DOCD, *DOCDATE, *DOCTYPE, *EXPDATE, *FILDATE, *IDXDATE, *KWD, *OWNER, *PROJECT, *REF, *REVDATE, *STATUS, *SUBJECT, *USEDATE	
	Element 2: Relational operator	*EQ, *GT, *LT, *NE, *GE, *NL, *LE, *NG, *CT, *BG	
	Element 3: Compare value	<i>Character value</i> , *YES, *NO	
	Element 4: Logical operator	<u>*AND</u> , *OR	
FLR	In folder	Single values: <u>*ALL</u> Other values (up to 100 repetitions): <i>Character value</i> , *NONE	Optional
SCHSUBFLR	Search subfolder	<u>*NO</u> , *YES	Optional

Keyword	Description	Choices	Notes
QRYTXT	Query text	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Start query	*IF	
	Element 2: Search criteria	Values (up to 30 repetitions): <i>Element list</i>	
	Element 1: Phrase	<i>Character value</i>	
	Element 2: Type of matching	*ALL, *EXACT	
	Element 3: Allow synonyms	*NO, *YES	
	Element 4: Logical operator	*OR, *AND, *ANDNOT	
TXTLANGID	Language ID	<i>Character value, *JOB</i>	Optional
DOCL	Document list	<i>Character value, *DFT, *NONE</i>	Optional
TEXT	Text	<i>Character value, *BLANK</i>	Optional
OUTFILE	File to receive output	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	*REPLACE, *ADD	
OUTDTATYP	Type of data for output	Single values: *DFT, *ALL Other values (up to 21 repetitions): <i>*ACTDATE, *AUTHOR, *CHGDATE, *CMPDATE, *CPYLST, *CRTDATE, *DOCCLS, *DOCD, *DOCDATE, *EXPDATE, *FILCAB, *FILDATE, *IDP, *IDXDATE, *KWD, *PROJECT, *REF, *REVDATE, *STATUS, *SUBJECT, *USEDATE</i>	Optional
USRID	User identifier	Single values: *CURRENT Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
TIMLMT	Time limit	1-9999, *NOMAX	Optional
SELLMT	Selection limit	1-32767, *NOMAX	Optional
ORDER	Order by	Single values: *NONE Other values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Profile	<i>*ACTDATE, *ASP, *AUTHOR, *CHGDATE, *CMPDATE, *CPYLST, *CRTDATE, *DOCCLS, *DOCD, *DOCDATE, *DOCTYPE, *EXPDATE, *FILDATE, *IDXDATE, *KWD, *OWNER, *PROJECT, *REF, *REVDATE, *STATUS, *SUBJECT, *USEDATE</i>	
	Element 2: Selection order	*ASCEND, *DESCEND	
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

Query definition (QRYDFN)

Specifies what query definition selection values are used to select the documents. The values you specify on this parameter are used to search the document library. If values other than *NONE are specified on both the **Query text (QRYTXT)** parameter and this parameter, only documents that match both sets of values are selected. If you specify *NONE on both parameters, all documents to which you are authorized are selected.

Single values

***NONE**

No query definition is used to select the documents.

Element 1: Start query

***IF** A query definition is used to select the documents.

Element 2: Search criteria

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set must contain exactly four values:

1. The name of the document profile parameter to be compared (from the list that follows)
2. One of the relational operator values (from the list that follows)
3. The compare value
4. One of the logical operators *AND, *OR, or *ANDNOT

Values 1 and 3 are compared for the relationship specified by value 2. Each QRYDFN relational set must be enclosed in parentheses. A maximum of 49 sets of values can be specified.

Element 1: Profile

Specifies the name of the document profile parameter to be compared.

***ACTDATE**

Action due date

***ALWRPL**

Allow document replacement

***ASP** Auxiliary storage pool ID

***AUTHOR**

Document author

***CHGDATE**

Last change date

***CMPDATE**

Completion date

***CPYLST**

Copy list

***CRTDATE**

Create date

***DOCCLS**

Document class

*DOCD	Document description
*DOCDATE	Document date
*DOCTYPE	Document type
*EXPDATE	Expiration date
*FILDATE	File date
*IDXDATE	Last indexed date
*KWD	Keyword
*OWNER	Document owner
*PROJECT	Document project
*REF	Reference
*REVDATE	Last revision date
*STATUS	Document status
*SUBJECT	Document subject
*USEDATE	Last used date

Element 2: Relational operator

The relational operator indicates the relationship that must exist between the profile parameter contents in the document and the value specified as the compare value parameter of the query definition for the relationship to be true.

The *CT operator is used to perform a context search. It asks the system to determine whether the character string specified by the compare value is contained anywhere in the profile parameter.

The *BG operator is used to perform a search that compares the compare value with the beginning of the profile parameter. The profile parameter is truncated or extended as necessary to match the length of the specified value. It asks the system to determine whether the character string specified by the value is contained at the beginning of the profile parameter.

Some operators are not allowed for some profile parameters. In this case, a diagnostic message followed by an escape message is sent.

The following cases are not valid:

- The *ALWRPL (allow document replacement) is a YES/NO switch. The *EQ operator is the only operator allowed with *ALWRPL.

- The *CT and *BG operators are not allowed with the *ASP value or date values such as *CRTDATE and *EXPDATE.

*EQ	Equal
*GT	Greater than
*LT	Less than
*NE	Not equal
*GE	Greater than or equal
*NL	Not less than
*LE	Less than or equal
*NG	Not greater than
*CT	Contains
*BG	Begins

Element 3: Compare value

Specifies the compare value to be used in the search.

character

Specify the value to compare with the contents of the specified profile parameter. The parameter value must be specified in apostrophes if it contains blanks or special characters.

The *ALWRPL field has two special values: *YES and *NO. When these are specified with the *ALWRPL field, they are changed to internal values for the indicator. When *YES or *NO are specified for the text field, they are used like they are.

The *OWNER field is an 8-character user ID followed by its address. Trailing blanks cannot be omitted from the user ID. For example, if the user ID is JMDOE and the address is SYSTEM1, the query request would be:

```
(*IF ((*OWNER *EQ 'JMDOE  SYSTEM1'))).
```

If the user ID is JIMSMITH, the query request would be:

```
(*IF ((*OWNER *EQ 'JIMSMITHSYSTEM1'))).
```

Dates must be entered in the system date format.

Note: If one of the date profile parameters is specified and the compare value is '(blank)', then the compare value is equated to 01/01/01 or 01/001 for Julian date. '/' is changed depending on the system separator value QDATSEP.

The allowable length for the search fields is limited by the Document Interchange Architecture (DIA) search database. When the length of the value is greater than the maximum, the value is truncated to the allowed length. The maximum lengths are:

Value Maximum Length

*DOCD
44 characters

- ***DOCCLS**
16 characters
- ***SUBJECT**
60 characters
- ***AUTHOR**
20 characters
- ***KWD**
60 characters
- ***CPYLST**
60 characters
- ***OWNER**
16 characters
- ***REF** 60 characters
- ***STATUS**
20 characters
- ***PROJECT**
10 characters

For all operators except *CT and *BG, if a value is specified that is shorter than the profile parameter value, then the specified value is extended with blanks to match the length of the profile parameter.

The case (upper, lower, or mixed) that is used to enter the original parameter value or the case of the comparison value do not matter. The system changes both the entered comparison value and the original parameter value to upper case before making a comparison.

Element 4: Logical operator

The logical operators are used to group conditions. The first **AND** operator encountered signifies that a condition group starts with the condition immediately preceding the **AND** operator. Subsequent conditions with the **AND** operator are added to the condition group. The first condition encountered containing the **OR** operator or the last condition in the query definition ends the condition group.

*AND The profile parameter value relational groups on both sides of the *AND value must all be satisfied before a document is selected.

*OR If the parameter value relational group on either side of the *OR value is satisfied, the document is selected.

Top

In folder (FLR)

Specifies the folders to search for the documents that match the search values specified on the **Query definition (QRYDFN)** parameter and the **Query text (QRYTXT)** parameter.

*ALL All the folders on the system are searched.

***NONE**

Documents not located in any folder are searched.

name Specify the name of the folders to search for the documents. This is the only folder searched. A folder name can consist of a series of folder names (FLR1/FLR2/etc.) if the documents being searched for are located in a folder contained in another folder. A maximum of 100 folders can be specified and each folder name can be a maximum of 63 characters in length.

Top

Search subfolder (SCHSUBFLR)

Specifies whether subfolders of the folder specified on the **In folder (FLR)** parameter are searched.

***NO** Subfolders are not searched.

***YES** Subfolders of the specified folder are searched.

Top

Query text (QRYTXT)

Specifies the text search values used to select documents. The values you specify on this parameter are used to search the text index. If values other than *NONE are specified on both the **Query definition (QRYDFN)** parameter and this parameter, only documents that match both sets of values are selected. If you specify *NONE on both parameters, all documents to which you are authorized are selected.

Single values

***NONE**

No text search values are entered.

Element 1: Start query

***IF** Text search values are used in the document search.

Element 2: Search criteria

To specify the conditions under which documents are selected, a set of values is specified for each condition. Each set contains four values:

1. A phrase, which the system compares to entries in the text search index
2. One of the 'type of matching' values
3. One of the 'allow synonyms' values
4. One of the logical operators

A maximum of 30 sets of values can be specified. Each set must be enclosed in parentheses.

Element 1: Phrase

Specifies a phrase which the system compares to entries in the text search index.

character

Specify a phrase of one or more words. Do not use any punctuation. When specifying phrases, you can:

- use an asterisk (*) to mask a whole word within a phrase. For example, if you want to search for documents referring to various annual reports, you can specify the phrase:
annual * report

The search results will include documents containing such phrases as annual budget report, annual progress report, and annual sales report. The search results will also include documents containing the phrase 'annual report' without a word in between.

When using a word mask, you must specify a word before and after the asterisk. A word mask at the beginning or end of a phrase is ignored.

- use an asterisk (*) to mask part of a word within a phrase. The mask can be used at the beginning, middle, or end of a word. For example, if you want to search for documents referring to word processing, you can specify the phrase:
word process*

The search results will include documents containing such phrases as word processing, word processor, and word processed.

- use a question mark (?) to mask one or more characters in a word. For example, if you want to search for documents referring to the various spellings of Johnson, you can specify the phrase:
j?hns?n

The search results will include documents containing such phrases as johnson, johnsen, and jahnsn.

Element 2: Type of matching

Specifies a type of matching value to be used in the search.

***ALL** The phrase must be contained within one sentence, but the words do not have to be in the specified order.

***EXACT** The phrase must be contained within one sentence and the words must be in the specified order.

Element 3: Allow synonyms

Specifies whether synonyms are to be used in the search.

***NO** No synonyms are used.

***YES** Synonyms for each word in the phrase, if available, are also used to compare to entries in the text index.

Note: Using synonyms may affect the performance of the request by causing more words to be searched for, and possibly by causing more documents to be selected.

Element 4: Logical operator

Specifies a logical operator to be used in the search.

***OR** If the phrase on either side of the *OR value is found, the document is selected.

***AND** If the phrases on both sides of the *AND value are found, the document is selected.

***ANDNOT**

If the phrase following the *ANDNOT value is not found, the document is selected.

Top

Language ID (TXTLANGID)

Specifies the language identifier for the phrases in the query text. This parameter is required if the **Query text (QRYTXT)** parameter is specified; it is not allowed if the QRYTXT parameter is not specified or has a value of *NONE.

***JOB** The language identifier specified for the job in which this command is entered is used.

character

Specify a language identifier. Press the PF4 key when prompting the **Language ID (TXTLANGID)** parameter to see a list of valid identifiers.

Top

Document list (DOCL)

Specifies the name of the document list. The document list contains a pointer to each document in the document library that is qualified for search. This list is a copy of the library at the time the search was run. As documents are deleted from the library or added to the library, the document list is not updated. The document library list name is specified with the name of the user requesting the search.

***DFT** A system created name is used as the default name. The default list is the same as the user ID on the **User identifier (USRID)** parameter.

***NONE**

No document list is created.

character

Specify the name of the document list. A maximum of 8 characters can be used.

Top

Text (TEXT)

Specifies the text that briefly describes the object.

***BLANK**

No text is specified.

character

Specify a maximum of 50 characters, enclosed in apostrophes.

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, the system uses QAOSIQDL in QSYS with a format name of OSQDL 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

Type of data for output (OUTDTATYP)

Specifies that certain information about the selected documents is written to the output file if a value is specified on the **File to receive output (OUTFILE)** parameter.

***DFT** The document information record is written to the output file. This is the same as specifying *DOCD. The record code is 105.

***ALL** All information records about the document are written.

***ACTDATE**

The action due date record is written. The record code is 135.

***AUTHOR**

The author records are written. The record code is 145.

- *CHGDATE**
The date last changed record is written. The record code is 130.
- *CMPDATE**
The completion date record is written. The record code is 140.
- *CPYLST**
The copy list records are written. The record code is 150.
- *CRTDATE**
The create date record is written. The record code is 110.
- *DOCCLS**
The document class record is written. The record code is 155.
- *DOCD**
The document description record is written. The record code is 105.
- *DOCDATE**
The document date record is written. The record code is 120.
- *EXPDATE**
The expiration date record is written. The record code is 115.
- *FILCAB**
The file cabinet reference record is written. The record code is 160.
- *FILDATE**
The file date record is written. The record code is 125.
- *IDP** The interchange document profile (IDP) is written. The record code is 500.
- *IDXDATE**
The last indexed date record is written to the output file.
- *KWD**
The keyword records are written. The record code is 170.
- *PROJECT**
The project record is written. The record code is 185.
- *REF** The reference record is written. The record code is 175.
- *REVDATE**
The date of the last revision to the document content is written to the output file.
- *STATUS**
The status record is written. The record code is 180.
- *SUBJECT**
The subject records are written. The record code is 165.
- *USEDATE**
The date last used record is written. The record code is 200.

Top

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

Single values

***CURRENT**

You are performing the request for yourself.

Element 1: User ID

character

Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have all object (*ALLOBJ) special authority.

Element 2: Address

character

Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Top

Time limit (TIMLMT)

Specifies the amount of time allowed for the requested search to run.

*NOMAX

No time limit for the search is set. All qualified documents are searched.

1-9999 Specify the maximum time limit (in minutes) that the search runs. A two-hour limit is specified as TIMLMT(120). If the search has not been completed when the time limit is reached, the search ends with an informational message followed by a completion message. The output file, and if specified the document list, will contain the documents found within the specified time limit.

Top

Selection limit (SELLMT)

Specifies the allowed number of documents to select in the search.

*NOMAX

No document limit for the search is set. All qualified documents are selected, up to the system limit of 32,767.

1-32,767

Specify the maximum number of documents to select. If there are more documents than the set limit, the document list and the output file contain information about the selected documents up to this limit and an informational message indicating that the limit was reached. The completion message indicates the number of documents selected.

Top

Order by (ORDER)

Specifies that the selected documents are to be ordered (ascending or descending) when placed in the created document list or output file. The order is defined for one or more document profile parameters specified, up to a maximum of 5.

Note: If a value other than *NONE is specified on the **Query text (QRYTXT)** parameter, ordering is not allowed.

When a user specifies an order to the search request, the performance of the request may be affected. The request performs best if no order is specified.

Single values

***NONE**

No order is applied to the selected documents.

Element 1: Profile

***ACTDATE**

The returned documents are ordered by the action due date.

***ASP** The returned documents are ordered by the auxiliary storage pool ID (ASPID) parameter.

***AUTHOR**

The returned documents are ordered by the author.

***CHGDATE**

The returned documents are ordered by the last changed date.

***CMPDATE**

The returned documents are ordered by the completion date.

***CPYLST**

The returned documents are ordered by the copy list.

***CRTDATE**

The returned documents are ordered by the create date.

***DOCCLS**

The returned documents are ordered by the document class.

***DOCD**

The returned documents are ordered by the document description.

***DOCDATE**

The returned documents are ordered by the document date.

***DOCTYPE**

The returned documents are ordered by the document type profile parameter. Valid values range from 2 through 65535.

***EXPDATE**

The returned documents are ordered by the expiration date.

***FILDATE**

The returned documents are ordered by the date the documents were filed.

***IDXDATE**

The returned documents are ordered by the last indexed date profile parameter. Text search services must be installed if this value is specified.

***KWD**

The returned documents are ordered by the keyword.

***OWNER**

The returned documents are ordered by the name of the owner user profile name.

***PROJECT**

The returned documents are ordered by the project.

***REF** The returned documents are ordered by the reference.

***REVDATE**

The returned documents are ordered by the last content revision date.

***STATUS**

The returned documents are ordered by the status.

***SUBJECT**

The returned documents are ordered by the subject.

***USEDATE**

The returned documents are ordered by the last used date.

Element 2: Selection order

***ASCEND**

The returned documents are ordered in the ascending collating sequence.

***DESCEND**

The returned documents are be ordered in the descending collating sequence.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

The CMDCHRID parameter applies to the following parameters and means that the data is translated to the code page and character set common to all documents in the search database. That character set and code page is '697 500', except for the **User identifier (USRID)** parameter, which is '930 500'.

The following parameters are translated:

- **User identifier (USRID)**
- **Document list (DOCL)**
- **Query definition (QRYDFN)**
- **Query text (QRYTXT)**
- **Text (TEXT)**

Single values

***SYSVAL**

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEV D**

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Examples

```
QRYDOCLIB  USRID(*CURRENT)  OUTFILE(*NONE)  DOCL(MYLIST)
           QRYDFN(*IF ((*DOCD *EQ DOCDESC *AND)
                       (*DOCCLS *BG CLASS *OR)
                       (*FILDATE *LE '06/13/88'))))
```

This command searches for documents that meet the following search conditions: document description equals DOCDESC and document class starts with Class; or the file date is on or before 06/13/88. The results of the search will be stored in the document list MYLIST.

[Top](#)

Error messages

*ESCAPE Messages

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF905D

Query of document library failed.

CPF9096

Cannot use CMDCHRID(*DEVLD), DOCCHRID(*DEVLD) in batch job.

CPF9860

Error occurred during output file processing.

[Top](#)

Query Distributions (QRYDST)

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

Parameters
 Examples
 Error messages

The Query Distribution (QRYDST) command allows you to request distribution information either for yourself or on behalf of another user.

Restrictions:

1. If you request distribution information for another user, you must have been given permission to work on behalf of that user with the GRTUSRPMN command.
2. If user ID(*ALLAUT) is specified, and if you do not have permission to work on behalf of the other user, only information about your own distributions is returned.
3. The requester of the command must be enrolled in the system distribution directory. Personal distribution cannot be requested if the requester is working on behalf of another user.
4. The DLTSTS parameter does not apply to incoming distributions. When OPTION (*IN) is specified, the DLTSTS parameter is ignored.
5. Personal distribution cannot be queried if the requester is working on behalf of another user.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Incoming or outgoing	*IN, *OUT	Optional, Positional 2
USRID	User identifier	Single values: *CURRENT, *ALLAUT Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
DLTSTS	Delete status	*NO, *YES	Optional
OUTFILE	File to receive output	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	*REPLACE, *ADD	
STATUS	Status	*ALL, *NEW, *OLD, *OPENED, *UNOPENED, *LOCAL, *REMOTE, *FILEPND, *DELETED, *DAMAGED	Optional
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Incoming or outgoing (OPTION)

Specifies the type of distribution information that is returned.

***IN** Information about incoming distribution is returned.

***OUT** Information about outgoing distribution is returned. An outgoing distribution is the status being saved by the system for a distribution sent to one or more users with confirmation of delivery requested.

Top

User identifier (USRID)

Specifies the user ID and the user ID address associated with this request.

***CURRENT**
Your distribution information is returned.

***ALLAUT**
Your distribution information and information for all users who have given you permission to work on their behalf is returned.

user-ID
Specify the user ID of the user whose distribution information is being returned. You must have permission to work on behalf of the user specified on this parameter if the named user ID is not your own.

user-ID-address
Specify the user ID address of the user whose distribution information is being returned. You must have permission to work on behalf of the user specified on this parameter if the named user ID address is not your own.

Top

Delete status (DLTSTS)

Specifies whether the distribution status being kept for outgoing distributions is deleted from the system.

***NO** The distribution status is not deleted from the system.

***YES** The distribution status is deleted if all receivers have returned status.

Note: Other products use this status information. Care should be taken not to delete information that other products use to track their distribution.

Top

File to receive output (OUTFILE)

Specifies the name of the database file to which the output is directed. If the output file does not exist, a database file is created in the specified library. If the file is created by this function, the text description will read: 'OUTFILE is created by QRYDST', and the authority for users with no specific authority 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.

If a new file is created and *IN is specified on the **Incoming or outgoing** prompt (OPTION parameter), the system uses QAOSILIN in QSYS with a format name OSLIN as a model.

If a new file is created and *OUT is specified on the **Incoming or outgoing** prompt (OPTION parameter) the system uses QAOSILOT in QSYS with a format name OSLOUT as a model.

*NONE

The output is not directed to a database file. If *NONE is specified, the output from this command is a completion message containing the number of distributions on the Document Interchange Architecture (DIA) Distribution Recipient Index (*DRX) for the option and user ID and address specified.

When *IN is specified on the **Incoming or outgoing** prompt (OPTION parameter), the total count of new and old status is returned. After each QRYDST, any mail with new status is changed to old status.

Note: Office Vision/400 disregards the old mail status and treats both old and new mail as if they have new status.

data-base-file-name

Specify the name of the database file and library that receives the output of the display.

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

library-name

Specify the library where the database file is located.

Top

Output member options (OUTMBR)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible **member to receive output** values are:

*FIRST

The first member in the file receives the output. If the member does not exist, the system creates a member with the name of the file specified on the **File to receive output** prompt (OUTFILE parameter).

member-name

Specify the name of the file member that receives the output. If the member does not exist, the system creates the file member. If the member already exists, the system adds records to the end of the member or clears the member and then adds the records.

The possible **add or replace** values are:

*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

Status (STATUS)

Specifies the mail entry status of the distribution for which you are requesting information. This parameter is valid only if *IN is specified on the **Incoming or outgoing** prompt (OPTION parameter) and an output file is specified on the **File to receive output** prompt (OUTFILE parameter).

- *ALL** Distribution information is returned regardless of the distributions' mail entry status.
- *NEW** Distribution information is returned only for those distributions whose mail entry status is NEW.
- *OLD** Distribution information is returned only for those distributions whose mail entry status is OLD, which indicates that the distribution has been queried once but has not been processed.
- *OPENED**
Distribution information is returned only for those distributions whose mail entry status is OPENED.
- *UNOPENED**
Distribution information is returned for those distributions whose mail entry status is either NEW or OLD.
- *LOCAL**
Distribution information is returned only for those distributions whose mail entry status is LOCAL, which indicates that the distribution has been filed on the local system.
- *REMOTE**
Distribution information is returned only for those distributions whose mail entry status is REMOTE, which indicates that the distribution has been filed on a remote system.
- *FILEPND**
Distribution information is returned only for those distributions whose status is FILEPND, which indicates that the distribution is being filed on the local system or a remote system, but the file operation has not finished.
- *DELETED**
Distribution information is returned only for those distributions whose status is DELETED, which indicates that the document referred to by the distribution has been deleted.
- *DAMAGED**
Distribution information is returned only for those distributions whose status is DAMAGED, which indicates that the document referred to by the distribution is damaged.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

The value specified on the **User identifier** prompt (USRID parameter) is translated to character set and code page '930 500'.

Single values

- *SYSVAL**
The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.
- *DEVVD**
The system determines the graphic character set and code page values from the display device

description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Examples

```
QRYDST  USER(*CURRENT)  OPTION(*IN)
        OUTFILE(*CURLIB/MYFILE)  OUTMBR(*FIRST *ADD)
```

This command requests information about incoming distributions for the current user of this command. The output is directed to the database file MYFILE in the user's current library and is added to the first member in that file.

Top

Error messages

*ESCAPE Messages

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF907C

&1 requested distributions completed, acknowledge failed.

CPF9096

Cannot use CMDCHRID(*DEVVD), DOCCHRID(*DEVVD) in batch job.

CPF9097

Query distribution request failed.

CPF9845

Error occurred while opening file &1.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9860

Error occurred during output file processing.

Top

Query Problem Status (QRYPRBSTS)

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

Parameters
Examples
Error messages

The Query Problem Status (QRYPRBSTS) command retrieves problem status information from *IBMSRV (RETAIN) or from another system that is enlisted as a service provider.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
PRBID	Problem identifier	Character value, *PMR	Required, Positional 1
ORIGIN	Origin	Element list	Optional
	Element 1: Network identifier	Communications name, *NETATR	
	Element 2: Control point name	Communications name, *NETATR	
RMTCPNAME	Remote control point	Communications name, *IBMSRV, *SELECT	Optional
RMTNETID	Remote network identifier	Communications name, *NETATR	Optional
SRVID	Service number	Character value	Optional
BRANCH	Branch number	Character value, *SRVID	Optional
COUNTRY	Country or region number	Character value, *SRVID	Optional
AUTOPRBCRT	Auto problem create	*YES, *NO	Optional

Top

Problem identifier (PRBID)

Specifies the problem identifier of the problem log entry. Problems with different system origins can have the same identifier. This parameter can be used with the ORIGIN parameter to select a single problem from a particular system.

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

Note: A value must be specified for the **Service number (SRVID)** parameter when PRBID is *PMR.

character-value

Specify a problem identifier.

This is a required parameter.

Top

Origin (ORIGIN)

Specifies the node of the system from which the problem log entry originated. This parameter is used with the PRBID parameter to uniquely identify the problem.

Element 1: Network identifier

*NETATR

The LCLNETID value specified in the system network attributes is used.

communications-name

Specify a network identifier.

Element 2: Control point name

*NETATR

The LCLNETID value specified in the system network attributes is used.

communications-name

Specify a control point name.

Top

Remote control point (RMTCPNAME)

Specifies the destination of the service provider to whom the service request is sent.

*IBMSRV

The service request is sent to IBM service support.

*SELECT

A list of service providers is shown from which the user can select the destination the service request is sent to.

communications-name

Specify the name of the control point that is the destination of the request.

Top

Remote network identifier (RMTNETID)

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

*NETATR

The service provider is in the local network.

communications-name

Specify the network name of the service provider to whom the request is sent.

Top

Service number (SRVID)

Specifies the service assigned number for the problem log entry. This number was assigned when the problem was reported to IBM service support.

character-value

Specify the service assigned number for the problem log entry.

Branch number (BRANCH)

Specifies the branch number for the problem log entry. This number was assigned when the problem was reported to IBM service support.

*SRVID

The branch number on an existing problem log entry in the current system that contains the specified service number (SRVID parameter) will be used. If no problem log entry with the specified service number exists, a valid branch number must be specified.

character-value

Specify the three-digit branch number for the problem log entry. All three digits must be specified.

Country or region number (COUNTRY)

Specifies the country or region number for the problem log entry. This number was assigned when the problem was reported to IBM service support.

*SRVID

The country or region number on an existing problem log entry in the current system that contains the specified service number (SRVID parameter) will be used. If no problem log entry with the specified service number exists, a valid country or region number must be specified.

character-value

Specify the three-digit country or region number for the problem log entry. All three digits must be specified.

Auto problem create (AUTOPRBCRT)

Specifies whether a problem should automatically be created, if a problem does not exist on the system.

*YES Create a problem.

*NO Do not create a problem.

Examples

Example 1: Querying Problem Status on Another System

```
QRYPRBSTS  PRBID(1234567890)  RMTCPNAME(SYSTEM99)
           RMTNETID(IBMNETID)  AUTOPRBCRT(*YES)
```

This command searches for the status of a specific problem on another system (SYSTEM99).

Example 2: Querying IBM Service

QRYPRBSTS PRBID(*PMR) RMTCPNAME(*IBMSRV) RMTNETID(*NETATR)
SRVID(23774) AUTOPRBCRT(*YES)

This command searches the IBM Service database for the status of PMR 23774.

Top

Error messages

*ESCAPE Messages

CPF7AA7

Problem &1 not found or in use.

CPF7AD4

Network ID &1 not in correct format.

CPF7A84

Query status request routed to different system than specified.

CPF7A88

Error indicated in reply to request.

CPF7A9A

Remote control point and network identifier not valid.

CPF7A9B

Problem &1 cannot be queried.

CPF7A97

Invalid service identifier.

CPF7A98

Service identifier not allowed.

CPF7A99

Query must be sent to *IBMSRV.

CPF7B18

Control point &1 not in correct format.

CPF8C08

Cannot specify *SELECT for the control point name.

CPF8C09

&1 not defined as a service provider.

CPF8C24

Error occurred while processing request.

CPFC680

Branch number not allowed.

CPFC681

Country or region number not allowed.

CPFC682

Branch number not valid.

CPFC683

Country or region number not valid.

***STATUS Messages**

CPZ7A80

Sending query status request to &1.

Top

Query TIE Files (QRYTIEF)

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 Query Technical Information Exchange File (QRYTIEF) command allows you to find out whether files are ready to be received from the remote support network. A message is returned that specifies the size of the largest file that is to be received.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

QRYTIEF

This command sends a message that specifies the number of files to be received from the remote support network and the size of the largest file to be received.

[Top](#)

Error messages

None

[Top](#)

Start QSH (QSH)

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

Parameters
Examples
Error messages

The Start QSH (STRQSH) command, as known as QSH, starts the **qsh** shell interpreter.

If run in an interactive job, STRQSH starts an interactive shell session. If a shell session is not already active in the job, then:

1. A new shell session is started and a terminal window is displayed.
2. **qsh** runs the commands from the file `/etc/profile` if it exists.
3. **qsh** runs the commands from the file `.profile` in the user's home directory if it exists.
4. **qsh** runs the commands from the file specified by the expansion of the ENV variable if it exists.

If a shell session is already active in an interactive job, you are reconnected to the existing session. From the terminal window, you can enter shell commands and view output from the commands.

Using the Terminal Window

The terminal window has two parts:

- an input line for entering commands, and
- an output area that contains an echo of the commands you entered and any output generated by the commands.

The terminal window supports the following function keys:

F3 (Exit)

Close the terminal window and end the qsh session.

F5 (Refresh)

Refresh the output area.

F6 (Print)

Print the output area to a spool file.

F7 (Up)

Roll output area up one page.

F8 (Down)

Roll output area down one page.

F9 (Retrieve)

Retrieve a previous command. You can press this key multiple times to retrieve any previous command. For example, to retrieve the second to last command you entered, press this key two times. You can also select a specific command to be run again by placing the cursor on that command and pressing this key. When the interactive job is running in a double-byte CCSID, this key is not available.

F11 (Toggle line wrap)

Toggle the line wrap/truncate mode in the output area. In line wrap mode, lines longer than the width of the terminal window are wrapped to the next line. In truncate mode, the portion of a line beyond the width of the terminal window is not shown.

F12 (Disconnect)

Disconnect from the **qsh** session. This key only closes the terminal window and does not end the **qsh** session. You can redisplay the disconnected **qsh** session by running STRQSH again.

F13 (Clear)

Clear the output area.

F17 (Top)

Display top of output area.

F18 (Bottom)

Display bottom of output area.

F19 (Left)

Shift output area to the left.

F20 (Right)

Shift output area to the right.

F21 (CL command entry)

Display a command entry window where you can enter CL commands.

Also, you can use SysReq 2 to interrupt the currently running command.

Error messages for STRQSH

ESCAPE Messages*QSH0002**

Error found with QSH session, reason code &1, errno &2.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	Character value, <u>*NONE</u>	Optional

Top

Command (CMD)

Specifies the shell command to be run.

The possible values are:

***NONE:**

No command is provided and an interactive session is started.

command

A shell command to run. The command can be a maximum of 5000 bytes in length. If a blank or other special characters are used, the command must be enclosed in apostrophes ('). If an apostrophe is intended, two apostrophes must be used (").

Note: The case is preserved when lowercase characters are specified.

Top

Examples

None

[Top](#)

Error messages

*ESCAPE Messages

QSH0002

Error found with QSH session, reason code &1, errno &2.

[Top](#)

Qualifier Definition (QUAL)

Parameters
Examples
Error messages

The Qualifier (QUAL) command definition statement describes one part of a qualified name. If a name is the allowable value of a parameter or list item defined in a PARM or ELEM statement, it can be changed to a qualified name by using a QUAL statement for each qualifier used to qualify the name.

The order in which the QUAL statements are entered into the source file determines the positional order in which the qualifiers must be specified and passed to the validity checker and the command processing program. The first qualification of a qualified name must be either a simple name, a generic name, or a defined special value.

The QUAL statement (or only the *first* QUAL statement if there are more than one) *must* have a statement label that matches the statement label value that must be specified in a PARM or ELEM statement for which the qualifier is being defined. The qualifiers for the parameter or list item are then entered on the command in the form: value3/value2/value1, where values 1 through 3 are qualifiers that are each described by a QUAL statement. The values are passed to the command processing program in the same order, with the periods removed, and with each value padded to its maximum length.

Note: The QUAL statement contains certain parameters and predefined values that can be used only when IBM-supplied command processing programs are called by the command being defined. Because of limitations in some high-level languages, these values may not be useful in the definition statements of user-defined commands. These parameters and values are identified by the phrase (*For IBM-supplied commands*) that immediately follows the parameter keyword (if the entire parameter is for IBM-supplied commands only) or the predefined value to which it applies.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of value	*NAME, *GENERIC, *CHAR, *INT2, *INT4, *SNAME, *CNAME, *UINT2, *UINT4, *CNAME	Required, Positional 1
LEN	Length specification	<i>Element list</i>	Optional, Positional 2
	Element 1: Value length	<i>Integer</i>	
CONSTANT	Constant value	<i>Character value</i>	Optional
RSTD	Restricted values	*NO, *YES	Optional
DFT	Default value	<i>Character value</i>	Optional
VALUES	Valid values	Values (up to 300 repetitions): <i>Character value</i>	Optional
REL	Relational expression	<i>Element list</i>	Optional
	Element 1: Relational operator	*GT, *EQ, *GE, *NL, *LT, *NE, *LE, *NG	
	Element 2: Value	<i>Character value</i>	
RANGE	Range of values	<i>Element list</i>	Optional
	Element 1: Lower value	<i>Character value</i>	
	Element 2: Upper value	<i>Character value</i>	

Keyword	Description	Choices	Notes
SPCVAL	Special values	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: From value	<i>Character value</i>	
	Element 2: To replacement value	<i>Character value</i>	
MIN	Minimum values required	0, 1	Optional
ALWUNPRT	Allow unprintable characters	*YES, *NO	Optional
ALWVAR	Allow variable names	*YES, *NO	Optional
FULL	Full field required	*NO, *YES	Optional
EXPR	Value an expression	*NO, *YES	Optional
VARY	Varying length	Single values: *NO Other values: <i>Element list</i>	Optional
	Element 1: Return length value	*YES	
	Element 2: Value length	*INT2, *INT4	
PASSATR	Pass attribute byte	*NO, *YES	Optional
DSPINPUT	Display input	*YES, *PROMPT, *NO	Optional
CHOICE	Choice text	<i>Character value</i> , *VALUES, *NONE, *PGM	Optional
CHOICEPGM	Choice program	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Choice program	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
INLPMTLEN	Initial prompt length	*CALC, *PWD, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 25, 32, 50, 80, 132, 256, 512	Optional
PROMPT	Prompt text or message ID	<i>Character value</i> , *NONE	Optional

Top

Type of value (TYPE)

Specifies the type of qualifier used to qualify a parameter name or list element name. The qualifier can be a name or generic name, a quoted or not quoted character string, or an integer. Enter one of the following options to specify the type of qualifier. The first qualifier for any qualified name must have a type of name (*NAME) or generic name (*GENERIC).

*NAME

The qualifier is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters, \$, @, or #. The remaining characters can be alphanumeric, a period, an underscore, or one of special characters, \$, @, or #. The name can also be a string of characters starting and ending with double quotation marks (") or enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it should be specified on the **Special values (SPCVAL)** parameter.

*SNAME

The qualifier is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters \$, @, or #. The remaining characters can be alphanumeric, an underscore, or one of the special characters \$, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the **Special values (SPCVAL)** parameter.

*CNAME

The qualifier is a character string that represents a name. The maximum length of the name is 256 bytes. The first character must be alphabetic or one of the special characters, \$, @, or #. The

remaining characters can be alphanumeric or one of special characters, \$, @, or #. The character string can be enclosed in parentheses. If a special value is used (as in *LIBL or *NONE), it must be specified on the **Special values (SPCVL)** parameter.

***GENERIC**

The qualifier is a character string that represents a generic name. A generic name contains one or more characters followed by an asterisk (*); the name identifies a group of objects whose names all begin with the characters preceding the asterisk (*). If an asterisk is not included, the system assumes that the generic name is a complete object name.

***CHAR**

The qualifier is a character string that can (optionally) be enclosed in apostrophes. If the character string contains any special characters (not including an asterisk (*)), it *must* be enclosed in apostrophes. The maximum length of the character string is 5000 bytes if *JOB is specified for the CCSID parameter, or 10000 bytes if *UTF16 is specified for the CCSID parameter.

***INT2** The qualifier is an integer that is passed as a 2-byte signed binary number.

***INT4** The qualifier is an integer that is passed as a 4-byte signed binary number.

***UINT2**

The qualifier is an integer that is passed as a 2-byte unsigned binary number.

***UINT4**

The qualifier is an integer that is passed as a 4-byte unsigned binary number.

Top

Length specification (LEN)

Specifies the maximum length of the qualifier, if *NAME, *SNAME, *CNAME, *GENERIC, or *CHAR is specified for the **Type of value (TYPE)** parameter.

The value specified for this parameter will be the number of **bytes** passed for the qualifier value to the command processing program. If a shorter value is specified for this qualifier, the value will be padded on the right with blanks.

If the LEN parameter is allowed but not specified, a default length will be used based on the value specified for the TYPE parameter. The following table shows the default length value for each TYPE and the maximum value that can be specified for the LEN parameter.

TYPE	Default Length	Maximum Length
*CHAR	32	5000
*NAME	10	256
*GENERIC	10	256
*SNAME	10	256
*CNAME	10	256

Top

Constant value (CONSTANT)

Specifies that a value is passed to the command processing program as a constant for the qualifier when the command being defined is processed; the qualifier is not to appear externally on the command. If specified, the value must satisfy the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Valid values (VALUES parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

If a character constant is specified in this parameter, it can be no longer than 32 bytes.

If a constant is specified in this QUAL statement and other QUAL statements immediately follow it, they must also be defined as constants, unless a label precedes one of them. A label indicates the beginning of a new group of QUAL statements, which can be defined differently.

Also, if a constant is specified for the qualifier being defined, no prompt text can be specified for the **Prompt text or message ID (PROMPT)** parameter of this QUAL statement. However, any other qualifiers or groups of qualifiers are still prompted, and their values are still passed to the command processing program as a qualified name.

This parameter is not valid if the **Default value (DFT)** parameter is specified or if *YES is specified for the **Value an expression (EXPR)** parameter.

Variables cannot be coded for this parameter.

Top

Restricted values (RSTD)

Specifies whether the value entered for the qualifier is restricted to only one of the values given in the **Valid values (VALUES)** parameter or the **Single values (SNGVAL)** parameter, or whether any value can be used that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

***NO** The value entered for the qualifier defined by this QUAL statement can be anything that satisfies the requirements specified by the following parameters:

- Type of value (TYPE parameter)
- Length specification (LEN parameter)
- Relational expression (REL parameter)
- Range of values (RANGE parameter)
- Special values (SPCVAL parameter)

***YES** The value entered for the qualifier in this QUAL statement is restricted to one of the values in the **Valid values (VALUES)** parameter, or to one of the from-values in the **Special values (SPCVAL)** parameter.

Default value (DFT)

Specifies the default value assigned to the qualifier if a value is not specified by the user. The default value must satisfy one of the following:

- It must match the qualifier requirements specified by the following parameters:
 - Type of value (TYPE parameter)
 - Length specification (LEN parameter)
 - Relational expression (REL parameter)
 - Range of values (RANGE parameter)
- It must be one of the from-values in the **Special values (SPCVAL)** parameter.
- If *YES is specified for the **Restricted values (RSTD)** parameter, it must be in the list of values in the **Valid values (VALUES)** parameter or in the list of from-values in the **Special values (SPCVAL)** parameter.
- If the default is a character constant, it can have no more than 32 bytes.

This parameter is valid only if the **Minimum values required (MIN)** parameter is 0, which means the qualifier defined by this QUAL statement for this list is optional. A default is not meaningful on this QUAL statement if it is the first one (defining the first part) for a qualified name and if a default is specified on the PARM or ELEM statement that this QUAL statement further defines. If this parameter is not specified, it has a default of its own: the default is blank if *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter. The default is zero (0) if *INT2, *INT4, *UINT2 or *UINT4 is specified for the **Type of value (TYPE)** parameter. An *assumed* default value is not displayed by the command prompt; a blank input field is shown instead. If a default is specified in this parameter, it is displayed by the prompt exactly as specified.

The DFT parameter is not valid if the **Constant value (CONSTANT)** parameter is specified.

value Specify the default value that meets the specified requirements or that is one of the values specified in the **Valid values (VALUES)** parameter or the **Special values (SPCVAL)** parameter.

Variables cannot be coded for this value.

Top

Valid values (VALUES)

Specifies a list of up to 300 constants (fixed values) from which one constant can be entered as the value of the qualifier. This parameter is valid only if all of the following are true:

- *YES is specified for the **Restricted values (RSTD)** parameter.
- Both the **Range of values (RANGE)** parameter, and the **Relational expression (REL)** parameter are *not* specified,
- The constant matches the attributes specified by the **Type of value (TYPE)** parameter, and the **Length specification (LEN)** parameter in this QUAL statement.

Character constants specified in this parameter can be no longer than 32 bytes. Specify the constants (not more than 300) that can be entered as the value of the qualifier.

Top

Relational expression (REL)

Specifies the relationship between the qualifier value and the value of another parameter or constant. To specify the relationship, enter one of the following relational operators followed by a constant or the value of another parameter.

- *LT less than
- *LE less than or equal to
- *EQ equal to
- *GE greater than or equal to
- *GT greater than
- *NL not less than
- *NE not equal to
- *NG not greater than

This parameter is not valid if either the **Valid values (VALUES)** parameter or the **Range of values (RANGE)** parameter is specified. If *CHAR (character type) is specified by **Type of value (TYPE)** parameter, the EBCDIC value of the character string is used as an unsigned integer in the comparison. If a character constant is specified in this parameter, it can be no longer than 32 bytes.

Top

Range of values (RANGE)

Specifies the range, or limits, for the value of the qualifier. The qualifier value must be greater than or equal to the lower limit value specified, and it must be less than or equal to the upper limit value specified. For nonnumeric data types, such as *CHAR, the range of values and the data specified is right-justified and padded on the left with blanks. A numeric range should not be used to define an interval for nonnumeric data unless leading zeros are specified or the data is only 1 character in length. This parameter is not valid if either the **Valid values (VALUES)** parameter, or the **Relational expression (REL)** parameter is specified. Character constants specified in this parameter can be no longer than 32 bytes.

Top

Special values (SPCVL)

Specifies a list of up to 300 entries that define special values that can be entered on the parameter named in the **Keyword (KWD)** parameter on the PARM statement. Each entry specifies a character string (a from-value) that can be entered even though it may not meet all validity checking requirements. If the entered character string matches the from-value of one of the entries, and the to-value is specified, the string is replaced with the to-value and is then passed to the command processing program without further checking. If the to-value is omitted, the from-value is passed to the command processing program. The from-value is a character string, but the to-value can be anything that is passable. If a CL variable is used for the from-value, its type must be *CHAR. However, the first qualifier can only have special to-values with the from-values that are a name, a generic name, or an asterisk (*) followed by a name such as *ALL.

Each to-value must be passable to the command processing program. The to-value must be no longer than is specified on the **Length specification (LEN)** parameter and, if *INT2, *INT4, *UINT2 or *UINT4 is specified for the **Type of value (TYPE)** parameter, the type of the to-value must be the same. If a character type (such as *CHAR or *NAME) is specified for the **Type of value (TYPE)** parameter, the

to-value must be a character string. Character constants specified in this parameter can be no longer than 32 bytes. If a to-value is not specified, the from-value must be passable.

If a to-value of *CURLIB is specified, the name of the current library is passed to the command processing program rather than the value *CURLIB. If the from-value is *CURLIB and no to-value is specified, or if the to-value is *CURLIB and it is enclosed in apostrophes, the value *CURLIB is passed to the command processing program.

Variables cannot be coded for this value.

Top

Minimum values required (MIN)

Specifies whether the qualifier being defined in this QUAL statement is required or optional. If this parameter is not specified, 0 is assumed, which means the qualifier is optional. If a required qualified name is needed, 1 must be specified for this parameter on both the first QUAL and on the PARM or ELEM that refers to it.

- 0 The qualifier is optional on the name being qualified.
- 1 The qualifier is required on the name being qualified; it must be entered.

Top

Allow unprintable characters (ALWUNPRT)

Specifies whether the qualifier value should accept the hexadecimal character X'FF' or those hexadecimal characters in the range of X'00' to X'3F'. This parameter is valid only if *CHAR or *X is specified for the **Type of value (TYPE)** parameter,

- *YES All characters can be sent to the display or printer.
- *NO Unprintable characters cannot be passed to the command processing program.

Top

Allow variable names (ALWVAR)

Specifies whether variable names are allowed for the qualifier. *NO is not allowed if *VARNAME, *ZEROELEM, *NULL, or a statement label is specified for the **Type of value (TYPE)** parameter,

- *YES Variable names can be used for the qualifier.
- *NO Variable names cannot be used for the qualifier.

Top

Full field required (FULL)

Specifies whether the number of characters in the qualifier value must be exactly the same as the number specified in the **Length specification (LEN)** parameter (if specified) or its default length (if LEN is not specified).

- *NO The number of characters in the qualifier value can be less than that specified by the LEN parameter.
- *YES The number of characters in the qualifier value must equal the number specified by the LEN

parameter or the default length for that type. The exact length is valid only if *CHAR, *NAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter.

Top

Value an expression (EXPR)

Specifies whether the qualifier can accept an expression containing a character concatenation.

***NO** The qualifier value cannot be a concatenation expression.

***YES** The qualifier value can be a concatenation expression.

Top

Varying length (VARY)

Specifies whether the qualifier value that is passed to the command processing program is preceded by a length value that indicates the number of bytes entered for the qualifier's value.

Single values

***NO** The qualifier value is not preceded by a length value.

Element 1: Return length value

***YES** The qualifier value passed to the command processing program is preceded by a binary length field that indicates the number of bytes actually specified for the qualifier. *YES is valid only if *CHAR, *NAME, *SNAME, *CNAME, or *GENERIC is specified for the **Type of value (TYPE)** parameter. *YES must be specified if PASSATR(*YES) and RTNVAL(*YES) are specified.

Note: The length value is the actual number of bytes entered for the command parameter with trailing blanks removed. The length value passed may be different than the defined parameter length or the declared variable length. The length of the field containing the character string data is determined by the defined length for the parameter or the declared LEN for CL Program variables. The length value defines how many bytes in the character string data field were actually entered for the command parameter.

Element 2: Value length

***INT2** The qualifier length value is an integer passed as a 2-byte signed binary number.

***INT4** The qualifier length value is an integer passed as a 4-byte signed binary number.

Top

Pass attribute byte (PASSATR)

Specifies whether an attribute byte is to be passed to the command processing program with the qualifier. The attribute byte precedes the qualifier data.

***NO** No attribute byte is passed with the qualifier.

***YES** An attribute byte is passed with the qualifier.

The attribute byte has two fields:

1. The leftmost bit of the attribute byte indicates whether or not a value was specified. If the leftmost bit is '0'B, the value passed to the command processing program is a default value

and was not specified in the command string. If the leftmost bit is '1'B, the value passed to the command processing program was specified in the command string.

2. The remaining seven bits describe the value passed to the command processing program when *CHAR is specified for the **Type of value (TYPE)** parameter.

Attribute	Description
'0000010'B	Meets *NAME rules, like A_B
'0000100'B	Meets *GENERIC rules, like AB*
'1000101'B	Quoted character string, like 'A B'
'0000101'B	Unquoted character string, like 5A
'1001000'B	Logical constant, '0' or '1'
'0001100'B	Hexadecimal value, like X'C1C2'
'0100001'B	Unsigned numeric value, like 5
'0101001'B	Unsigned numeric with decimal point, like 5.2
'0110001'B	Signed numeric value, like -5
'0111001'B	Signed numeric with decimal point, like -5.2

Top

Display input (DSPINPUT)

Identifies whether the keyword value is to be shown in the job log or in a prompt display.

***YES** Indicates that the parameter value is shown on the prompt display and in the job log.

***PROMPT**

The response *PROMPT indicates that the parameter value is shown on the prompt display but not in the job log.

***NO** Indicates that the parameter value is not shown on either the prompt display or in the job log.

Top

Choice text (CHOICE)

Specifies the choices text that is displayed to the right of the input field on the prompt screen. Up to 30 characters of text can be displayed.

***VALUES**

The choices text is generated based on the values specified for the TYPE, RSTD, RANGE, SNGVAL, SPCVAL, and VALUES parameters. If constants are specified for the RANGE parameter, the choices text begins with the minimum value and the maximum value separated by a hyphen. If RANGE is not specified with constants as the minimum and maximum values, and RSTD(*NO) is specified, the choices text begins with a short description of the parameter type based on the value specified for the TYPE parameter. Values specified for the SNGVAL parameter are added to the choices text, in the order the values are defined in the command definition source and separated by a comma and a blank. The last entries added to the choices text are values specified for the SPCVAL or VALUES parameter, in the order the values are defined in the command definition source and separated by a comma and a blank. If there are too many values to fit in 30 characters, the last value is followed by three periods.

The following are examples of possible choices text generated by CHOICE(*VALUES):

- If TYPE(*DEC) and RANGE(1.0 999.9) and SPCVAL((*NOMAX -1)) are specified, the choices text will be:
1.0-999.9, *NOMAX
- If TYPE(*NAME) and RSTD(*NO) and SNGVAL(*ALL) and SPCVAL(*LIBL *CURLIB) are specified, the choices text will be:

Name, *ALL, *LIBL, *CURLIB

- If RSTD(*YES) and SNGVAL(*ALL) and SPCVAL(*ALRTBL *BNDDIR *CHTFMT *CLD *CLS *CMD) are specified, the choices text will be:
*ALL, *ALRTBL, *BNDDIR...

***NONE**

No values are displayed.

***PGM** A program that is called determines the values that are displayed. The program that is called is identified in **Choice program (CHOICEPGM)** parameter of the PARM statement.

message-identifier

Specify the message ID of the message used to retrieve the message containing the text for the possible values field. The message file specified on the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command is used to find the message.

'choices-text'

Specify no more than 30 characters, enclosed in apostrophes.

Top

Choice program (CHOICEPGM)

Specifies the program to be called during command prompting to fill in the possible choices text and the permissible values. This parameter must be specified if *PGM is specified on the **Choice text (CHOICE)** parameter and may not be specified otherwise.

Single values

***NONE**

No program is identified to fill in the possible choices text and permissible values.

Qualifier 1: Choice program

name Specifies the name of the program to be called during prompting to fill in the possible choices text or permissible values. If an exception occurs when the program is called, no possible choices text is left blank, and the list of permissible values is taken from the command.

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 name of the library where the program is located.

Top

Initial prompt length (INLPMTLEN)

Specifies the length of the input field initially displayed for the qualifier when the command is prompted. The user can extend the field to a maximum length of 512 bytes by entering an ampersand (&) in the first position of the field, followed by a blank. INLPMTLEN is valid only if TYPE is specified as *NAME, *SNAME, *CNAME, *GENERIC, or *CHAR. If FULL(*YES), RSTD(*YES), or CONSTANT are specified, INLPMTLEN(*CALC) must be specified or defaulted.

*CALC

The prompter will determine the length of the prompt field based on the type and length of the parameter.

***PWD** If the current value of system value QPWDLVL is '0' or '1', the prompt field will be 10 bytes long. Otherwise, the length of the prompt field will be determined by the length of the parameter. INLPMTLEN(*PWD) is valid only if TYPE is specified as *CHAR, *NAME, *SNAME, or *CNAME.

initial-prompt-length

Specify the initial length in bytes. Valid values are 1-12, 17, 25, 32, 50, 80, 132, 256, and 512.

Top

Prompt text or message ID (PROMPT)

Specifies the prompt text, if any, that is used for the qualifier (defined in this QUAL statement). This parameter is not allowed for the first qualifier or for a qualifier for which the **Constant value (CONSTANT)** parameter is specified. The prompt text for the first qualifier comes from the PROMPT parameter of the PARM or ELEM statement pointing to the qualifier. The prompt text gives a short description of the qualifier which appears next to the qualifier input field when the command is prompted.

*NONE

No prompt text is shown for the qualifier defined by this QUAL statement. This qualifier is still prompted by an input field, but no text is shown with it.

message-identifier

Specify the message identifier that specifies the message containing the prompt text of up to 30 bytes that is shown when the program is prompting the qualifier. If a message having the specified identifier cannot be found in the message file specified in the **Message file for prompt text (PMTFILE)** parameter of the Create Command (CRTCMD) command, the message identifier itself is used as the prompt text.

'prompt-text'

Specify the prompt text that is shown when the program is prompting the qualifier. The text must be a character string of no more than 30 bytes, enclosed in apostrophes.

Top

Examples

Example 1: Qualified Job Name as One Element

```
      PARM  KWD(SPLFILE)  TYPE(L1)  DFT(*)  SNGVAL(*)
L1:   ELEM  TYPE(*NAME)  MIN(1)    /* For file name */
      ELEM  TYPE(Q1)
Q1:   QUAL  TYPE(*NAME)  MIN(1)    /* For job name  */
      QUAL  TYPE(*NAME)  /* For user name */
      QUAL  TYPE(*CHAR)  LEN(6)    /* For job number */
```

The SPLFILE parameter is optional and, if not specified, defaults to an asterisk (*). Otherwise, the value consists of a two-element list. The first element is a file name and it is required. The second element is a qualified job name. The first qualifier is required; the last two qualifiers are optional. The following are some examples of valid SPLFILE parameter syntax:

- SPLFILE(*)
- SPLFILE(MYSPLFILE MYJOB)

- SPLFILE(MYSPLFILE 123456/USERA/MYJOB)

Example 2: List of Qualified Object Names as One Element

```

      PARM  KWD(DTAMBR)  TYPE(L1)  DFT(*ALL)  MAX(32)  +
            SNGVAL(*ALL)
L1:  ELEM  TYPE(Q1)  MIN(1)
      ELEM  TYPE(*NAME)  MIN(0)  MAX(32)  SPCVAL(*NONE)  +
            DFT(*NONE)
Q1:  QUAL  TYPE(*NAME)  MIN(1)
      QUAL  TYPE(*NAME)  DFT(*CURRENT)  SPCVAL(*CURRENT)

```

The parameter named DTAMBR is optional and, if not specified, defaults to *ALL. Otherwise, the value consists of a list, each element of which is itself a list. Each sublist consists of a qualified file name optionally followed by one or more member names. If no member name is specified, *NONE is taken as the default. If no library qualifier is specified for the file, *CURRENT is taken as the default. Each sublist can contain one file name and up to 32 member names. Up to 32 such sublists can appear as the value of DTAMBR. The following are some examples of valid DTAMBR parameter syntax:

- DTAMBR(*ALL)
- DTAMBR((PFILE1 *NONE))
- DTAMBR((LIB1/PFILE1 (MBR1 MBR2)))
- DTAMBR((*CURRENT/PFILE1 (MBR1 MBR2 MBR3)) (LIB2/PFILE2 (MBRA MBRB)))

Top

Error messages

None

Top

Reclaim Activation Group (RCLACTGRP)

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

Parameters
Examples
Error messages

The Reclaim Activation Group (RCLACTGRP) command deletes a specified activation group and frees the resources that are scoped to it. It does not reclaim resources scoped to the job or scoped to the default activation group. This command is normally used only in the controlling program of the application.

An activation group is eligible to be reclaimed if it meets the following criteria:

- The activation group is not the default activation group.
The default activation group cannot be reclaimed.
- The activation group is not active.
An activation group cannot be reclaimed if there are programs or procedures running within the activation group.
- The activation group is not one of the debug activation groups.
When the job is in debug mode, the activation groups in use do not appear as active on the Call Stack or Display Activation Group displays.
- The activation group is not a shared activation group.
A shared activation group cannot be reclaimed because it may be in use by another job.

When an activation group is reclaimed, all resources within the scope of the activation group are reclaimed. Resources within the scope of the activation group include static storage for programs in the activation group, open files, user interface manager (UIM) application resources, Common Programming Interface (CPI) Communications conversations, hierarchical file systems (HFS) resources, user-defined communications sessions, and pending changes for the commitment definition.

A close option can be specified on this command, and is used when closing mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files. If an activation group level commitment definition has been started for the activation group, and it has pending committable changes, the close option also indicates whether the system implicitly commits or rolls back the pending changes before ending the commitment definition. When specifying a close option of *NORMAL, and there are no errors when closing files using the activation group level commitment definition, a commit is performed. Otherwise, a rollback is performed. See the Recovering your system book, SC41-5304 book for information on how the system performs the rollback operation under commitment control.

An activation group should only be reclaimed if it will never be needed again within the same job. Otherwise, errors and unpredictable results may occur if other programs later attempt to access the resources that were reclaimed. Therefore, this command should normally only be used in the controlling program of an application. Specifying ACTGRP(*ELIGIBLE) requires full knowledge of the job environment. Otherwise, unpredictable results can occur.

For more information on the appropriate use of this command, see the IBM System Manager for i5/OS.

Restrictions:

1. This command is not threadsafe. This command fails in a job that allows multiple threads.

Top

Parameters

Keyword	Description	Choices	Notes
ACTGRP	Activation group	<i>Name</i> , *ELIGIBLE	Required, Positional 1
OPTION	Close option	* <u>NORMAL</u> , *ABNORMAL	Optional, Positional 2

Top

Activation group (ACTGRP)

Specifies the activation group to be reclaimed.

*ELIGIBLE

All eligible activation groups within the job are deleted.

activation-group-name

Specify the activation group to be reclaimed. The activation group can only be reclaimed if it has no active calls. If active calls exist, a message is displayed informing the user that the request failed. If the activation group is not found, a message is displayed informing the user that the request failed because the activation group was not found.

Top

Close option (OPTION)

Specifies whether to commit or roll back pending changes for an activation group level commitment definition, and whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed. This parameter is ignored for all other files and objects within the scope of the activation group.

*NORMAL

The changes pending for an activation group level commitment definition are committed (if there are no errors when closing files using the commitment definition), and a normal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

*ABNORMAL

The changes pending for an activation group level commitment definition are rolled back and an abnormal close notification is sent to the attached host system when mixed, communications, BSC, and ICF files are closed.

Top

Examples

```
RCLACTGRP ACTGRP(MYGROUP)
```

This command reclaims the activation group MYGROUP.

Top

Error messages

*ESCAPE Messages

CPF1653

Activation group &1 not found.

CPF1654

Activation group &1 cannot be deleted.

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

Top

Reclaim DB Cross-Reference (RCLDBXREF)

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

Parameters
Examples
Error messages

The Reclaim Database Cross-Reference (RCLDBXREF) command recovers the database cross-reference catalog data for a specific library. This command provides a subset of the function provided by the Reclaim Storage (RCLSTG) command to reclaim the database cross-reference catalog data for the system by specifying SELECT(*DBXREF). The difference is that this command does not require the system to be in restricted state, and it supports a reclaim of a specific library, rather than all libraries on the system.

Additionally this command provides an interface for the user to determine which catalogs have inconsistencies, and which libraries are affected.

If an auxiliary storage pool (ASP) group has been set for the current thread, this command will take affect for the independent ASPs in that ASP group and also for the system and basic user ASPs (ASP numbers 1-32).

This command should only be used when problems with the database cross-reference catalogs occur, and RCLSTG SELECT(*DBXREF) is not an option due to critical business requirements.

Because this command does not need to run in restricted state, it may not always be able to recover cross-reference information. In such cases, running RCLSTG SELECT(*DBXREF) will be required.

When the command is used to recover cross-reference information, the user must not allow applications to use or modify objects in a library that is being reclaimed. Failure to do this could cause unpredictable results.

If RCLDBXREF does not correct the problem, the library being reclaimed may have more inconsistencies than existed before using this command. If this happens, running RCLSTG SELECT(*DBXREF) will be required.

Restrictions:

- You must have all object (*ALLOBJ) special authority to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	*CHECK, *FIX	Optional, Positional 1
LIB	Library	Name, *ERR	Optional

Top

Option (OPTION)

Specifies what action the command will take. The command can either check for problems, or attempt to fix them.

***CHECK**

The command should check to see if any cross-reference catalog inconsistencies have been found. As each catalog is inspected, a diagnostic message (CPD32A7) is sent to the job log for each library known to have inconsistent data in that catalog. After all of the cross-reference catalogs have been checked, an escape message (CPF32AC) is signaled if problems were found. If no inconsistencies are found in the catalogs, a completion message (CPC32AC) is sent indicating the cross-reference data appears to be correct.

***FIX** The command should attempt to fix the cross-reference data for the library specified for the **Library (LIB)** parameter.

Note: In some cases, the system may determine that a full reclaim of the database cross-reference catalogs is needed by running the Reclaim Storage (RCLSTG) command and specifying SELECT(*DBXREF). In this case, a CPF32AB escape message is sent and the library's cross-reference data will not be fixed.

When the *FIX option is specified, the user must be careful not to interrupt the reclaim process, or to attempt to use or modify objects in a library that is being reclaimed. Failure to do this could cause the reclaim to fail. If the reclaim fails, that library may have more catalog inconsistencies than existed before invoking this command. If this happens, it may be possible to recover by using the command again. If this does not correct the problem, a Reclaim Storage DBXREF (RCLSTG SELECT(*DBXREF)) will be required.

Top

Library (LIB)

Specifies which library should be recovered.

Note: This parameter is only valid when *FIX is specified for the **Option (OPTION)** parameter.

***ERR** Recovery should be attempted for all libraries known to have had errors in the cross-reference catalogs. The command will attempt to recover all cross-reference data for any library found having inconsistent data when RCLDBXREF was run previously with OPTION(*CHECK) specified.

name Specify the name of the library for which cross-reference data will be fixed. All cross-reference data for the specified library will be recovered, even if the library is not known to have inconsistent data in the catalogs for it. The library does not need to exist to be specified on this command.

Top

Examples

Example 1: Check for Cross-Reference Problems

```
RCLDBXREF OPTION(*CHECK)
```

This command checks the cross-reference catalogs for errors.

Example 2: Recover the Cross-Reference Catalogs

```
RCLDBXREF OPTION(*FIX) LIB(*ERR)
```


This command recovers the cross-reference information for all libraries known to be in error.

Example 3: Recover the Cross-Reference Catalog Data for One Library

```
RCLDBXREF OPTION(*FIX) LIB(ABCD)
```

This command recovers the cross-reference information for library ABCD.

[Top](#)

Error messages

***ESCAPE Messages**

CPF32AB

Database cross-reference information not recovered.

CPF32AC

Database cross-reference problem(s) exist.

CPF32A4

Internal failure in system cross-reference program.

[Top](#)

Reclaim DDM Conversations (RCLDDMCNV)

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

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

The Reclaim Distributed Data Management Conversations (RCLDDMCNV) command reclaims all Distributed Data Management (DDM) source system conversations that are not currently being used by this source job, even if the attribute value for the job is *KEEP. By using the RCLDDMCNV command, you do not have to close all open files or do any of the other functions performed by the Reclaim Resources (RCLRSC) command. This command applies only to the DDM conversations for the job on the source system in which the command is entered.

Although this command applies to *all* DDM conversations used by this job, using it does *not* mean that all of them are taken down. A conversation is taken down *only* if there are no active users for that conversation.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

RCLDDMCNV

This command checks all DDM conversations for the job in which the command is entered, determines if there are any users of each conversation, and reclaims each one not being used.

[Top](#)

Error messages

None

[Top](#)

Reclaim Document Lib Object (RCLDLO)

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

Parameters
 Examples
 Error messages

The Reclaim Document Library Object (RCLDLO) command allows you to reclaim:

- a document
- a folder
- a folder and all documents and folders directly or indirectly contained within it
- internal Document Library system objects.
- internal Document Library system objects, unfiled distribution documents, and all filed folders and documents on the system. Related document details are synchronized.
- internal document library system objects and all filed folders and documents in one auxiliary storage pool (ASP). Related document details are synchronized.

When using the RCLDLO command to reclaim all document library objects, the objects are synchronized with the search details index and the text search index. The synchronization with the text search index is complete when all the requests to update the text search index generated using the RCLDLO command have been processed.

Restrictions:

- To reclaim internal document library system objects or all Document Library objects, you must have all object (*ALLOBJ) or security administrator (*SECADM) special authority. These objects can be reclaimed only when no folders or documents are in use.
- You must have exclusive use of the document or folder while it is being reclaimed. You do not need authority to a document or folder to reclaim it. You also do not need to be enrolled in the System Distribution Directory.
- While using this command, you may encounter an error message indicating that internal objects are locked. Another user may be using Document Library functions, which prevents the RCLDLO command from running. Retry this command after other Document Library activity has ended.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value, *SYSOBJNAM, *INT, *ALL, *FLR, *DOCDTL	Required, Positional 1
FLR	In folder	Character value, *NONE	Optional, Positional 2
SYSOBJNAM	System object name	Name	Optional
SYSOBJATR	System object attributes	*NONE, *INTDOC, *DST	Optional
ASP	Auxiliary storage pool ID	1-32, *ANY	Optional
OUTFILE	File to receive output	Qualified object name	Optional
	Qualifier 1: File to receive output	Name, *NONE	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	

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

Top

Document library object (DLO)

Specifies the name of object to reclaim.

This is a required parameter.

***FLR** A folder and all folders and documents directly or indirectly within it are to be reclaimed.

*SYSOBJNAM

A system object name is used to identify the folder or document to be reclaimed. This parameter must be used when reclaiming a document that is **not** in a folder including internal and distribution documents.

***INT** Internal document library system objects are to be reclaimed.

Note: The internal document library system objects are used to manage the documents and folders on the system. RCLDLO DLO(*INT) is only necessary if the internal objects become damaged. If the internal objects are damaged, attempts to access documents and folders will result in the message CPF8A46 (Internal system objects are damaged), possibly followed by the message CPF9032 (Document interchange session not started),

*DOCCTL

Internal document library system objects and document details are to be reclaimed. DLO(*DOCCTL) synchronizes the relationships between all document library objects and their document details and will fix inconsistencies between them.

Note: The RCLDLO DLO(*DOCCTL) command can be a long-running function, performing a subset of the RCLDLO DLO(*ALL) processing necessary to guarantee consistency between internal system objects, document details, and DLOs.

***ALL** Internal document library system objects and all documents and folders (as specified on the ASP parameter) are to be reclaimed. DLO(*ALL) synchronizes the relationships between all document library objects and their document details and can be used to fix inconsistencies between them.

Note: The RCLDLO DLO(*ALL) command can be a long-running function, depending on the number of documents and folders on the system. If the RCLDLO command can be issued at the user's discretion, the user may wish to avoid the operation until the time required can be scheduled.

name Specify the name of the document or folder to be reclaimed.

Top

In folder (FLR)

Specifies the folder containing the folder or document to reclaim, or specifies the folder to reclaim along with all documents and folders directly or indirectly within it. A folder name is entered on this parameter only if a folder or document name or *FLR is entered on the **Document library object (DLO)** parameter.

*NONE

The folder or document to reclaim is not located in a folder.

name Specify the name of the folder that contains the document or folder to reclaim or specify the name of the folder to reclaim along with all folders and documents directly contained within it.

Top

System object name (SYSOBJNAM)

Specifies the system object name of the object to reclaim. A system object name must be entered on this parameter if *SYSOBJNAM is specified on the **Document library object (DLO)** parameter.

Top

System object attributes (SYSOBJATR)

Specifies the attributes of the object to reclaim. A value other than *NONE may be entered on this parameter only if *SYSOBJNAM is specified on the **Document library object (DLO)** parameter.

*NONE

No attributes are specified for the object.

*INTDOC

The object to reclaim is an internal document.

*DST The object to reclaim is a distribution document.

Top

Auxiliary storage pool ID (ASP)

Specifies the identifier (ID) of the auxiliary storage pool (ASP) of the document library object to be reclaimed. A value other than *ANY can be specified on this parameter only if *ALL or *DOCDDL is specified on the DLO parameter.

*ANY The objects to be reclaimed reside in any ASP. When *ALL is specified on the DLO parameter, all document library objects on the system are reclaimed.

1-32 Only document library objects that reside in the specified ASP are to be reclaimed. All document library objects in other ASPs are ignored. The value must designate an existing ASP that contains document library objects. ASP 1 is the system ASP.

Note: Unfiled distribution documents are classified as document library objects in the system ASP.

Top

File to receive output (OUTFILE)

Specifies the name of the database file to which special output is directed. If the output file does not exist, this command creates a database file in the specified library. If the file is created by this function, the descriptive text is "OUTFILE created by RCLDLO" and the authority for users without specific authority to the file is *EXCLUDE. A value other than *NONE can be specified on this parameter only if *ALL or *DOCDDL is specified on the DLO parameter.

The output directed to this file includes the names of any documents that are physically damaged (and therefore unusable) or documents or folders that were missing from the system (and for which the document details have been removed). This file is intended to provide the user with a record of what was lost (such as a user ASP) when recovering from hardware failure.

Qualifier 1: File to receive output

*NONE

No output is directed to a database file.

name Specify the qualified name of the database file that is to receive the output. This file can be reused when other RCLDLO commands are issued. Output is added to the file as specified on the OUTMBR parameter. The IBM-supplied database file, QSYS/QARCLDLO, cannot be specified.

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

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: Reclaiming a Folder

```
RCLDLO  DLO(FLR1)
```

This command reclaims folder FLR1.

Example 2: Reclaiming a Document Within a Folder

```
RCLDLO DLO(A) FLR(FLR2)
```

This command reclaims folder or document A in folder FLR2.

Example 3: Reclaiming a Folder and All Documents and Folders Within It

```
RCLDLO DLO(*FLR) FLR(FLR3)
```

This command reclaims folder FLR3 and all folders and documents directly or indirectly contained within it.

Example 4: Reclaiming an Internal Document

```
RCLDLO DLO(*SYSOBJNAM) SYSOBJNAM(AMBT133080)  
SYSOBJATR(*INTDOC)
```

This command reclaims the internal document specified by the system object name AMBT133080.

Example 5: Reclaiming a Distribution Document

```
RCLDLO DLO(*SYSOBJNAM) SYSOBJNAM(AMBT133082)  
SYSOBJATR(*DST)
```

This command reclaims the distribution document specified by the system object name AMBT133082.

Example 6: Reclaiming Document Library System Objects

```
RCLDLO DLO(*INT)
```

This command reclaims internal document library system objects.

Example 7: Reclaiming Document Library System Objects and Document Details

```
RCLDLO DLO(*DOCCTL)
```

This command reclaims internal document library system objects and document details for all folders and documents.

Example 8: Reclaiming Document Library System Objects and All Documents and Folders

```
RCLDLO DLO(*ALL)
```

This command reclaims internal document library system objects and all documents and folders and synchronizes their document details.

Example 9: Reclaiming Document Library System Objects and All Documents and Folders in an ASP

RCLDLO DLO(*ALL) ASP(2)

This command reclaims internal document library system objects and all documents and folders residing in ASP 2 and synchronizes their document details.

[Top](#)

Error messages

***ESCAPE Messages**

CPF8A44

Error occurred in document details.

CPF8AA4

Reclaim of &2 not successful.

CPF8A29

Reclaim of &2 partially successful.

[Top](#)

Reclaim Library (RCLLIB)

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

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

The Reclaim Library (RCLLIB) command rebuilds the internal objects of a library that contain the object descriptive information for all objects in the library and the library object itself.

This command rebuilds, where possible, internal objects of the library that were damaged or destroyed.

Restrictions:

1. You must have object existence (*OBJEXIST) and use (*USE) authorities for the library to be rebuilt. This is the same authority required to delete a library with the Delete Library (DLTLIB) command.
2. Only the internal objects of a library which contain the object descriptive information are rebuilt. No other objects in the library are validated or rebuilt.
3. This command can be used on any library other than QTEMP.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i>	Required, Positional 1

[Top](#)

Library (LIB)

Specifies the library to be rebuilt.

This is a required parameter.

name Specify the name of the library to be rebuilt.

[Top](#)

Examples

RCLLIB LIB(TESTLIB)

This command determines if the object descriptive information of library TESTLIB is damaged. The damaged parts of the library are rebuilt.

[Top](#)

Error messages

*ESCAPE Messages

CPF210A

Cannot reclaim library &1.

CPF210B

Attempt to reclaim library &1 failed.

CPF2127

User profile &2 damaged.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9830

Cannot assign library &1.

Top

Reclaim Object Links (RCLLNK)

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

Parameters
Examples
Error messages

The Reclaim Object Links (RCLLNK) command allows a single object or a group of objects to be reclaimed.

The RCLLNK command can be used to reclaim a directory tree where the specified directory, its contents, and the contents of all of its subdirectories are reclaimed. If SUBTREE(*ALL) is specified, the command will attempt to reclaim as many objects as possible within the subtree. A diagnostic message will be sent for each object that cannot be reclaimed. Additionally, an informational message will be sent if a specific problem is corrected, or a diagnostic message will be sent if a specific problem cannot be corrected. If all of the objects have been reclaimed, with all of the problems corrected, then a completion message will be sent. Otherwise, an escape message will be sent.

The RCLLNK command does the following:

- Corrects object user profile problems
- Corrects user-defined file system problems
- Corrects internal object problems
- Removes invalid object links
- Handles damaged objects as specified on the **Damaged object option (DMGOBJOPT)** parameter
- Creates missing system objects

A full Reclaim Storage (RCLSTG) fixes the above problems, as well as others such as lost objects or problems which require the system to be in a restricted state. Unlike RCLSTG, the system does not have to be in a restricted state to run RCLLNK.

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- You must have all object (*ALLOBJ) special authority to run this command.
- Only objects found in the "root" (/), QOpenSys, or mounted user-defined file systems are eligible to be reclaimed. All other objects are ignored during a subtree reclaim.
- An independent auxiliary storage pool (ASP) must have a status of "Available" in order for objects residing on the independent ASP to be reclaimed.
- Storage freed objects are not reclaimed unless a storage freed object is specified on the **Object (OBJ)** parameter.
- Objects that are being saved or restored cannot be reclaimed.
- An object's last access, data change, and attribute change date and time values are not normally updated as a result of being reclaimed. However, if the object has a problem that is corrected, these values may be updated.
- Problems found in the contents of a damaged directory object are not corrected.
- The directory conversion function must not be active while RCLLNK runs. The Convert Directory (CVTDIR) command with OPTION(*CHECK) can be used to verify directory conversion is not active.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
SUBTREE	Directory subtree	*DIR , *NONE, *ALL	Optional, Positional 2
DMGBOBJOPT	Damaged object option	<i>Element list</i>	Optional
	Element 1: Usable objects	*KEEP , *DELETE	
	Element 2: Unusable objects	*DELETE , *KEEP	

Top

Object (OBJ)

Specifies the path name of the object to be reclaimed. The object must be in the "root" (/), QOpenSys, or a user-defined file system. The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern cannot be specified ('*' or '?'). If the path name is qualified, it must be enclosed in apostrophes.

RCLLNK will not follow symbolic links for the last component in the path name.

If the last component in the path name is a block special file (*BLKSF) then only the block special file object will be reclaimed, not the user-defined file system that it represents.

The last component name in the path name cannot be '.' (dot) or '..' (dot-dot).

The effective root directory must be the "root" (/). Refer to the i5/OS PASE chroot command in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

Directory subtree (SUBTREE)

Specifies whether or not to reclaim the objects within the subtree if the object specified by the **Object (OBJ)** parameter is a directory.

***DIR** The object specified by OBJ is reclaimed. If the object is a directory, its contents are reclaimed, but the contents of all of its subdirectories are not reclaimed.

***NONE**
Only the object specified by OBJ is reclaimed.

***ALL** The object specified by OBJ is reclaimed. If the object is a directory, its contents, as well as the contents of all of its subdirectories, are reclaimed.

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Damaged object option (DMGOBJOPT)

Specifies how damaged objects are handled that are considered **usable** or **unusable**. If no operations can be performed on a damaged object it is considered unusable. Otherwise, it is considered usable.

Deleting a usable damaged directory object will cause all its contents and the contents of all its subdirectories to become lost. These lost objects will need to be restored from media or can be recovered via a RCLSTG. The contents of an unusable damaged directory object are already lost.

The following considerations apply:

- Damaged file system root directory objects are not deleted
- Usable damaged objects that are in use are not deleted
- The contents of an unusable damaged directory object are not reclaimed
- The contents of a usable damaged directory object are reclaimed only if ***KEEP** is specified for usable objects

Element 1: Usable objects

***KEEP**

Usable damaged objects are not deleted.

***DELETE**

Usable damaged objects are deleted, if possible.

Element 2: Unusable objects

***DELETE**

Unusable damaged objects are deleted, if possible.

***KEEP**

Unusable damaged objects are not deleted.

Top

Examples

Example 1: Reclaim Object Links for a Directory

```
RCLLNK OBJ('/MYOBJ') SUBTREE(*DIR)
```

The object MYOBJ will be reclaimed. If MYOBJ is a directory, all of the objects this directory contains will be reclaimed because *DIR is specified for the SUBTREE parameter.

Example 2: Reclaim Object Links for an Object

```
RCLLNK OBJ('/MYOBJ') SUBTREE(*NONE)
```

Only the object MYOBJ will be reclaimed because *NONE is specified for the SUBTREE parameter.

Example 3: Reclaim Object Links for a Directory Subtree

```
RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL)
```

The object MYOBJ will be reclaimed. If MYOBJ is a directory, all of the objects this directory contains, as well as all of the objects contained in the subdirectories, will be reclaimed because *ALL is specified for the SUBTREE parameter.

Example 4: Reclaim Damaged Objects in a Directory Subtree

```
RCLLNK OBJ('/MYDIR') SUBTREE(*ALL) DMGOBJOPT(*KEEP *DELETE)
```

If the MYDIR directory is not damaged, keep all usable damaged objects and delete all unusable damaged objects found in the directory subtree of MYDIR. All other problems found are corrected, if necessary and possible.

Example 5: Search for All Damaged Objects in a Directory Subtree

```
RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL) DMGOBJOPT(*KEEP *KEEP)
```

Check MYOBJ for damage. If MYOBJ is a directory, search for all damaged objects found in the directory subtree of MYOBJ. All other problems found are corrected, if necessary and possible.

Example 6: Delete All Damaged Objects in a Directory Subtree

```
RCLLNK OBJ('/MYOBJ') SUBTREE(*ALL) DMGOBJOPT(*DELETE *DELETE)
```

Delete MYOBJ if it is damaged. If MYOBJ is a directory that is not damaged, delete all damaged objects found in the directory subtree of MYOBJ. All other problems found are corrected, if necessary and possible.

Top

Error messages

*ESCAPE Messages

CPF8206

Directory conversion cannot be active during RCLSTG or RCLLNK.

CPFA085

Home directory not found for user &1.

CPFA089

Pattern not allowed in path name.

CPFA0A2

Information passed to this operation was not valid.

CPFA0A7

Path name too long.

CPFA0A9

Object not found. Object is &1.

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0DF

Error reclaiming objects in directories.

CPFA0F1

&1 of &2 object links reclaimed with &3 of &4 problems corrected.

Top

Reclaim Objects by Owner (RCLOBJOWN)

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

Parameters
Examples
Error messages

The Reclaim Objects by Owner (RCLOBJOWN) command checks the objects owned by a user profile to make sure those objects which must be in a library are actually in one. If any of the owned objects are not in a library, they are inserted in one. If the object can only be in a specific system library, the object is inserted in that library. If the object can exist in any user library and is reclaimed from the system auxiliary storage pool (ASP 1) or a basic user ASP (ASPs 2-32), the object is inserted in library QRCL. If the object is reclaimed from a primary or secondary ASP, the object is inserted in library QRCLnnnnn where nnnnn is the number assigned by the operating system to the primary ASP.

When a user profile is deleted and the profile owns objects which are not in a library, message CPC2216 is sent after the user profile is deleted stating that the ownership for the objects was changed and that a reclaim operation, either RCLOBJOWN or RCLSTG, is necessary to place the objects in a library. Since RCLSTG can be a long-running command, you can try running RCLOBJOWN which is much faster because it only checks the objects owned by the specified user profile. If RCLOBJOWN is unable to reclaim the objects, you will need to run the RCLSTG command to reclaim the objects.

When a user profile is deleted because it is damaged, the objects owned by the user profile are left with no owner. In this case, you need to run the RCLSTG command to assign those objects to the QDFTOWN user profile. After running RCLSTG, you can use the Change Object Owner (CHGOBJOWN) command to change the ownership of objects assigned to QDFTOWN by the RCLSTG command.

If RCLOBJOWN is run against a user profile which has objects residing in an Independent Auxiliary Storage Pool (IASP), this IASP must be in AVAILABLE status, in order that the objects are reclaimed.

Note: The RCLOBJOWN command performs only a subset of the functions performed by the Reclaim Storage (RCLSTG) command. Refer to the RCLSTG command information for other reclaim commands that can be used to perform specific parts of the RCLSTG function.

Restrictions:

- This command is shipped with public *EXCLUDE authority.
- All subsystems must be inactive before the RCLOBJOWN command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS) command with *ALL specified on the SBS parameter can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.
- This command will not reclaim document (*DOC) objects or the folder (*FLR) objects where documents reside. It also will not reclaim directory (*DIR) objects, or the following types of objects which reside in directories:
 - stream file (*STMF)
 - symbolic link (*SYMLNK)
 - FIFO queue (*FIFO)
 - block special file (*BLKSF)
 - socket (*SOCKET)

Top

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	<i>Simple name</i>	Required, Positional 1

Top

User profile (USRPRF)

Specifies the user profile for which the reclaim will be performed.

simple-name

Specify the name of the user profile to be processed.

Top

Examples

Example 1: Reclaim Objects for a User Profile

```
RCLOBJOWN USRPRF(MYPROFILE)
```

This command will check all objects owned by MYPROFILE. If there are objects that are not in a library, they will be placed in a system library. A message is sent for each object inserted into a system library by the reclaim operation.

Top

Error messages

*ESCAPE Messages

CPF8204

Commitment control cannot be active during RCLSTG or RCLOBJOWN.

CPF8208

System not in proper state to reclaim objects by owner.

CPF8213

Object &1 cannot be inserted in &2 library.

CPF8215

User profile &1 damaged.

CPF8220

Library &1 damaged. Reclaim Objects by Owner (RCLOBJOWN) command ended.

CPF8232

Commitment definitions prevent RCLSTG or RCLOBJOWN.

CPF8282

User profile &1 not found.

Top

Reclaim Optical (RCLOPT)

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

Parameters
Examples
Error messages

The Reclaim Optical (RCLOPT) command rebuilds the optical index database files and/or the internal library indexes. These files and indexes are used to maintain volume and directory information. Following are some of the reasons the files and indexes may need to be rebuilt: after CISC to RISC migration, after optical media is physical moved or removed by hand, after a hardware failure while media was being moved, or when a database or index has been damaged or destroyed. This command requires exclusive use of the directly-attached optical device.

Note: This command may take an extended period of time to complete - possibly more than 24 hours. The time to completion is influenced by several factors including: number of optical libraries, number of media in each library, capacity of each media, number of files and directories on each media and the options chosen on the command.

Restriction: You must have *USE authority to use this command. It is shipped with *EXCLUDE public authority.

Top

Parameters

Keyword	Description	Choices	Notes
MLB	Optical media library	<i>Name</i> , *ALL	Required, Positional 1
OPTION	Option	* <u>SYNC</u> , *UPDATE, *RESET	Optional, Positional 2
VOL	Volume identifier	<i>Character value</i> , * <u>ALL</u>	Optional
DIR	Rebuild directory index	* <u>NO</u> , *YES	Optional

Top

Optical media library (MLB)

Specifies the directly-attached optical device for which the optical indexes are rebuilt.

*ALL The optical indexes for all directly-attached optical devices are rebuilt.

optical-device-name

Specify the name of the directly-attached optical device for which the optical indexes are rebuilt.

Top

Option (OPTION)

Specifies which type of reclaim operation is performed.

*SYNC

The optical volume index is synchronized with the internal device index. If a volume is not in the optical volume index, both the volume and the directory index entries are created for the volume. This option will usually complete in the shortest amount of time and, in most cases, should be attempted first before trying the *UPDATE or *RESET option.

*UPDATE

The optical volume index is rebuilt and the optical directory index is optionally rebuilt with information read from the optical cartridge. In most cases this option will be used to rebuild the indexes for a single volume or optical cartridge.

*RESET

The internal device index and the optical volume index are rebuilt. The optical directory index is optionally rebuilt with information read from the optical cartridge. Selecting *RESET requires every volume to be mounted into a drive at least once. The optical directory index is built if the Rebuild directory index parameter is set to *YES. Specifying DIR(*YES) will require every volume to be mounted a second time. In most cases *RESET should be used only after *SYNC and/or *UPDATE have already been attempted.

Top

Volume identifier (VOL)

Specifies which volumes are used during the reclaim operation when OPTION(*UPDATE) is specified.

*ALL All volumes in the optical device are used.

volume-identifier

Specify the volume identifier of a specific volume to use during the reclaim operation.

Top

Rebuild directory index (DIR)

Specifies whether the optical directory index is rebuilt for each volume processed during Reclaim Optical processing. This parameter only applies to options *UPDATE and *RESET. When *SYNC is specified, DIR(*NO) will be used. In addition, this option does not apply to volumes which have a media format type of *UDF since directory index information is not maintained for *UDF volumes. When *UDF volumes are processed, DIR(*NO) will be used.

*NO The optical directory index is not rebuilt for each volume processed.

*YES The optical directory index is rebuilt for each volume processed. Only applies to *UPDATE and *RESET for non-UDF volumes.

Note: Choosing the value *NO may result in less time being required to complete the Reclaim Optical request.

Top

Examples

```
RCLOPT  MLB(OPT01)  OPTION(*UPDATE)  VOL(VOL01)  DIR(*YES)
```

This command re-creates both the optical volume index and the optical directory index for the optical volume VOL01 in the optical media library OPT01.

Error messages

*ESCAPE Messages

OPT0125

Command &1 completed with errors, more information in job log.

OPT1320

Optical volume &1 in use.

OPT1325

Optical volume format not recognized.

OPT1330

Optical volume not found or not useable.

OPT1331

Optical volume &1 not found.

OPT1346

Operation not allowed to volume located in a remote optical device.

OPT1360

Media directory corrupted on optical volume &1.

OPT1460

Optical volume &1 is not in an optical device.

OPT1530

&1 does not represent a valid optical device.

OPT1555

Optical device &1 in use.

OPT1605

Media or device error occurred.

OPT1640

Error occurred reading files or directories.

OPT1652

Device &1 is not an optical media library.

OPT1790

Operation not allowed or conflicts with another request.

OPT1805

Error accessing optical volume index file.

OPT1810

Error accessing optical directory index file.

OPT1815

Internal program error occurred.

OPT1820

Internal error occurred on optical device &1.

OPT1821

Error occurred on optical device &1.

OPT1825

Optical indexes are incorrect for optical device &1.

- OPT1860**
Request to optical device &1 failed.
- OPT1861**
No device description configured for resource &1.
- OPT1862**
No active device description for resource &1.
- OPT1863**
Optical libraries need to be reclaimed.
- OPT1872**
Optical request timed out or was cancelled.
- OPT2040**
Error accessing backup control file.
- OPT2155**
Reclaim Optical failed for optical device &1.
- OPT2165**
Reclaim Optical failed for volume &1.
- OPT2187**
Optical index inconsistency, reclaim optical required.
- OPT2188**
Optical index files are damaged.
- OPT2190**
Error occurred during reclaim optical processing while accessing volume &1.
- OPT2191**
Error occurred while reading volume directory for volume &1.
- OPT2301**
Internal system object in use.
- OPT7740**
User not authorized to object &2 in library &3 type &4.

Top

Reclaim Resources (RCLRSC)

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

Parameters
Examples
Error messages

The Reclaim Resources (RCLRSC) command is intended for use by the controlling program in an application to free resources that were allocated by programs that have finished running and returned control to the controlling program. The resources used by these programs can then be used by other programs running on the system. Based on the reference level specified by the LVL parameter, this command reclaims resources that are scoped to the default activation group. This command does not reclaim resources that are scoped to the job or resources that are scoped to any activation group other than the default activation group. The resources that are reclaimed by this command are:

- static storage
For an original program model (OPM) program, its static storage will be reclaimed. For a user state Integrated Language Environment (ILE) program, its static storage will be marked so that it will be re-initialized on the next call. Static storage is not affected for a service program, a system state ILE program, or any ILE program running in an activation group other than the default activation group.
- open files of object type *FILE
- user interface manager (UIM) application resources
- Common Programming Interface (CPI) Communications conversations
- hierarchical file systems (HFS) resources
- user-defined communications sessions.

In addition, after distributed data management (DDM) files are closed based on the criteria described above, all DDM source system conversations in this job that are not currently in use will be ended, regardless of the activation group or reference level from which they were started.

The RCLRSC command is **not** needed to reclaim the files and static storage of most programs, such as CL programs that end (return) normally, RPG programs that have the last record (LR) indicator set on, and COBOL programs. The RCLRSC command should **not** be used if it might be processed while any COBOL program is still active in the application.

For more information on the appropriate use of this command, see the documentation associated with the programming language(s) used in the application.

Restrictions:

1. This command is not threadsafe. However, it can be run in the primary thread of a multi-threaded job.
2. Do not specify LVL(*CALLER) on this command if it is used in a CL program that also uses the Send File (SNDF), Receive File (RCVF), Send/Receive File (SNDRCVF) commands. Specifying RCLRSC LVL(*CALLER) in such a program causes unpredictable results when the SNDF, RCVF, or SNDRCVF commands are used after the program runs.
3. Do not specify LVL(*CALLER) on this command if it is issued from a command line, as this will result in unpredictable results.

Top

Parameters

Keyword	Description	Choices	Notes
LVL	Program level	*, *CALLER _	Optional, Positional 1
OPTION	Close option	*NORMAL, *ABNORMAL	Optional

Top

Program level (LVL)

Specifies the reference level at which resources are reclaimed.

*
_ The reference level is the program or procedure that contains this RCLRSC command. The resources are reclaimed for programs or procedures that have finished running and returned control to this program.

*CALLER

The reference level is the program or procedure that called the program or procedure containing this RCLRSC command. This value allows controlling programs or procedures written in a high-level language to call a CL program to reclaim resources to the level of the controlling program or procedure. The effect is the same as if the command were issued from the controlling program or procedure.

Note: Using the *CALLER value can cause unexpected results when running the RCLRSC command from a command line or from within a program that works with open files.

Top

Close option (OPTION)

Specifies whether a normal or abnormal close notification is sent to the attached host system when mixed, communications, binary synchronous (BSC), and intersystem communications function (ICF) files are closed. This parameter is ignored for all other files and objects.

*NORMAL

The attached host system is given a normal close notification when mixed, communications, BSC, and ICF files are closed.

*ABNORMAL

The attached host system is given an abnormal close notification when mixed, communications, BSC, and ICF files are closed. Use this when the controlling program detects error conditions that should be communicated to the host systems (the error condition need not be file-related).

Top

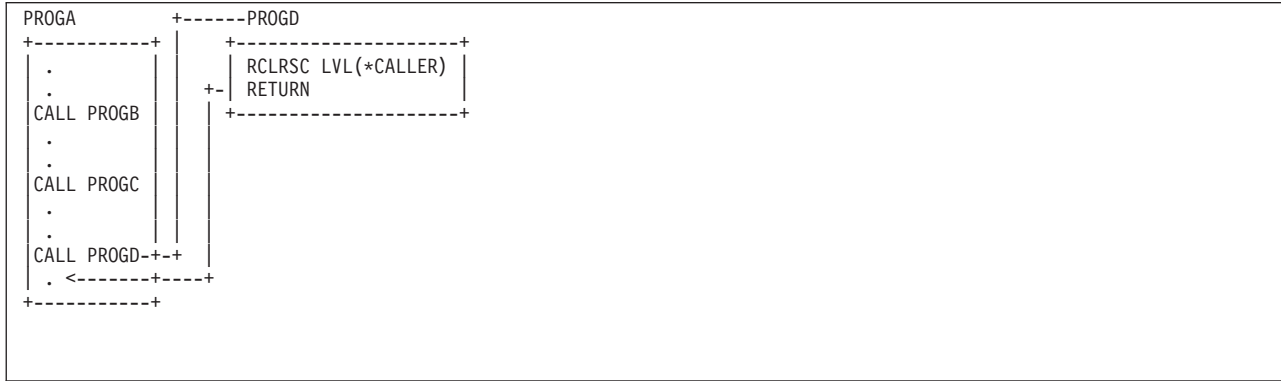
Examples

Example 1 (OPM)

```
PROGA
:
CALL PROGB
RCLRSC
:
CALL PROGC
RCLRSC
:
```

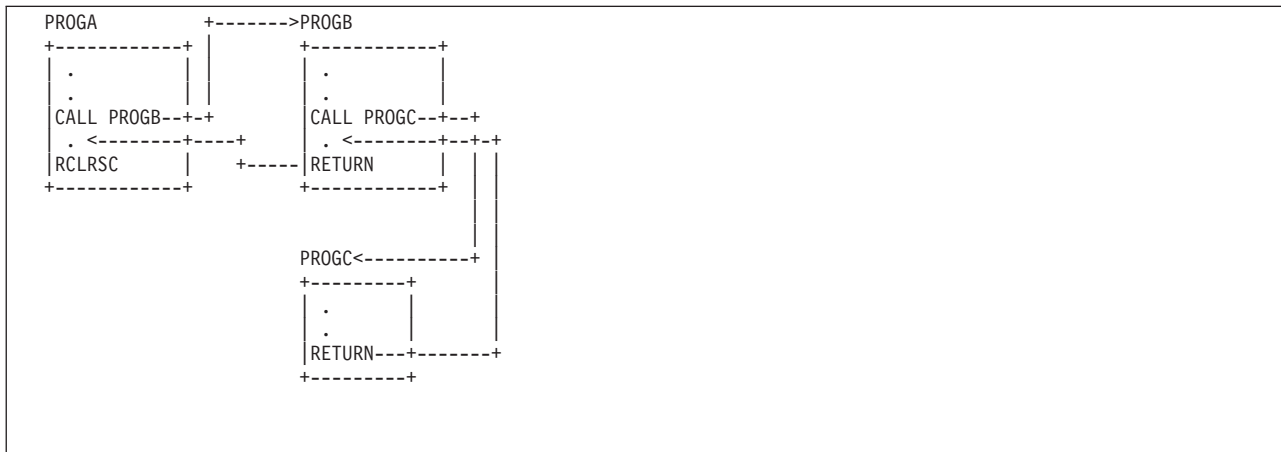
In this example, PROGA is a controlling program in an application. PROGA calls other programs, which return control to PROGA when they have finished running. Because control is returned to the next sequential instruction, the RCLRSC command is issued following each CALL command to free the static storage that was used by the called program, and to close the files that were left open.

Example 2 (OPM)



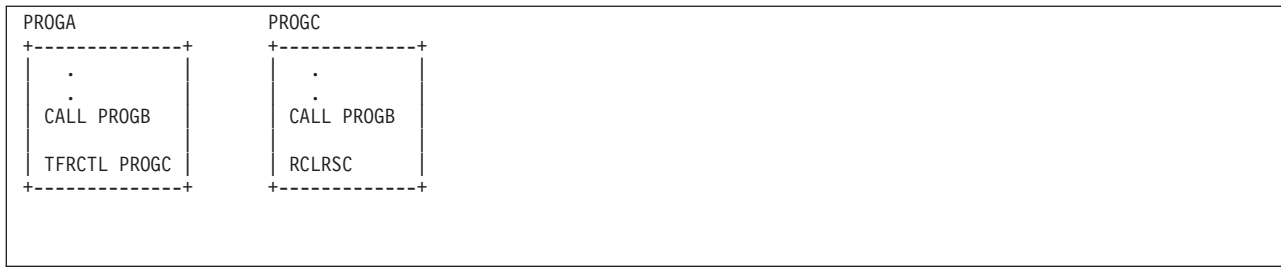
In this example, PROGA is a controlling program that is written in a high-level language. The RCLRSC command cannot be issued from the high-level language program so PROGD, a CL program, is called to issue the command. When the RCLRSC command is issued in PROGD, the static storage used by PROGB and PROGC is freed; files that were left open are closed.

Example 3 (OPM)



In this example, PROGA is a controlling program. When the RCLRSC command is issued, the static storage used by PROGB and PROGC is freed; files that were left open are closed.

Example 4 (OPM)

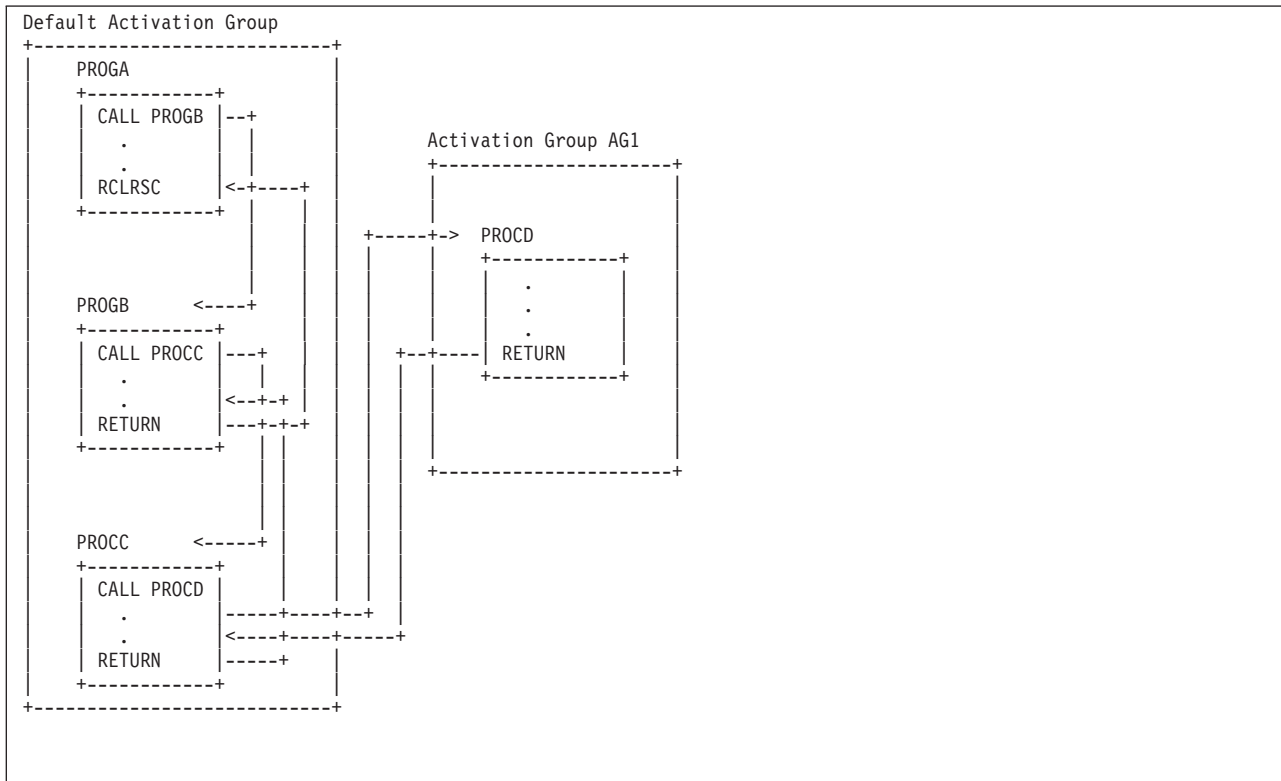


In this example, PROGA calls PROGB and, after returning from PROGB, PROGA transfers to program PROGC. Because PROGB has already been called, static storage exists, and the call to PROGB from PROGC does not cause any new allocation for static storage; PROGC cannot reclaim the static storage used by PROGB. If PROGB opened files when it was called by PROGA, these files would remain open; if PROGB opened files when it was called by PROGC, these files are closed.

Example 5 (OPM)

In this example, PROGA calls PROGB, which in turn calls PROGC. PROGC opens a file. Control returns to PROGA. PROGA calls PROGB a second time, and PROGB invokes RCLRSC. Since RCLRSC is based upon the position in the stack of the current invocation of a program, the file opened by PROGC is not affected. PROGC was invoked earlier than the current PROGB. In order for RCLRSC to close the file, the RCLRSC command will have to be invoked by PROGA.

Example 6 (OPM & ILE)



This example shows how ILE procedures and activation groups are affected by the RCLRSC command.

In this example, PROGA is a program running in the default activation group. PROGA calls program PROGB which runs in the default activation group. PROGB calls ILE procedure PROCC which runs in

the default activation group. PROCC calls ILE procedure PROCD which causes activation group AG1 to be created. PROCD returns to PROCC. PROCC returns to PROGB. PROGB returns to PROGA, which then calls the RCLRSC command.

PROGA calls the RCLRSC command. Any resources in use by PROGA are still open, since PROGA is still in use. Any resources by program PROGB or procedure PROCC are reclaimed, since the program and procedure ran in the default activation group and are no longer active. Any resources opened by procedure PROCD are left alone, since procedure PROCD ran in activation group AG1 and only the default activation group is affected by the RCLRSC command.

Any other use of the RCLRSC command can result in files remaining open and storage remaining allocated.

[Top](#)

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

[Top](#)

Reclaim Spool Storage (RCLSPLSTG)

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

Parameters
Examples
Error messages

The Reclaim Spool Storage (RCLSPLSTG) command reclaims unused storage for spooled files that have not been used for more than the number of days specified by the user. Spooled files are stored with database file members on the system. When a spooled file is deleted, the member is emptied but not deleted. Therefore, the member can be reused for the next spool file created. Reusing empty members improves the performance time when creating new spooled files. The RCLSPLSTG command deletes unused and empty database members in the specified ASP group. The QRCLSPLSTG system value can be set to automatically delete unused and empty database members in primary or secondary ASPs. This command uses synchronous processing. More information about synchronous processing is in the Recovering your system book, SC41-5304.

Restriction: If *ALL or a specific **ASP group** is specified for the ASPGRP parameter, the user must have use (*USE) authority to all ASP device descriptions in the ASP group and the status of the ASP group must be AVAILABLE.

Top

Parameters

Keyword	Description	Choices	Notes
DAYS	Days	1-366, *NONE	Required, Positional 1
ASPGRP	ASP group	Name, <u>*SYSBAS</u> , *ALL, *, *CURASPGRP	Optional

Top

Days (DAYS)

Specifies an interval in days. If existing spool storage has not been used to create new spooled files in the number of days specified, it will be deleted.

This is a required parameter.

*NONE

No interval is used. All unused spool storage is deleted. No storage will remain for creating new spooled files. This will lengthen the time it takes to create a new spooled file.

Note: Using this value can have adverse effects on system performance. More information is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

1-366 Specify the number of days. The specified number of days is measured to the second.

Top

ASP group (ASPGRP)

Specifies the auxiliary storage pool (ASP) group that contains the unused database members to delete.

*SYSBAS

Unused database members which are found in the system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) are deleted.

***ALL** Unused database members which are found in the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32) and all primary and secondary ASPs are deleted.

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

***CURASPGRP**

Unused database members which are found in the primary and secondary ASPs in the thread's ASP group are deleted. If no ASP group is associated with the thread, an error will be issued.

name Specify the name of an ASP group. Unused database members which are found in the primary and secondary ASPs of the specified ASP group are deleted.

Top

Examples

Example 1: Delete Unused Database Members in the System ASP (ASP 1) and All Defined Basic User ASPs (ASPs 2-32)

```
RCLSPLSTG  DAYS(30)
```

This command reclaims all unused storage for spooled files in the system auxiliary storage pool (ASP 1) and all defined basic user ASPs (ASPs 2-32) that have remained unused for more than 30 days. When storage has been unused for 1 second over 30 days it is reclaimed because a date and time stamp is placed on the storage area.

Example 2: Delete Unused Database Members in the Current User's ASP Group

```
RCLSPLSTG  DAYS(*NONE)  ASPGRP(*CURASPGRP)
```

This command reclaims all unused storage for spooled files in the current user's ASP group.

Example 3: Deleting All Unused Database Members

```
RCLSPLSTG  DAYS(*NONE)  ASPGRP(*ALL)
```

This command deletes all unused database members on the system.

Example 4: Deleting All Expired Spooled Files in the System ASP (ASP 1) and All Defined Basic User ASPs (ASPs 2-32)

```
RCLSPLSTG  DAYS(*NONE)  ASPGRP(*)
```


This command reclaims all unused storage for spooled files in the system auxiliary storage pool (ASP 1), all defined basic user ASPs (ASPs 2-32), and if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

[Top](#)

Error messages

***ESCAPE Messages**

CPF9825

Not authorized to device &1.

CPFB8E9

ASP group &1 not set for thread &2.

[Top](#)

Reclaim Storage (RCLSTG)

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

Parameters
Examples
Error messages

The Reclaim Storage (RCLSTG) command corrects, where possible, objects that were incompletely updated (such as database files, libraries, device descriptions, directories and stream files) and user profiles containing incorrectly recorded object ownership information. Any unusable objects or fragments are deleted.

This command reclaims all objects secured by an authorization list that is damaged or destroyed and assigns the objects to the authorization list QRCLAUTL.

Because the amount of time required to run this command varies with the number of objects in auxiliary storage, the system will display a panel to the work station where the command was specified, showing the progress of the command's execution. The 'Time Remaining' column will show blanks for a RCLSTG step if no statistics have been collected yet that would enable the RCLSTG command to estimate the total time required for that step.

You can specify *YES for the ESTIMATE parameter to get an estimate of how long the RCLSTG command will take to run. When ESTIMATE(*YES) is specified, messages that show the estimated amount of time are sent to the job log, and no reclaim function is performed.

The RCLSTG command can also be used to reclaim storage when, during an IPL, not enough storage is available to make the system fully operational. In that case, the system operator can specify the command immediately after receiving the message about insufficient storage.

If very little additional auxiliary storage is available, the system overhead required to run the RCLSTG command may need more than the remaining storage; in that case, the RCLSTG command fails.

Note: The RCLSTG command can be a long-running function, depending on the number and type of objects in the system, and the amount of damage to them. Because RCLSTG touches each object multiple times, having enough memory can significantly reduce the time required to run RCLSTG. Conversely, having too little memory can lead to storage thrashing which can significantly increase the time required to run RCLSTG. If database file objects are damaged, the keyed access paths may need to be rebuilt; that operation takes a substantial amount of time. If the RCLSTG command can be run at the user's discretion, the user may want to avoid the operation until the required time can be scheduled.

You can select to run just the directory recovery portion of RCLSTG by specifying SELECT(*DIR). This will reclaim only directories and objects related to the integrated file system.

There are several reclaim commands that perform a subset of the RCLSTG's functions. These commands are: Reclaim Object Links (RCLLNK), Reclaim Database Cross-Reference (RCLDBXREF), Reclaim Objects by Owner (RCLOBJOWN), Reclaim Document Lib Object (RCLDLO), Reclaim Library (RCLLIB) and Reclaim Spool Storage (RCLSPLSTG). Refer to those commands for details on the functions that they perform.

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.
2. If option *SYSBAS is specified on the ASPDEV parameter, all subsystems must be inactive before the RCLSTG command can be specified. The End System (ENDSYS) or End Subsystem (ENDSBS)

command with *ALL specified on the SBS parameter can be used to make the subsystems inactive. You must have job control (*JOBCTL) authority to use the ENDSYS or the ENDSBS command.

3. Only permanent objects in auxiliary storage are reclaimed; temporary objects are reclaimed by running a system initial program load (IPL).
4. Before running the RCLSTG command after an IPL, you may need to wait several minutes for the IPL to complete. Use the Work with Active Jobs (WRKACTJOB) command to verify that no jobs are running.
5. If option *SYSBAS is specified for the ASPDEV parameter, this job must be in the controlling subsystem and must be the only job active in the system.
6. Network server descriptions (NWSD) must be varied-off in order to run RCLSTG.
7. When option *SYSBAS is specified for the ASPDEV parameter, all the auxiliary storage pool (ASP) devices configured in the system must be in the VARY OFF status.
8. If an auxiliary storage pool (ASP) device or an ASP group is specified on the ASPDEV parameter, it cannot have active users or be in use by a different job. The ASP device or group has to be in the AVAILABLE or FAILED status to allow the RCLSTG command to run.
9. The directory conversion function must not be active while RCLSTG runs. The Convert Directory (CVTDIR) command with OPTION(*CHECK) can be used to verify directory conversion is not active. Otherwise, option OMIT(*DIR) should be specified to omit the directory recovery portion of the reclaim function.
10. In order to provide an estimate of the amount of time that a future RCLSTG will require, statistics are collected when running the RCLSTG ESTIMATE(*NO) command. If you specify ESTIMATE(*YES) and these statistics are not available, escape message CPF8281 is issued.

Top

Parameters

Keyword	Description	Choices	Notes
ESTIMATE	Estimate time required	*NO, *YES	Optional
SELECT	Select	*ALL, *DBXREF, *DIR	Optional
OMIT	Omit	*NONE, *DBXREF, *DIR	Optional
ASPDEV	ASP device or group	Name, *SYSBAS	Optional

Top

Estimate time required (ESTIMATE)

Specifies whether to calculate an estimate for the amount of time that RCLSTG command will take to run. The estimate is calculated by using statistics collected during previous RCLSTG operations and the values specified for the other RCLSTG parameters.

- *NO** The estimate function is not performed. The reclaim function specified by the other parameters is performed.
- *YES** The estimate function is performed for the type of reclaim specified by the other parameters. Nothing is reclaimed. If SELECT(*DBXREF) is specified, no estimate can be provided. Instead, the time it took to run the last RCLSTG SELECT(*DBXREF) is provided, if available. Messages are displayed showing the results of the estimate.

Top

Select (SELECT)

Specifies whether to perform all reclaim functions or only one specific reclaim function.

***ALL** All reclaim functions are performed including, but not limited to, database cross-reference table reclaim and directory recovery functions.

***DBXREF** Only the database cross-reference table reclaim function is performed.

***DIR** Only the directory recovery portion of the reclaim function is performed.

Top

Omit (OMIT)

Specifies the reclaim functions to be omitted from the reclaim operation.

***NONE** No reclaim functions are omitted.

***DBXREF** The database cross-reference table reclaim function is omitted.

***DIR** The directory recovery portion of the reclaim function is omitted.

Top

ASP device or group (ASPDEV)

Specifies the auxiliary storage pool (ASP) to be reclaimed.

***SYSBAS** The system ASP and all basic ASPs are reclaimed. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers of 2 through 32.

auxiliary-storage-pool-device-name

The specified ASP device is reclaimed. ASP devices have ASP numbers greater than 32. Reclaim storage for an ASP device can be run without being in restricted state. The ASP device must be in the AVAILABLE or FAILED status to reclaim it. You can submit multiple jobs, each performing RCLSTG on a different ASP device, to reclaim multiple ASP devices in parallel.

auxiliary-storage-pool-group-name

The primary ASP and the secondary ASPs within the ASP group are reclaimed. The name of any ASP within the ASP group is accepted. Primary and secondary ASPs have ASP numbers greater than 32. Reclaim storage for an ASP group can be run without being in restricted state. The ASP device must be in the AVAILABLE or FAILED status to reclaim it. You can submit multiple jobs, each performing RCLSTG on a different ASP group, to reclaim multiple ASP groups in parallel.

Top

Examples

Example 1: Reclaim Storage of the System ASP and All Basic ASPs

RCLSTG

This command, specified interactively, locates all system objects. Objects without owners are given default owners, and those that are lost from their specified libraries are inserted into the QRCL library or the default library, or are deleted.

Objects that are lost from their specified directories are inserted into the '/QReclaim' directory (if the object was originally located in the root file system) or the '/QOpenSys/QReclaim' directory (if the object was originally located in the QOpenSys file system).

Lost objects that are deleted are certain user objects and certain system objects that are damaged and not usable.

The QRCL library, which is created (when needed) by the RCLSTG command, is a permanent library.

The '/QReclaim' and '/QOpenSys/QReclaim' directories, which are created (when needed) by the RCLSTG command, are permanent directories; but if they contain no objects at the end of the operation because they were all reclaimed, the directories are deleted.

Example 2: Reclaim Storage to Reclaim the Database Cross-reference Table

```
RCLSTG  SELECT(*DBXREF)
```

This command reclaims the database cross-reference table.

Example 3: Reclaim Storage of the Entire System that Omits the Reclaim of the Database Cross-reference Table

```
RCLSTG  OMIT(*DBXREF)
```

This command performs all reclaim storage functions but omits the reclaim of the database cross-reference table.

Example 4: Reclaim Storage that Omits the Reclaim of the Directories

```
RCLSTG  OMIT(*DIR)
```

This command performs all reclaim storage functions but omits the reclaim of the directories.

Example 5: Reclaim Storage of an ASP Device

```
RCLSTG  ASPDEV(MYASPDEV)
```

This command reclaims storage for the ASP device name MYASPDEV.

Example 6: Reclaim Storage to Reclaim the Directory Portion of the System ASP and All Basic ASPs

```
RCLSTG  SELECT(*DIR)
```

Objects that are lost from their specified directories are inserted into the '/QReclaim' directory (if the object was originally located in the root file system) or the '/QOpenSys/QReclaim' directory (if the object was originally located in the QOpenSys file system).

Example 7: Reclaim Storage to Reclaim the Directories of an ASP Device

```
RCLSTG  SELECT(*DIR)  ASPDEV(MYASPDEV)
```

This command reclaims directories for the ASP device name MYASPDEV.

Example 8: Estimate the Time that the Next Full RCLSTG Will Take to Run

```
RCLSTG  ESTIMATE(*YES)
```

This command estimates the time that the next full RCLSTG will take to run for each one of the major steps:

- Reading objects from disk.
- File ID table recovery.
- Directory recovery.
- Object description verification.

Messages are sent to the job log giving the estimated amount of time for each major RCLSTG step.

Example 9: Estimate the Time that the Next RCLSTG of an ASP Device Will Take to Run

```
RCLSTG  ESTIMATE(*YES)  ASPDEV(MYASPDEV)
```

This command estimates the time that the next RCLSTG of ASP device name MYASPDEV will take. Messages are sent to the job log giving the estimated amount of time for each major RCLSTG step.

Top

Error messages

*ESCAPE Messages

CPF180B

Function &1 not allowed.

CPF180C

Function &1 not allowed.

CPF2119

Library &1 locked.

CPF2120

Library &1 locked.

CPF2126

&1 libraries not validated.

CPF2127

User profile &2 damaged.

CPF5729

Not able to allocate object &1.

CPF8201

User profile &1 does not exist or is damaged.

CPF8204
Commitment control cannot be active during RCLSTG or RCLOBJOWN.

CPF8205
Library &1 does not exist or is damaged.

CPF8206
Directory conversion cannot be active during RCLSTG or RCLLNK.

CPF8209
System not in proper state to reclaim storage.

CPF8210
Reclaim storage failed for auxiliary storage pool device or group &2.

CPF8211
Library &1 damaged. RCLSTG command ended.

CPF8212
SELECT(*DBXREF) or OMIT(*DBXREF) not allowed when reclaiming a UDFS ASP.

CPF8214
Estimate option cannot be performed.

CPF8216
SELECT(*DBXREF) or OMIT(*DBXREF) not allowed for a UDFS ASP.

CPF8224
Duplicate object found while moving or renaming member.

CPF8232
Commitment definitions prevent RCLSTG or RCLOBJOWN.

CPF8251
RCLSTG command ended. Library &1 damaged.

CPF8252
Error occurred during rebuild of damaged library &1.

CPF8281
Unable to estimate time required to run RCLSTG.

CPF9814
Device &1 not found.

CPFA473
Network server &1 must be varied off.

CPFB8ED
Device description &1 not correct for operation.

Top

Reclaim Temporary Storage (RCLTMPSTG)

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

Parameters
Examples
Error messages

The Reclaim Temporary Storage (RCLTMPSTG) command allows you to reclaim storage used by temporarily decompressed copies of panel groups, menus, display files, and printer files, thereby freeing up system storage space.

- Compressed Objects are objects that consume less storage space than decompressed objects. When a compressed object is used or a compressed program is called, a decompressed version of the object automatically becomes available to the user.
- Decompressed Objects are objects that use the system storage space allocated to them and are in a final, ready-to-use state.
- Temporarily Decompressed Objects. are temporarily decompressed copies of compressed objects. The system allocates storage space for the decompressed objects, which is consumed by the temporary copies until the system or the user determines that the temporary storage space needs to be reclaimed.

Temporary storage is automatically reclaimed when:

- the RCLTMPSTG command is run
- the next IPL is run
- the object is used often enough to cause the system to permanently decompress it

When an object is permanently decompressed, the compressed version of the object is destroyed as well as any temporary forms of the object; however, compressed versions remain intact as long as the objects are temporarily decompressed.

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.
2. The user must have object management authority to the object specified and execute authority to the library.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, *ALL, *ALLUSR, *CURLIB, *LIBL, *USRLIBL	Optional, Positional 1
DAYS	Days unused	1-366, 7, *NONE	Optional, Positional 2

Top

Library (LIB)

Specifies the name of the library from which storage is reclaimed. All temporarily decompressed objects that are in the library you specify on this parameter and that have not been used for more than the number of days you specify on the **Days unused** prompt (DAYS parameter) are reclaimed.

The possible library values are:

***ALL** All libraries in the system, including QSYS, are searched.

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

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

***CURLIB**

Only the current library is searched. If no current library entry exists in the library list, QGPL is used.

***ALLUSR**

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

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

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVRxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVRxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

library-name

Specify the name of the library to be searched.

Top

Days unused (DAYS)

Specifies the number of days an object has not been used or changed. If a temporarily decompressed object has not been used or changed for more than the specified number of days, it is reclaimed. If it has been used or changed, it is left temporarily decompressed.

The possible values are:

7 Objects that have not been used or changed for more than seven days are reclaimed.

***NONE**

The object is reclaimed regardless of the number of days it has not been used or changed.

days-unused

Specify the number of days. Valid values range from 1 through 366.

Top

Examples

```
RCLTMPSTG LIB(QGPL)
```

This command reclaims the space consumed by all of the temporarily decompressed copies of objects in library QGPL that have not been used or changed in the last 7 days.

Top

Error messages

*ESCAPE Messages

CPF2110

Library &1 not found.

CPF2113

Cannot allocate library &1.

CPF2176

Library &1 damaged.

CPF3B07

&1KB storage reclaimed, &5 objects not processed.

CPF9838

User profile storage limit exceeded.

Top

Receive Distribution (RCVDST)

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

Parameters
 Examples
 Error messages

The Receive Distribution (RCVDST) command allows you to receive incoming distributions such as documents or files. The documents or files can be placed in folders or document objects or can be placed in an output file for processing.

Restriction: The requester of the command must be enrolled in the system distribution directory. If you request distribution information for another user, you must have been given permission to work on behalf of that user with the Grant User Permission (GRTUSRPMN) command. Personal distribution cannot be requested if the requester is working on behalf of another user.

Top

Parameters

Keyword	Description	Choices	Notes
DSTID	Distribution identifier	Character value	Required, Positional 1
USRID	User identifier	Single values: <u>*CURRENT</u> Other values: <u>Element list</u>	Optional
	Element 1: User ID	Character value	
	Element 2: Address	Character value	
DOC	Document	Character value, <u>*NONE</u>	Optional, Positional 2
FLR	In folder	Character value, <u>*NONE</u>	Optional, Positional 3
OUTFILE	File to receive output	Single values: <u>*NONE</u> Other values: <u>Qualified object name</u>	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , <u>*CURLIB</u>	
OUTMBR	Output member options	Element list	Optional
	Element 1: Member to receive output	Name, <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , <u>*ADD</u>	
OUTDTATYP	Type of data for output	Single values: <u>*DFT</u> , <u>*ALL</u> Other values (up to 24 repetitions): <u>*DSTINFO</u> , <u>*MSG</u> , <u>*DOCD</u> , <u>*CRTDATE</u> , <u>*EXPDATE</u> , <u>*DOCDATE</u> , <u>*FILDATE</u> , <u>*CHGDATE</u> , <u>*ACTDATE</u> , <u>*CMPDATE</u> , <u>*AUTHOR</u> , <u>*CPYLST</u> , <u>*DOCCLS</u> , <u>*FILCAB</u> , <u>*SUBJECT</u> , <u>*KWD</u> , <u>*REF</u> , <u>*STATUS</u> , <u>*PROJECT</u> , <u>*RPYDATE</u> , <u>*AUTUSR</u> , <u>*DSTEXPDATE</u> , <u>*IDP</u> , <u>*DOC</u>	Optional
ACKRCV	Acknowledge receipt	<u>*YES</u> , <u>*NO</u>	Optional
DSTIDEXN	Distribution ID extension	1-99, <u>*NONE</u>	Optional
KEEP	Keep in mail log	<u>*NO</u> , <u>*YES</u>	Optional

Keyword	Description	Choices	Notes
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

Distribution identifier (DSTID)

Specifies the unique distribution identifier of the distribution. The identifier is assigned to the distribution by the system that originated it. Only incoming distributions can be received. If the identifier represents an outgoing distribution, an error message is returned.

distribution-id

The distribution identifier is composed of the second part of the sender's user ID (padded on the right to 8 characters), the first part of the sender's user ID (padded on the right to 8 characters), and a 4-digit zoned sequence number with the leading zeros. For example, 'NEWYORK SMITH 0204'. This parameter is required when *DSTID is specified on the Information to be sent prompt (TYPE parameter).

This is a required parameter.

Top

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

*CURRENT

You are performing the request for yourself.

user-ID

Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

user-ID-address

Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Top

Document (DOC)

Specifies the name of the document object in which the distribution is placed when it is received. This document must not already exist and it is created as a private document. The document is either owned by you, or by an authority-granting user for whom you work.

*NONE

The distribution being received is not placed in a document object.

document-name

Specify the name of the document in which the distribution is placed. A maximum of 12 characters can be specified.

In folder (FLR)

Specifies the name of the folder that contains the document receiving the distribution. The folder must already exist and you must have the authority to create new documents in the folder.

*NONE

The document being received is not placed in a folder. Specify this value if the document is received into a database file for processing and **Document**(DOC parameter) is not specified.

folder-name

Specify the name of the folder that contains the document. A folder name can consist of a series of folder names if the document is located in a folder contained within another folder. A maximum of 63 characters can be specified.

File to receive output (OUTFILE)

Specifies the name and library of the database file to which the output is directed. If the output file does not exist, this command creates a database file in the specified library. The authority for users with no specific authority 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.

*NONE

The output is not directed to a database file.

data-base-file-name

Specify the name of the database file that receives the output.

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

library-name

Specify the library where the database file is located.

Note: If a new file is created, the system uses QAOSIRCV in QSYS with a format name of OSRCVD as a model.

Output member options (OUTMBR)

Specifies whether the output from the command is displayed at the requesting work station or printed with the job's spooled output.

The possible **member to receive output** values are:

*FIRST

The first member in the file receives the output. If the member does not exist, the system creates a member with the name of the file specified on the **File to receive output** prompt (OUTFILE parameter).

member-name

Specify the name of the file member that receives the output. If the member does not exist, the system creates the file member. If the member already exists, the system adds records to the end of the member or clears the member and then adds the records.

The possible **add or replace** values are:

***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

Type of data for output (OUTDTATYP)

Specifies which type of distribution data is written to the database file.

***DFT** The following record codes are written to the output file:

Record code	Description
010	Distribution Description
020	Message Text
105	Document Description
800	Document Data

***ALL** All record formats are written to the output file.

***DSTINFO**

The distribution description record is written. The record code is 010.

***MSG** The message text record is written. The record code is 020.

***DOCD**

The document description record is written. The record code is 105.

***DOCCLS**

The document class record is written. The record code is 155.

***SUBJECT**

The subject records are written. The record code is 165.

***FILCAB**

The file cabinet reference record is written. The record code is 160.

***AUTHOR**

The author records are written. The record code is 145.

***KWD**

The keyword records are written. The record code is 170.

***CPYLST**

The copy list records are written. The record code is 150.

***FILDATE**

The file date record is written. The record code is 125.

***EXPDATE**

The expiration date record is written. The record code is 115.

***DOCDATE**

The document date record is written. The record code is 120.

***CRTDATE**

The create date record is written. The record code is 110.

***ACTDATE**

The action due date record is written. The record code is 135.

***CHGDATE**

The date last changed record is written. The record code is 130.

***CMPDATE**

The completion date record is written. The record code is 140.

***REF** The reference record is written. The record code is 175.

***STATUS**

The status record is written. The record code is 180.

***PROJECT**

The project record is written. The record code is 185.

***AUTUSR**

The authorizing Userid and Address is written. This is the user that authorized the content of this distribution. The record code is 190.

***DSTEXPDATE**

The distribution expiration date and time is written. The record code is 195.

***RPYDATE**

The reply requested date and time is written. The record code is 200.

***IDP** The interchange document profile (IDP) is written. The record code is 500.

***DOC** The document data records are written. The record code is 800.

Top

Acknowledge receipt (ACKRCV)

Specifies whether a confirmation of delivery (acknowledgement) is sent back to the sender of the distribution.

***YES** The confirmation of delivery (COD) is sent back to the sender.

***NO** The confirmation of delivery is not sent back to the sender.

Top

Distribution ID extension (DSTIDEXN)

Specifies the extension of the distribution identifier (if any) specified on the **Distribution identifier** prompt (DSTID parameter). This 2-digit extension has a value ranging from 01 through 99 that uniquely identifies duplicate distributions. The default value is 01.

***NONE**

There is no duplicate distribution. *NONE is equivalent to an extension of 01.

distribution-id-extension

Specify the extension associated with the distribution. This is used to uniquely identify duplicate distributions.

Keep in mail log (KEEP)

Specifies whether the received distribution is deleted from the mail log or kept in the mail log.

- *NO** When all the information requested has been written to the OUTFILE or DOC, the distribution is removed from the user's incoming mail.
- *YES** When all the information requested has been written to the OUTFILE or to DOC, the distribution is not removed from the user's incoming mail.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

If the values specified on the **Distribution identifier** prompt (DSTID parameter) and **User identifier** prompt (USRID parameter) are being read from an output file created by the Query Distribution (QRYDST) command, specify '930 500' on this parameter.

***SYSVAL**
The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVVD**
The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767
Specify the graphic character set to use.

Element 2: Code page

1-32767
Specify the code page to use.

Top

Examples

Example 1: Receiving Current User Distribution

```
RCVDST  DISTID('SYSTEM1 USERA 0001')
        OUTFILE(MYLIB/MYFILE)  OUTMBR(MYMBR *ADD)
        OUTDTATYP(*ALL)  CMDCHRID(*DEVVD)
```

This command receives the current user distribution into output file MYFILE located in library MYLIB. The distribution is added to member MYMBR. All output file information is added to the output file MYFILE.

Example 2: Receiving Distribution Sent to a User

```
RCVDST  DSTID('BAKER RCH38P 0019')  DSTINDEXN(01)  
        OUTFILE(JOWLIB/DOCUMENTS)  USRID(*CURRENT)
```

This command receives a distribution that was sent to a user. It is copied into the first member in a database file called DOCUMENTS in a library called JOWLIB.

Top

Error messages

*ESCAPE Messages

CPF8A87

Document name &2 not correct.

CPF8A97

Folder name &1 not correct.

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF9096

Cannot use CMDCHRID(*DEVD), DOCCHRID(*DEVD) in batch job.

CPF9098

Distribution not received.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9860

Error occurred during output file processing.

Top

Receive File (RCVF)

Where allowed to run:

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

Threadsafe: Conditional

Parameters
Examples
Error messages

The Receive File (RCVF) command is used by a CL program or ILE CL procedure to receive data from a display device or database file. The command reads a record from the file and puts the data from the record into one or more CL variables. These CL variables were automatically declared in the program when the CL source program was compiled and a Declare File (DCLF) command was processed as part of the source. There is one CL variable for each field in the record format used to receive the data. The data that is entered by a user at the display or is contained in the input record is copied into CL variables in the program by the RCVF command, where it is processed by the program.

Only one record format, of those specified in the DCLF command, can be specified in each RCVF command. If the file has not been opened by a previous RCVF, SNDRCVF, or SNDF command, it is opened by this command. If the file has been previously closed due to an end-of-file condition on a previous RCVF command, an error occurs. The file specified in this command can be overridden if the override command is entered before the file is opened. If the file specified in the DCLF command was a display file when the program was compiled, the file may only be overridden to another display file. If the file was a database file, the file may only be overridden to another database file that has a single record format. However, care should be taken that the fields in the overriding record format correspond to the CL variables declared in the program.

Restrictions:

- This command is valid only within a CL program or ILE CL procedure.
- This command is conditionally threadsafe. The RCVF command is threadsafe when issued against a database file. RCVF is not threadsafe when issued against a display file, and should not be used in a job with multiple threads to receive data from a display file.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Display device	Name, <u>*FILE</u>	Optional, Positional 1
RCDFMT	Record format	Name, <u>*FILE</u>	Optional, Positional 2
OPNID	Open file identifier	Simple name, <u>*NONE</u>	Optional
WAIT	Wait	<u>*YES</u> , *NO	Optional

Top

Display device (DEV)

Specifies the name of the display device from which data is to be received. If a CL variable name is used in this parameter, only one RCVF command is needed in the program to receive data from several devices. (The variable specifying the device name can be changed while repeatedly running the same command.) This parameter may be specified only if the file is a display device file.

***FILE** The user's data is to be received from the device associated with the device file (the device file that was declared in the FILE parameter of the DCLF command). If more than one device name is specified in the device file, *FILE cannot be specified.

name Specify the name of the device or the name of the CL variable that contains the name of the device from which the user's data is to be received.

Top

Record format (RCDFMT)

Specifies the name of the record format that is used to receive data from the file. The format contains all the fields in the record. This parameter must be coded with a record format name if there is more than one record format in the device file. If the file is a database file, the specified record format is used to map the data from the record into the CL variables. The actual record format name in the file at run time may be different. RCVF ignores the INVITE DDS keyword.

***FILE** There is only one record format in the device file; that is the format in which the data is to be received. If more than one record format is specified in the device file, *FILE cannot be specified.

name Specify the name of the record format in which the data records from the display device are to be received. A CL variable cannot be used to specify the record format name.

Top

Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL program or ILE CL procedure. A CL variable cannot be specified for this parameter value.

***NONE**

No open file identifier is provided. This command will use the file associated with the DCLF command that had *NONE specified for the OPNID parameter. Only one file can be declared in a CL program or ILE CL procedure with *NONE as the open file identifier.

simple-name

Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL program or ILE CL procedure.

Top

Wait (WAIT)

Specifies whether the CL program or ILE CL procedure waits for the data to be received from the user's device or continues processing the commands that follow this RCVF command. If WAIT(*NO) is specified, the program must issue a WAIT command later in the program to complete the input operation. This parameter may be specified only if the file is a display device file.

***YES** The program waits until the input operation from the device is completed; the next command is not processed until then.

***NO** The program does not wait for the input data; commands continue running until a WAIT command is reached later in the program.

Top

Examples

Example 1: Receive Data from Database File

```
DCLF FILE(MENU1)
:
RCVF OPNID(*NONE)
```

The CL program or ILE CL procedure receives data from the database file named MENU1. The RCVF command is associated with the Declare File command that does not have an open file identifier specified.

Example 2: Receive Data from Display Device

```
DCLF FILE(SCREENX) RCDfmt(R1 R2)
:
RCVF DEV(DISPLAY2) RCDfmt(R1)
```

The CL program or ILE CL procedure receives data from the user at the display station named DISPLAY2. The data is received in the record format named R1 in the device file named SCREENX. The procedure waits for the user data before it continues processing.

Example 3: Handling End-of-File Exception

```
DCLF FILE(INPUT) OPNID(INFILE1)
:
RCVF OPNID(INFILE1)
MONMSG CPF0864 EXEC(GOTO EOF)
```

The CL program or ILE CL procedure receives a record sequentially from the database file named INPUT. The procedure monitors for the end-of-file exception CPF0864 and goes to label EOF when the message is received.

Example 4: Using RCVF with WAIT command

```
DCLF FILE(MSCREEN) RCDfmt(MIN1 MIN2 MIN3)
:
RCVF DEV(&DNAME) RCDfmt(MIN2) WAIT(*NO)
WAIT DEV(&DNAME)
```

The CL program or ILE CL procedure receives user data from several devices one at a time by way of the device file named MSCREEN. The procedure receives data from the device named in the variable &DNAME using the record format MIN2, but it does not wait for the data to come in. The same RCVF command is used to receive data from several devices; because the CL variable &DNAME is used, only the device name in the DEV parameter must be changed each time the command is run. A WAIT command for each device must be issued later in the procedure because the WAIT command actually receives the data. Both the RCVF and the WAIT commands may be processed for each device (one at a

time) to send data to the procedure. If a user response is delayed, the commands can be processed as many times as necessary until the user responds with the data or a End Receive (ENDRCV) command cancels the request.

Top

Error messages

*ESCAPE Messages

CPF0859

File override caused I/O buffer size to be exceeded.

CPF0860

File &1 in &2 not a data base file.

CPF0861

File &1 in library &2 is not a display file.

CPF0863

Value of binary data too large for decimal CL variable.

CPF0864

End of file detected for file &1 in &2.

CPF0865

File &1 has more than one record format.

CPF0883

*FILE not valid in DEV parameter for file &1.

CPF0886

Record contains a data field that is not valid.

CPF4101

File &2 in library &3 not found or inline data file missing.

CPF502A

Variable length record error on member &4.

CPF502B

Error occurred in trigger program.

CPF502D

Referential constraint violation on member &4.

CPF502E

Referential constraints could not be validated for member &4.

CPF502F

Check constraint violation on member &4.

CPF5029

Data mapping error on member &4.

CPF503A

Referential constraint violation on member &4.

CPF503B

Record could not be inserted or updated in member &4.

CPF5068

Program device &4 not found in file &2 in library &3.

CPF5070

File &2 in library &3 has no program devices acquired.

Top

Receive Journal Entry (RCVJRNE)

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

Parameters
Examples
Error messages

The Receive Journal Entry (RCVJRNE) command allows a specified user exit program to continuously receive journal entries. This program can be set up, for example, to write the entries either (1) to an ICF file, supplying updates to a file on a backup system, or (2) on a tape, imitating a journal-to-tape function. The information in the journal entries received can be used to update the objects being journaled to minimize the loss of data in the event of a disk failure, and to update objects on a backup system in case of a system failure on the primary system.

The value specified on the ENTFMT parameter determines the format of the journal entries passed to the exit program.

Restrictions:

- If the sequence number is reset in the range of the receivers specified, the first occurrence of FROMENTLRG or FROMENT is used, if they are specified. If TOENTLRG or TOENT is specified, the first occurrence after the FROMENTLRG or FROMENT entry is used, if FROMENTLRG or FROMENT is specified. Otherwise the first occurrence is used.
- The FILE, OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, OBJJID, JRNCDE, ENTTYP, JOB, PGM, USRPRF, CCIDLRG, CMTCYCID, and DEPEND parameters can be used to specify a subset of all available entries within a range of journal entries.
 - If no values are specified using these parameters, all available journal entries are received.
 - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT or *IGNOBSLT is specified on the JRNCDE parameter.
 - If a journal code is specified on the JRNCDE parameter and *IGNFILSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the FILE parameter.
 - If a journal code is specified on the JRNCDE parameter and *IGNOBSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, and OBJJID parameters.
- The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to the journal when a receiver size option (RCVSIZOPT) or a fixed length data option (FIXLENDTA) that would have omitted this data was in effect.
- If more than the maximum number of objects is identified (32767 objects), an error occurs and no entries are received. This restriction is ignored if *ALLFILE is specified or no objects are specified.
- When journal caching is being used, entries that are in the cache are not returned.

Top

Parameters

Keyword	Description	Choices	Notes
JRN	Journal	Qualified object name	Required, Positional 1
	Qualifier 1: Journal	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
EXITPGM	Program to receive entries	Qualified object name	Required, Positional 2
	Qualifier 1: Program to receive entries	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
FILE	Journalled file	Single values: *ALLFILE Other values (up to 300 repetitions): <i>Element list</i>	Optional, Positional 3
	Element 1: File	Qualified object name	
	Qualifier 1: File	Name, *ALL	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
	Element 2: Member	Name, <u>*FIRST</u> , *ALL, *NONE	
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	Qualified object name	
	Qualifier 1: Object	Name, *ALL	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
	Element 2: Object type	*FILE, *DTAARA, *DTAQ, *LIB	
	Element 3: Member, if data base file	Name, <u>*FIRST</u> , *ALL, *NONE	
OBJPATH	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Name	Path name	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
SUBTREE	Directory subtree	<u>*NONE</u> , *ALL	Optional
PATTERN	Name pattern	Values (up to 20 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	Character value, *	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
RCVRNG	Range of journal receivers	Single values: *CURRENT, *CURCHAIN Other values: <i>Element list</i>	Optional, Positional 4
	Element 1: Starting journal receiver	Qualified object name	
	Qualifier 1: Starting journal receiver	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
	Element 2: Ending journal receiver	Single values: <u>*CURRENT</u> Other values: <i>Qualified object name</i>	
	Qualifier 1: Ending journal receiver	Name	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
FROMENTLRG	Starting large sequence number	Character value, <u>*FIRST</u>	Optional
FROMTIME	Starting date and time	<i>Element list</i>	Optional
	Element 1: Starting date	Date	
	Element 2: Starting time	Time	
TOENTLRG	Ending large sequence number	Character value, *LAST, <u>*NONE</u>	Optional

Keyword	Description	Choices	Notes
TOTIME	Ending date and time	<i>Element list</i>	Optional
	Element 1: Ending date	<i>Date</i>	
	Element 2: Ending time	<i>Time</i>	
NBRENT	Number of journal entries	<i>Integer, *ALL</i>	Optional
JRNCDL	Journal codes	Single values: <i>*ALL, *CTL</i> Other values (up to 16 repetitions): <i>Element list</i>	Optional
	Element 1: Journal code value	A, B, C, D, E, F, J, L, M, P, R, Q, S, T, U, Y	
	Element 2: Journal code selection	<i>*ALLSLT, *IGNFILSLT, *IGNOBSLT</i>	
ENTTYP	Journal entry types	Single values: <i>*ALL, *RCD</i> Other values (up to 300 repetitions): <i>Character value</i>	Optional
JOB	Job name	Single values: <i>*ALL, *</i> Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
PGM	Program	<i>Name, *ALL</i>	Optional
USRPRF	User profile	<i>Name, *ALL</i>	Optional
CCIDLRG	Commit cycle large identifier	<i>Character value, *ALL</i>	Optional
DEPENT	Dependent entries	<i>*ALL, *NONE</i>	Optional
OBJFID	File identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
OBJJID	Object journal identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
ENTFMT	Entry format	<i>*TYPE1, *TYPE2, *TYPE3, *TYPE4, *TYPEPTR, *JRNENTFMT</i>	Optional
FMTMINDTA	Format minimized data	<i>*NO, *YES</i>	Optional
NULLINDLEN	Null value indicators length	Single values: <i>*ENTFMT</i> Other values: <i>Element list</i>	Optional
	Element 1: Field data format	1-8000, <i>*VARLEN</i>	
	Element 2: Variable length field length	1-8000	
DELAY	Delay time	<i>Element list</i>	Optional
	Element 1: Delay time value	1-99999, <u>30</u> , <i>*NEXTENT</i>	
	Element 2: Maximum delay time value	1-99999, <i>*CLS</i>	
BLKLEN	Block length	32-4000, <i>*NONE, *CALC</i>	Optional
JRNENTFMT	Journal entry format	<u>RJNE0100</u> , RJNE0200	Optional
RTNPTR	Return pointers	<i>*NONE, *SYSMNG, *USRMNG</i>	Optional
INCENT	Include entries	<i>*CONFIRMED, *ALL</i>	Optional
FROMENT	Starting sequence number	1-999999999, <i>*FIRST</i>	Optional
TOENT	Ending sequence number	1-999999999, <i>*LAST, *NONE</i>	Optional
CMTCYCID	Commit cycle identifier	1-999999999, <i>*ALL</i>	Optional

Top

Journal (JRN)

Specifies the journal from which the journal entries are received.

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

Program to receive entries (EXITPGM)

Specifies a user-written exit program that controls the receiving of each journal entry passed from the command. Additional information on the interface between this command and the exit program is supplied after the listing of possible values for this parameter, and is described in more detail in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "receiving journal entries in an exit program".

This is a required parameter.

Qualifier 1: Program to receive entries

program-name

Specify the name of the exit program that controls the reception of each journal entry passed from the command.

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.

library-name

Specify the library where the program is located.

Additional Information on the Exit Program Interface

When the program is called, two parameters are passed to it at a time. A single journal entry or a block of journal entries is passed in the first parameter.

- If a single journal entry is passed, and if the length of the parameter defined by the program is smaller than the length of the journal entry, the journal entry passed to the program is truncated. If the length of the parameter defined by the program is greater than the length of the journal entry, the parameter

positions beyond the length of the journal entry contain nonessential information. The user's program should not specifically refer to data in the positions beyond the length of the journal entry.

- If BLKLEN(*NONE) was specified, then the exit program may indicate to the system that multiple entries should be returned in subsequent calls to the exit program by specifying 8 in the first byte of the second parameter. This is called "Block Mode" and the exit program must specify the size of the block in bytes as a zoned value in the first 5 bytes of the first parameter. If an error is made in this specification, only one journal entry is passed in the block. If BLKLEN(*NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored.
- If pointers are being returned, this first parameter must be aligned on a 16-byte boundary since journal entry specific data could include actual pointers.

For *TYPE1, *TYPE2, *TYPE3 and *TYPE4 formats there is a zoned journal entry length field that is filled with zeros at the end of the single journal entry or block of journal entries passed. This field indicates that the last journal entry has been passed. The format of the information in each journal entry is shown in the ENTFMT parameter description. The format of the first parameter is detailed in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: The maximum length of the parameter specification in the exit program is language dependent (for example, for CL, the maximum length is 9999). For more information about limitations refer to the corresponding programming language book.

A character variable of LEN(3) is passed in the second exit program parameter. This parameter will be passed from the system to the exit program and can be passed from the exit program to the system. Its values are presented in the following lists:

- Information in the first byte of the second parameter:

Char(1)

Passed to the Exit Program

- 0 No journal entry is passed on this call of the exit program.
- 1 A single journal entry is passed to the exit program.
- 2 A block of one or more journal entries is passed to the exit program.
- 3 No journal entry is passed on this call to the exit program, and no more entries can be passed, because the journal receiver that was attached when the receive journal entry operation started is no longer attached.

Note: The system ends the RCVJRNE command after calling the exit program once with a reason code of 3.

- 4 No journal entry is passed on this call to the exit program, and no more entries can be passed unless the remote journal is activated.

Note: This value can only be passed to the exit program when receiving journal entries from the attached receiver of a remote journal and the journal state for the journal is currently *INACTIVE.

Char(1)

Passed to the System from the Exit Program

- 8 Requests the command processing program to start passing one or more journal entries in a block. If BLKLEN(*NONE) was not specified, then specifying 8 in the first byte of the second parameter will have no effect and the first 5 bytes of the first parameter will be ignored.
 - 9 Requests the RCVJRNE command to end. The exit program returns control to the system.
- Information in the second byte of the second parameter:

Char(1)**Passed to the Exit Program**

- N** Additional journal entries are not currently available to be passed after this call of the exit program, or the RCVJRNE command will be ending after this call of the exit program.
- Y** Additional journal entries are currently available to be passed after this call of the exit program.

Any information passed from the exit program to the system in this second character will be ignored. This second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

- Information in the third byte of the second parameter:

Char(1)**Passed to the Exit Program**

- '00'x** One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal.

Note: This value is returned only when receiving journal entries from a journal receiver that was attached to a journal prior to V4R2M0.

- 0** No journal entries are currently being passed, so the information normally returned in this byte is not applicable.
- 1** One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry reflect the name of the object at the time the journal entry was deposited into the journal.
- 2** One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal. The object name in the fixed length portion of the journal entry may be returned as a previously known name for the object prior to the journal entry being deposited into the journal or be returned as *UNKNOWN.
- Note:** This value will only be returned when receiving journal entries from a remote journal and the remote journal is currently being caught up from its source journal. A remote journal is being caught up from its source journal when the Change Remote Journal (CHGRMTJRN) command or Change Journal State (QjoChangeJournalState) API is invoked and is currently replicating journal entries to the remote journal. After the call to the CHGRMTJRN command or QjoChangeJournalState API returns, the remote journal is maintained with a synchronous or asynchronous delivery mode, and the remote journal is no longer being caught up.
- 3** One or more journal entries are being passed to the exit program and the object names in the fixed length portion of each journal entry do not necessarily reflect the name of the object at the time the journal entry was deposited into the journal. The object name in the fixed length portion of the journal entry may be returned as *UNKNOWN.

Any information passed from the exit program to the system in the third byte will be ignored. The second byte of the second exit program parameter is provided whether journal entries are being passed as a single journal entry per call of the exit program, or as a block of journal entries per call.

Note: When an **N** is passed to the exit program in the second byte of the second parameter indicating that no additional journal entries are currently available, it does not necessarily mean that when the exit program returns, that the RCVJRNE command will have to wait for additional journal entries to be deposited into the journal. By the time the exit program returns, additional journal entries may already be available and depending upon what was specified on the DELAY parameter, may or may not be immediately passed to the exit program. If DELAY(N) was specified the system will wait N seconds before passing the journal entries to the exit program. If DELAY(*NEXTENT) was specified, the journal entries will immediately be passed to the exit program.

The third byte of the second exit program parameter is provided whether journal entries are being processed as a single journal entry per call of the exit program, or as a block of journal entries per call. When returned for a block of journal entries, the attribute applies to the object names for all of the journal entries being returned in the block.

For more information on the exit program and these two parameters used to receive the journal entries, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Journalled physical file (FILE)

Specifies a maximum of 300 qualified file names whose journal entries are received. This parameter also specifies the name of the file member whose journal entries are to be received.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file member name, the following is done:

- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier is determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are received. Specify the library name or *CURLIB to have entries returned for the file.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved, or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying a database file on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters:
 - Journal code D (database file-level information entries).
 - Journal code F (file member-level information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are received.

Single values

*ALLFILE

The search for the journal entries received is not limited to a specified file name. All journal entries are received, regardless of which objects, if any, the entries are associated with.

Element 1: Journaled physical file

Qualifier 1: Journalled physical file

***ALL** Journal entries for all physical and logical files in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are received. If *ALL is specified and the user does not have the required authority to all of the files, an error occurs, and the command ends.

file-name

Specify the name of the database physical or logical file for which a journal entry is received.

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.

Element 2: Member

***FIRST**

Entries for the database file and the first member in the file are received. This value is not valid for remote journals.

***ALL** Entries for the database file and all the currently existing members of the file are received.

***NONE**

Only entries for the database file are received. Entries for members of the file are not received.

member-name

Specify the name of the member for which journal entries are received.

If *ALL is specified for the file-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FIRST is specified on the FILE parameter, the journal entries of the first members of all applicable files in the specified library are received.

Top

Objects (OBJ)

Specifies a maximum of 300 qualified object names whose journal entries are to be received. The possible object types are *FILE, *DTAARA, *DTAQ, and *LIB. If *FILE is specified, this parameter also specifies the name of the file member whose journal entries are to be received.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified object name, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are received. Specify the library name or *CURLIB to have entries returned for an object.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the object name specification:
 - Journal code D (database file-level information entries).
 - Journal code E (data area information entries).
 - Journal code F (file member-level information entries).
 - Journal code Q (data queue information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Journal code Y (library information entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

Element 1: Object

Qualifier 1: Object

***ALL** Journal entries for all objects of the specified object type in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are received. The library name must be specified. If *ALL is specified and the user does not have the required authority for all objects in the library, a message is sent and the command ends.

object-name

Specify the name of the object whose journaled changes are to be received.

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.

Element 2: Object type

Specify the object type of the object whose journaled changes are to be received.

***FILE** Entries for database files and database file members are received.

***DTAARA**

Entries for data areas are received.

***DTAQ**

Entries for data queues are received.

***LIB** Entries for libraries are received.

Element 3: Member, if data base file

Specify the name of the member in the file whose journal entries are to be received. If *ALL is specified for the first part of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if *FIRST is specified, the journal entries of the first member of all applicable files in the specified library are received

Note: If the specified object type is not *FILE, the member name element value is ignored.

*FIRST

Entries for the database file and the first member in the file are received. This value is not valid for remote journals.

***ALL** Entries for the database file and all the currently existing members of the file are received.

***NONE**

Only entries for the database file are received. Entries for members of the file are not received.

member-name

Specify the name of the member for which journal entries are received.

If *ALL is specified for the object-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FILE *FIRST is specified on the OBJ parameter, the journal entries of the first members of all applicable files in the specified library are received.

Top

Objects (OBJPATH)

Specifies a maximum of 300 objects whose journal entries are to be received. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" (/), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

This parameter is not valid for remote journals.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

Only objects that are currently linked with the specified path name and have a journal identifier associated with them are used in journal entry selection. If the specified object does exist, the journal identifier associated with that link is used for journal entry selection. If a specified object does not exist or does not have a journal identifier associated with it, that link is not used in selecting journal entries and no error is sent.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the object name specification:
 - Journal code B (integrated file system information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

Element 1: Name

path-name

Entries for objects identified by the path name are received.

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 topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 2: Include or omit

The second element specifies whether names that match the path name 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 included in determining what journal entries are received unless overridden by an *OMIT specification.

***OMIT**

The objects that match the object name pattern are not to be included in determining what journal entries are received. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Directory subtree (SUBTREE)

Specifies whether the directory subtrees are included in determining the objects for which journal entries are to be received.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

***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 subdirectories and the objects within those subdirectories.

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Name pattern (PATTERN)

Specifies a maximum of 20 patterns to be used to include or omit objects for which journal entries are to be received.

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.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

Element 1: Pattern

'*' All objects that match the input OBJPATH parameter are to be included.

name-pattern

Specify the pattern to be used to include or omit objects for which journal entries are received. 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.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

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 included in the operation, unless overridden by an *OMIT specification.

*OMIT

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

Top

Range of journal receivers (RCVRNG)

Specifies the starting (first) and ending (last) journal receivers used in the search for the journal entries that are received. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the last value) is processed.

If a problem is found in the receiver chain (such as damaged or not-found receivers) before the search operation begins, the system tries to use the second of the dual receivers. If these receivers also are damaged or not found, the operation ends.

Single values

*CURRENT

The journal receiver that is currently attached when starting to receive journal entries is used.

*CURCHAIN

The journal receiver chain that includes the journal receiver that is currently attached when starting to receive journal entries is used. This receiver chain does not cross a break in the chain. If there is a break in the chain, the receiver range is from the most recent break in the chain through the receiver that is attached when starting to receive journal entries.

Element 1: Starting journal receiver

Qualifier 1: Starting journal receiver

starting-journal-receiver-name

Specify the name of the first journal receiver containing journal entries that are to be received.

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 journal receiver. If no library is specified as the current library for the job, QGPL is used.

library-name

Specify the library where the journal receiver is located.

Element 2: Ending journal receiver

Single values

*CURRENT

The journal receiver that is currently attached when starting to receive journal entries is used.

Qualifier 1: Starting journal receiver

ending-journal-receiver-name

Specify the name of the last journal receiver containing journal entries that can be received. If the end of the receiver chain is reached before a receiver with this name is found, an error message is sent and no journal entries are received.

Note: The maximum number of receivers in the range is 1024. If more receivers than this maximum are specified, an exception is signaled, and no journal entries are received.

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 journal receiver. If no library is specified as the current library for the job, QGPL is used.

library-name

Specify the library where the journal receiver is located.

Top

Starting large sequence number (FROMENTLRG)

Specifies the first journal entry considered for reception.

Note: You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

*FIRST

The first journal entry in the specified journal receiver range is the first entry considered for reception.

starting-sequence-number

Specify the sequence number of the first journal entry considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

Top

Starting date and time (FROMTIME)

Specifies the date and time of the first journal entry considered for reception. The starting date and time of the first journal entry created either at or after the specified starting date and time is the starting point for reception of the journal entries.

Element 1: Starting date

starting-date

Specify a starting date.

Element 2: Starting time

starting-time

Specify a starting time. The time can be specified in 24-hour format with or without a time separator:

- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds.

Top

Ending large sequence number (TOENTLRG)

Specifies the last journal entry considered for reception.

Note: You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

***NONE**

No journal entry is specified. Journal entries are passed to the exit program until the command is canceled (by a cancel request or a cancel job command) or until an end reason code (9) is set by the exit program. If there are no more entries to pass, the RCVJRNE command waits the number of seconds indicated on the DELAY parameter before trying to find more entries to pass.

Note: TOENTLRG(*NONE) is valid only if the RCVRNG parameter specifies a receiver that is currently attached when starting to receive journal entries.

***LAST**

The last journal entry in the journal receiver range specified is the last entry considered for reception.

ending-sequence-number

Specify the sequence number of the final journal entry considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

Note: The values specified for the from and to prompts can be the same. For example, FROMENTLRG(234) and TOENTLRG(234) can be specified.

Top

Ending date and time (TOTIME)

Specifies the date and time of the last journal entry considered for reception. The ending date and time of the journal entry created at or before the specified ending date and time is the ending point for reception of the journal entries.

Element 1: Ending date

ending-date

Specify the date of the last entry received.

Element 2: Ending time

ending-time

Specify the creation time of the last entry received. The time can be specified in 24-hour format with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Number of journal entries (NBRENT)

Specifies the total number of journal entries that are received.

***ALL** All journal entries included in the specified journal receiver range that satisfy the selection values are received.

value Specify the maximum number of journal entries be received. If the specified journal entry

identified by the TOENTLRG, TOENT, or TOTIME parameter is reached before the value specified for NBRENT is met, the command ends normally.

Top

Journal codes (JRNCDE)

Specifies the journal code that is used to limit the entries being considered for reception.

Single values

- ***ALL** The journal entries received are not limited to those containing a specified code.
- ***CTL** The journal entries received are those written to control the journal functions. These journal entries have codes J or F.

Element 1: Journal code value

journal-code

Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are received.

An explanation of the journal codes that can be specified is in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Element 2: Journal code selection

***ALLSLT**

The journal entries with the specified journal code are received only if all other selection parameters are satisfied.

***IGNFILSLT**

Journal entries having the specified journal code are received only if all selection parameters, except the FILE parameter, are satisfied.

Note: This value is not valid for journal codes D, F, and R. This value is not valid if the OBJ, OBJPATH, OBJFID, or OBJJID parameters are specified.

***IGNOBSLT**

Journal entries having the specified journal code are received only if all selection parameters are satisfied except OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, and OBJJID.

Note: This value is not valid for journal codes B, D, E, F, Q, R, and Y. This value is not valid if the FILE parameter is specified.

Top

Journal entry types (ENTTYP)

Specifies whether to limit the journal entries received to those of a specified journal entry type.

Single values

- ***ALL** The journal entries that can be received are not limited to those of a specified journal entry type.
- ***RCD** Only entries that have an entry type for record level operations are received. The following entry types are valid: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

Other values

entry-type

Specify the entry type that limits the journal entries received. Only journal entries that contain the specified entry type are received. Up to 300 valid entry types can be specified. More information on entry types is in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Top

Job name (JOB)

Specifies the journal entries considered for reception based on their associated jobs.

Single values

***ALL** The journal entries received are not limited to those for a specified job.

***** The journal entries received are limited to those for the current job.

Other values

job-identifier

Specify the job name, the user name, and the job number of the job to use. You can also specify that the job name only, or that the job name and the user name be used.

job-name

Specify the job name of the job.

user-name

Specify the user name of the job.

job-number

Specify the system-assigned job number.

Top

Program (PGM)

Specifies the journal entries considered for reception based on their associated programs.

***ALL** The journal entries received are not limited to those created by a specified program.

program-name

Specify the name of the program whose journal entries are considered for reception.

Top

User profile (USRPRF)

Specifies that the journal entries considered for reception are limited to the journal entries created for the specified user profile.

***ALL** The journal entries received are not limited to those for a specified user profile.

user-profile-name

Specify the name of the user profile whose journal entries are considered for reception.

Top

Commit cycle large identifier (CCIDLRG)

Specifies the journal entries considered for reception based on their associated commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

***ALL** The journal entries received are not limited to a specified commit cycle identifier.

commit-cycle-identifier

Specify the commit cycle identifier of the journal entries to be considered for reception. The possible range is 1 to 18,446,744,073,709,551,600.

Top

Dependent entries (DEPENT)

Specifies whether to receive the journal entries recording actions

- that occur as a result of a trigger program
- on records that are part of a referential constraint
- that will be ignored during an Apply Journalized Changes (APYJRNCHG) or Remove Journalized Changes (RMVJRNCHG) operation.

***ALL** The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journalized Changes operations are received.

***NONE**

The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journalized Changes operations are not received.

Top

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) for which journal entries are to be received. 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 or *SYMLNK that are in the "root" (/), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file identifier, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are received.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are received.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are received only if they satisfy the values specified on the other parameters in addition to the FID specification:
 - Journal code B (integrated file system information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are received.

file-identifier

Entries for objects identified with the FID are received.

Top

Object journal identifier (OBJJID)

Specifies a maximum of 300 journal identifiers for which journal entries are to be received. This field is input in hexadecimal format. Hexadecimal zero is not valid.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying a journal identifier on this parameter, journal entries with the following journal code values are converted for output only if they satisfy the values specified on the other parameters in addition to the journal identifier specification:
 - Journal code B (integrated file system information entries).
 - Journal code D (database file-level information entries).
 - Journal code E (data area information entries).
 - Journal code F (file member-level information entries).
 - Journal code J (journal receiver information entries).
 - Journal code Q (data queue information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Journal code Y (library information entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are converted for output.

journal-identifier

Entries for objects associated with the specified journal identifier are received.

Top

Entry format (ENTFMT)

Specifies the format of the journal entries being received. For a description of what is represented by each of the fields in the journal entry, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Note: If ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is not specified, the NULLINDLEN parameter must be specified.

Note: If the **Receiver size options** field (RCVSIZOPT) for the journal was specified as *MAXOPT3, the sequence number, commit cycle identifier, the count of entries applied or removed or relative record number fields can reach a maximum value of 18,446,744,073,709,551,600. The length of these fields in the ENTFMT(*TYPE1), ENTFMT(*TYPE2), ENTFMT(*TYPE3), and ENTFMT(*TYPE4) formats is defined to hold a 10 digit number. If a sequence number, commit cycle identifier, or count of entries applied or removed or relative record number larger than 10 digits is found and one of these ENTFMT options is specified, the field is set to -1 for that entry.

*TYPE1

The journal entries received are formatted to include the minimum information that can be specified. The information fields and the format of the information in each journal entry is shown below:

Table 2. Figure: Table 1 - *TYPE1 Journal Entry Format

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (2)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Date	: 6	: 19	: 24
Time	: 6	: 25	: 30
Job Name	: 10	: 31	: 40
User Name	: 10	: 41	: 50
Job Number	: 6	: 51	: 56
Program Name	: 10	: 57	: 66
Object Name	: 10	: 67	: 76
Object Library	: 10	: 77	: 86
Member Name	: 10	: 87	: 96
Count/RRN (3)	: 10	: 97	: 106
Flag	: 1	: 107	: 107
Commit Cycle ID (4)	: 10	: 108	: 117
Incomplete data	: 1	: 118	: 118
Minimized Entry	: 1	: 119	: 119
Specific Data	:	:	:
Reserved	: 6	: 120	: 125
Entry-Specific Data	: N (1)	: 126	: N+125

Notes:

- (1) The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.
- (2) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

***TYPE2**

The journal entries received include the information returned when ENTFMT(*TYPE1) is specified, the user profile field, which gives the name of the user who caused the logging of the received journal entries, and the name of the system on which the entry was sent. The format for *TYPE2 journal entries is shown below:

Table 3. Figure: Table 2 - *TYPE2 Journal Entry Format

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (2)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Date	: 6	: 19	: 24
Time	: 6	: 25	: 30
Job Name	: 10	: 31	: 40
User Name	: 10	: 41	: 50
Job Number	: 6	: 51	: 56
Program Name	: 10	: 57	: 66
Object Name	: 10	: 67	: 76
Object Library	: 10	: 77	: 86
Member Name	: 10	: 87	: 96
Count/RRN (3)	: 10	: 97	: 106
Flag	: 1	: 107	: 107
Commit Cycle ID (4)	: 10	: 108	: 117
User Profile	: 10	: 118	: 127
System Name	: 8	: 128	: 135
Incomplete data	: 1	: 136	: 136
Minimized Entry	: 1	: 137	: 137
Specific Data	:	:	:
Reserved	: 18	: 138	: 155
Entry-Specific Data	: N (1)	: 156	: N + 155

Notes:

(1) The length of the entry-specific data field varies from entry to entry. It is long enough to accommodate all the entry-specific data in each received journal entry.

(2) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.

(3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.

(4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

***TYPE3**

The journal entries received include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE3.

***TYPE4**

The journal entries received include the information returned when ENTFMT(*TYPE3) is specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the three formats for *TYPE4.

***TYPEPTR**

The journal entries received include the information as if ENTFMT(*JRNENTFMT) JRNENTFMT(RJNE0100) RTNPTR(*SYSMNG) were specified, and the entry specific data could contain pointers for specific journal entry types. Because ENTFMT(*TYPEPTR) does not allow RTNPTR(*USRMNG), pointers will not be valid after the exit program returns control to the system. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the two formats for *TYPEPTR.

Note: NULLINDLEN(*VARLEN) and ENTFMT(*TYPEPTR) cannot be specified at the same time.

***JRNENTFMT**

The format of the received journal entries is determined by the JRNENTFMT parameter. The RTNPTR parameter indicates if the received journal entries could contain pointers. If RTNPTR(*USRMNG) is specified, pointers will still be valid after the exit program returns control to the system. The format of the received entries depends on the value specified on the NULLINDLEN parameter. The tables in the NULLINDLEN parameter description show the six formats for *JRNENTFMT.

Top

Format minimized data (FMTMINDTA)

Specifies whether entry specific data which has been minimized on field boundaries will be returned in a readable format.

***NO** The journal entries which have entry specific data that has been minimized on field boundaries will not be returned in a readable format. Therefore, the entry specific data may not be viewable.

***YES** The journal entries which have entry specific data that has been minimized on field boundaries will be returned in a readable format. Therefore, the entry specific data is viewable and may be used for auditing purposes. The fields that were changed are accurately reflected. The fields that were not changed and were not recorded, display default data and are indicated by a value of 'F9'X in the null value indicators field.

Top

Null value indicators length (NULLINDLEN)

Specifies the length, in bytes, used for the Null Value Indicators portion of the journal entry received by the user. This parameter is not valid if ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is specified.

Null value indicators are present in journal entries for record level operations as follows:

1. The corresponding physical file has null capable fields.
2. The record image has been minimized in the entry specific data.

If the record image has not been minimized in the entry specific data, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

- 'F0'X = Corresponding field is not null.

- 'F1'X = Corresponding field is null.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*YES) was specified on the RCVJRNE command, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

- 'F0'X = Corresponding field is not null.
- 'F1'X = Corresponding field is null.
- 'F9'X = Corresponding field was not changed and the default value for the field is returned.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*NO) was specified on the RCVJRNE command, then an internal value is returned for the null value indicator.

Single values

*ENTFMT

The null value indicators field is long enough to include all of the null value indicators in the received journal entries. Since the number of null value indicators can vary from entry to entry, the length of the null value indicators field also varies with each entry.

Notes:

If users select the *TYPE3 format, the following information is not available in this format:

1. Incomplete Data indicating if the journal entry data is incomplete due to either LOB fields or Byte Stream File write operations.
2. Minimized Entry Specific Data indicating if the journal entry has minimized entry specific data because the journal had MINENTDTA specified for the object type of the journal entry.

See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information on the incomplete data indicator, the minimized entry specific data indicator, and these journal entries.

The format for *TYPE3 journal entries when NULLINDLEN(*ENTFMT) is specified is shown below:

Table 4. Figure: Table 3 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE3)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (5)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (6)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (7)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Number of Null Value Indicators (1)	: 5	: 150	: 154
Null Value Indicators	: M (2)	: 155	: 154 + M
Length of Entry-Specific Data (3)	: 5	: 155 + M	: 159 + M
Entry-Specific Data	: N (4)	: 160 + M	: 159 + M
	:	:	: + N

Notes:

- (1) This field contains the number of null value indicators (in decimal digits) in the received journal entry.
- (2) The length of null value indicators can vary from entry to entry and is designated by the variable M.
- (3) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
- (4) The length of entry-specific data can vary from entry to entry and is designated by the variable N.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (6) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (7) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPE4 journal entries when NULLINDLEN(*ENTFMT) is specified is shown below:

Table 5. Figure: Table 4 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE4)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (5)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (6)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (7)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Journal Identifier	: 10	: 150	: 159
Referential Constraint	: 1	: 160	: 160
Trigger	: 1	: 161	: 161
Incomplete data	: 1	: 162	: 162
Ignore during APYJRNCHG or RMVJRNCHG	: 1	: 163	: 163
Minimized Entry	: 1	: 164	: 164
Specific Data	:	:	:
Reserved	: 5	: 165	: 169
Number of Null Value Indicators (1)	: 5	: 170	: 174
Null Value Indicators	: M (2)	: 175	: 174 + M
Length of Entry-Specific Data (3)	: 5	: 175 + M	: 179 + M
Entry-Specific Data	: N (4)	: 180 + M	: 179 + M
	:	:	: + N

Notes:

(1) This field contains the number of null value indicators (in decimal digits) in the received journal entry.

(2) The length of null value indicators can vary from entry to entry and is designated by the variable M.

(3) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.

(4) The length of entry-specific data can vary from entry to entry and is designated by the variable N.

(5) When the RCVSIZOPT of the journal is *MAXOPT3, this field

636 be set to 1 if the sequence number is larger than 10 digits.

(6) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.

(7) When the RCVSIZOPT of the journal is *MAXOPT3, this field

The format for *TYPEPTR or *JRNENTFMT journal entries when NULLINDLEN(*ENTFMT), JRNENTFMT(RJNE0100), and RTNPTR(*SYSMNG) are specified is shown below:

Table 6. Figure: Table 5 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPEPTR) or ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0100) is specified

Field Name	Length	From	To
Displacement to next journal entry's header (1)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (1)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (1)	: 4	: 8	: 11
Pointer handle (2)	: 4	: 12	: 15
Sequence Number	: 20	: 16	: 35
Journal Code	: 1	: 36	: 36
Entry Type	: 2	: 37	: 38
Time stamp	: 26	: 39	: 64
Job Name	: 10	: 65	: 74
User Name	: 10	: 75	: 84
Job Number	: 6	: 85	: 90
Program Name	: 10	: 91	: 100
Object	: 30	: 101	: 130
Count/RRN	: 10	: 131	: 140
Indicator Flag	: 1	: 141	: 141
Commit Cycle ID	: 20	: 142	: 161
User Profile	: 10	: 162	: 171
System Name	: 8	: 172	: 179
Journal Identifier	: 10	: 180	: 189
Referential Constraint	: 1	: 190	: 190
Trigger	: 1	: 191	: 191
Incomplete Data	: 1	: 192	: 192
Object Name Indicator	: 1	: 193	: 193
Ignore During APYJRNCHG: or RMVJRNCHG	: 1	: 194	: 194
Minimized Entry Specific Data	: 1	: 195	: 195
Length Of Null Vallue Indicators (1)	: 4	: NVI Disp : (3)	: NVI Disp : + 3
Null Value Indicators	: M (4)	: NVI Disp : + 4	: NVI Disp : + 3
Length Of Entry Specific Data	: 5	: ESD Disp : (5)	: ESD Disp : + 4
Reserved	: 11	: ESD Disp : + 5	: ESD Disp : + 15
Entry-Specific Data	: N (6)	: ESD Disp	: ESD Disp

Notes:
 (1) This field is represented in BINARY(4).
 (2) This field is represented in UNSIGNED BINARY(4).

The format for *JRNENTFMT journal entries when NULLINDLEN(*ENTFMT) and JRNENTFMT(RJNE0200) are specified is shown below:

Table 7. Figure: Table 6 - NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0200) is specified

Field Name	: Length	: From	: To
Displacement to next journal entry's header (5)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (5)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (5)	: 4	: 8	: 11
Displacement to this journal entry's trans-action identifier (1)	: 4	: 12	: 15
Displacement to this journal entry's logical unit of work (1)	: 4	: 16	: 19
Displacement to this journal entry's receiver information(1)	: 4	: 20	: 23
Sequence Number (1)	: 8	: 24	: 31
Unformatted Time stamp (1)	: 8	: 32	: 39
Thread Identifier (1)	: 8	: 40	: 47
System Sequence Number (1)	: 8	: 48	: 55
Count/RRN (1)	: 8	: 56	: 63
Commit Cycle ID (1)	: 8	: 64	: 71
Pointer Handle (1)	: 4	: 72	: 75
Remote Port (1)	: 2	: 76	: 77
Arm Number (1)	: 2	: 78	: 79
Program Library ASP Number (1)	: 2	: 80	: 81
Remote Address	: 16	: 82	: 97
Journal Code	: 1	: 98	: 98
Entry Type	: 2	: 99	: 100
Job Name	: 10	: 101	: 110
User Name	: 10	: 111	: 120
Job Number	: 6	: 121	: 126
Program Name	: 10	: 127	: 136
Program Library Name	: 10	: 137	: 146
Program Library ASP Device Name	: 10	: 147	: 156
Object	: 30	: 157	: 186
User Profile	: 10	: 187	: 196
Journal Identifier	: 10	: 197	: 206
System Name	: 8	: 208	: 215
Indicator Flag	: 1	: 216	: 216

field-length

Specify the field length of the null value indicators portion of the received journal entry. Valid values range from 1 through 8000 characters.

The format for *TYPE3 journal entries when NULLINDLEN(field-length) is specified is shown below:

Table 8. Figure: Table 7 - NULLINDLEN(field-length) Journal Entry Format for ENTfmt(*TYPE3)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (3)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (4)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (5)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Null Value Indicators	: field : length : (1)	: 150 : :	: 149 + : field : length
Entry-Specific Data	: M (2) : :	: 150 + : field : length	: 149 + M + : field : length

Notes:
 (1) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.
 (2) The length of the entry-specific data field varies from entry to entry and is designated by the variable M. This length accommodates all the entry-specific data in each received journal entry.
 (3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
 (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
 (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPE4 journal entries when NULLINDLEN(field-length) is specified is shown below:

Table 9. Figure: Table 8 - NULLINDLEN(field-length) Journal Entry Format for ENTfmt(*TYPE4)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (3)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (4)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (5)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Journal Identifier	: 10	: 150	: 159
Referential Constraint	: 1	: 160	: 160
Trigger	: 1	: 161	: 161
Incomplete data	: 1	: 162	: 162
Ignore during APYJRNCHG or RMVJRNCHG	: 1	: 163	: 163
Minimized Entry	: 1	: 164	: 164
Specific Data	:	:	:
Reserved	: 5	: 165	: 169
Null Value Indicators	: field : length : (1)	: 170 : :	: 169 + : field : length
Entry-Specific Data	: M (2) : :	: 170 + : field : length	: 169 + M + : field : length

Notes:

- (1) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.
- (2) The length of the entry-specific data field varies from entry to entry and is designated by the variable M. This length accommodates all the entry-specific data in each received journal entry.
- (3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPEPTR or *JRNENTFMT journal entries when NULLINDLEN(field-length) and JRNENTFMT(RJNE0100) are specified is shown below:

Table 10. Figure: Table 9 - NULLINDLEN(field-length) Journal Entry Format for ENTFRMT(*TYPEPTR) or ENTFRMT(*JRNENTFRMT) when JRNENTFRMT(RJNE0100) is specified

Field Name	: Length	: From	: To
Displacement to next journal entry's header (1)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (1)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (1)	: 4	: 8	: 11
Pointer handle (2)	: 4	: 12	: 15
Sequence Number	: 20	: 16	: 35
Journal Code	: 1	: 36	: 36
Entry Type	: 2	: 37	: 38
Time stamp	: 26	: 39	: 64
Job Name	: 10	: 65	: 74
User Name	: 10	: 75	: 84
Job Number	: 6	: 85	: 90
Program Name	: 10	: 91	: 100
Object	: 30	: 101	: 130
Count/RRN	: 10	: 131	: 140
Indicator Flag	: 1	: 141	: 141
Commit Cycle ID	: 20	: 142	: 161
User Profile	: 10	: 162	: 171
System Name	: 8	: 172	: 179
Journal Identifier	: 10	: 180	: 189
Referential Constraint	: 1	: 190	: 190
Trigger	: 1	: 191	: 191
Incomplete Data	: 1	: 192	: 192
Object Name Indicator	: 1	: 193	: 193
Ignore During APYJRNCHG: or RMVJRNCHG	: 1	: 194	: 194
Minimized Entry Specific Data	: 1	: 195	: 195
Null Value Indicators	: field length : (3)	: NVI Disp : (4)	: NVI Disp : + field length : - 1
Length Of Entry Specific Data	: 5	: ESD Disp : (5)	: ESD Disp : + 4
Reserved	: 11	: ESD Disp : + 5	: ESD Disp : + 15
Entry-Specific Data	: N (6)	: ESD Disp : + 16	: ESD Disp : + 15 + N

Notes:

- (1) This field is represented in BINARY(4).
- (2) This field is represented in UNSIGNED BINARY(4).
- (3) The length of the null value indicators field is the length specified on the NULLINDLEN parameter.

The format for *JRNENTFMT journal entries when NULLINDLEN(field-length) and JRNENTFMT(RJNE0200) are specified is shown below:

Table 11. Figure: Table 10 - NULLINDLEN(field-length) Journal Entry Format for ENTFFMT(*JRNENTFFMT) when JRNENTFFMT(RJNE0200) is specified

Field Name	: Length	: From	: To
Displacement to next journal entry's header (5)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (5)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (5)	: 4	: 8	: 11
Displacement to this journal entry's trans-action identifier (1)	: 4	: 12	: 15
Displacement to this journal entry's logical unit of work (1)	: 4	: 16	: 19
Displacement to this journal entry's receiver information(1)	: 4	: 20	: 23
Sequence Number (1)	: 8	: 24	: 31
Unformatted Time stamp (1)	: 8	: 32	: 39
Thread Identifier (1)	: 8	: 40	: 47
System Sequence Number (1)	: 8	: 48	: 55
Count/RRN (1)	: 8	: 56	: 63
Commit Cycle ID (1)	: 8	: 64	: 71
Pointer Handle (1)	: 4	: 72	: 75
Remote Port (1)	: 2	: 76	: 77
Arm Number (1)	: 2	: 78	: 79
Program Library ASP Number (1)	: 2	: 80	: 81
Remote Address	: 16	: 82	: 97
Journal Code	: 1	: 98	: 98
Entry Type	: 2	: 99	: 100
Job Name	: 10	: 101	: 110
User Name	: 10	: 111	: 120
Job Number	: 6	: 121	: 126
Program Name	: 10	: 127	: 136
Program Library Name	: 10	: 137	: 146
Program Library ASP Device Name	: 10	: 147	: 156
Object	: 30	: 157	: 186
User Profile	: 10	: 187	: 196
Journal Identifier	: 10	: 197	: 206
Address Family	: 1	: 207	: 207
System Name	: 8	: 208	: 215
Indicator Flag	: 1	: 216	: 216

Element 1: Field data format

***VARLEN**

The null value indicators field is a variable length field.

field-length

Specify the field length of the null value indicators portion of the received journal entry. Valid values range from 1 through 8000 characters.

Element 2: Variable length field length

maximum-field-length

Specify the maximum field length of the null value indicators portion of the received journal entry. Valid values range from 1 to 8000 characters. If a journal entry has more null value indicators than the value you specify and truncation results in the loss of a non 'F0'X indicator value, the RCVJRNE request is abnormally ended.

You can specify this element only if you also specify *VARLEN on the first element of this parameter.

Note: NULLINDLEN(*VARLEN) and ENTFMT(*TYPEPTR) cannot be specified at the same time.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

The format for *TYPE3 journal entries when NULLINDLEN(*VARLEN field-length) is specified is shown below:

Table 12. Figure: Table 11 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTfmt(*TYPE3)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (4)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (5)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (6)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Number of Null Value Indicators (1)	: 2	: 150	: 151
Null Value Indicators	: field : length :	: 152 : :	: 151 + : field : length
Length of Entry-Specific Data (2)	: 5 : :	: 152 + : field : length	: 156 + : field : length
Entry-Specific Data	: M (3) : :	: 157 + : field : length	: 156 + M + : field : length

Notes:

- (1) This field contains the number of null value indicators (in binary digits) in the received journal entry.
- (2) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
- (3) The length of entry-specific data can vary from entry to entry and is designated by the variable M.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (6) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

The format for *TYPE4 journal entries when NULLINDLEN(*VARLEN field-length) is specified is shown below:

Table 13. Figure: Table 12 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTfmt(*TYPE4)

Field Name	: Length	: From	: To
Entry Length	: 5	: 1	: 5
Sequence Number (4)	: 10	: 6	: 15
Journal Code	: 1	: 16	: 16
Journal Entry Type	: 2	: 17	: 18
Timestamp	: 26	: 19	: 44
Job Name	: 10	: 45	: 54
User Name	: 10	: 55	: 64
Job Number	: 6	: 65	: 70
Program Name	: 10	: 71	: 80
Object Name	: 10	: 81	: 90
Object Library	: 10	: 91	: 100
Member Name	: 10	: 101	: 110
Count/RRN (5)	: 10	: 111	: 120
Flag	: 1	: 121	: 121
Commit Cycle ID (6)	: 10	: 122	: 131
User Profile	: 10	: 132	: 141
System Name	: 8	: 142	: 149
Journal Identifier	: 10	: 150	: 159
Referential Constraint	: 1	: 160	: 160
Trigger	: 1	: 161	: 161
Incomplete data	: 1	: 162	: 162
Ignore during APYJRNCHG or RMVJRNCHG	: 1	: 163	: 163
Minimized Entry	: 1	: 164	: 164
Specific Data	:	:	:
Reserved	: 5	: 165	: 169
Number of Null Value Indicators (1)	: 2	: 170	: 171
Null Value Indicators	: field : length :	: 172 : :	: 171 + : field : length
Length of Entry-Specific Data (2)	: 5 : :	: 172 + : field : length	: 176 + : field : length
Entry-Specific Data	: M (3) : :	: 177 + : field : length	: 176 + M + : field : length

Notes:

- (1) This field contains the number of null value indicators (in binary digits) in the received journal entry.
- (2) This field contains the length of the entry-specific data (in decimal digits) in the received journal entry.
- (3) The length of entry-specific data can vary from entry to entry and is designated by the variable M.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field

The format for *JRNENTFMT journal entries when NULLINDLEN(*VARLEN field-length) and JRNENTFMT(RJNE0100) are specified is shown below:

Table 14. Figure: Table 13 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFRMT(*JRNENTFRMT) when JRNENTFRMT(RJNE0100) is specified

Field Name	: Length	: From	: To
Displacement to next journal entry's header (1)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (1)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (1)	: 4	: 8	: 11
Pointer handle (2)	: 4	: 12	: 15
Sequence Number	: 20	: 16	: 35
Journal Code	: 1	: 36	: 36
Entry Type	: 2	: 37	: 38
Time stamp	: 26	: 39	: 64
Job Name	: 10	: 65	: 74
User Name	: 10	: 75	: 84
Job Number	: 6	: 85	: 90
Program Name	: 10	: 91	: 100
Object	: 30	: 101	: 130
Count/RRN	: 10	: 131	: 140
Indicator Flag	: 1	: 141	: 141
Commit Cycle ID	: 20	: 142	: 161
User Profile	: 10	: 162	: 171
System Name	: 8	: 172	: 179
Journal Identifier	: 10	: 180	: 189
Referential Constraint	: 1	: 190	: 190
Trigger	: 1	: 191	: 191
Incomplete Data	: 1	: 192	: 192
Object Name Indicator	: 1	: 193	: 193
Ignore During APYJRNCHG: or RMVJRNCHG	: 1	: 194	: 194
Minimized Entry Specific Data	: 1	: 195	: 195
Length Of Null Value Indicators (1)	: 4	: NVI Disp	: NVI Disp
	:	: (3)	: + 3
Null Value Indicators	: field	: NVI Disp	: NVI Disp
	: length	: + 4	: + 3
	:	:	: + field
	:	:	: length
Length Of Entry Specific Data	: 5	: ESD Disp	: ESD Disp
	:	: (4)	: + 4
Reserved	: 11	: ESD Disp	: ESD Disp
	:	: + 5	: + 15
Entry-Specific Data	: M (5)	: ESD Disp	: ESD Disp
	:	: + 16	: + 15 + M

Notes:
(1) This field is represented in BINARY(4).

The format for *JRNENTFMT journal entries when NULLINDLEN(*VARLEN field-length) and JRNENTFMT(RJNE0200) are specified is shown below:

Table 15. Figure: Table 14 - NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFRMT(*JRNENTFRMT) when JRNENTFRMT(RJNE0200) is specified

Field Name	: Length	: From	: To
Displacement to next journal entry's header (5)	: 4	: 0	: 3
Displacement to this journal entry's null value indicators (5)	: 4	: 4	: 7
Displacement to this journal entry's entry specific data (5)	: 4	: 8	: 11
Displacement to this journal entry's trans-action identifier (1)	: 4	: 12	: 15
Displacement to this journal entry's logical unit of work (1)	: 4	: 16	: 19
Displacement to this journal entry's receiver information(1)	: 4	: 20	: 23
Sequence Number (1)	: 8	: 24	: 31
Unformatted Time stamp (1)	: 8	: 32	: 39
Thread Identifier (1)	: 8	: 40	: 47
System Sequence Number (1)	: 8	: 48	: 55
Count/RRN (1)	: 8	: 56	: 63
Commit Cycle ID (1)	: 8	: 64	: 71
Pointer Handle (1)	: 4	: 72	: 75
Remote Port (1)	: 2	: 76	: 77
Arm Number (1)	: 2	: 78	: 79
Program Library ASP Number (1)	: 2	: 80	: 81
Remote Address	: 16	: 82	: 97
Journal Code	: 1	: 98	: 98
Entry Type	: 2	: 99	: 100
Job Name	: 10	: 101	: 110
User Name	: 10	: 111	: 120
Job Number	: 6	: 121	: 126
Program Name	: 10	: 127	: 136
Program Library Name	: 10	: 137	: 146
Program Library ASP Device Name	: 10	: 147	: 156
Object	: 30	: 157	: 186
User Profile	: 10	: 187	: 196
Journal Identifier	: 10	: 197	: 206
Address Family	: 1	: 207	: 207
System Name	: 8	: 208	: 215
Indicator Flag	: 1	: 216	: 216

Delay time (DELAY)

Specifies the number of seconds that the command processing program (CPP) waits for a new journal entry to arrive if the last entry has already been received. After the last entry in the journal is received and passed to the exit program, the CPP tries to receive the next entry. If no new journal entry exists, the exit program is passed a value of 0 in the first byte of the second parameter.

Note: This parameter is valid only when TOENTLRG(*NONE) and TOENT(*NONE) is specified, and the last receiver specified on the RCVRNG parameter identifies the journal receiver that is currently attached when journal entries are starting to be received.

When the last entry on the journal has been passed to the exit program and no journal entries are currently available to be passed to the exit program, one of the following occurs:

- If a number of seconds is specified for the first element in the list, the exit program is immediately called and a '0' is passed to the first byte of the second exit parameter indicating that no additional journal entries are currently available. When the exit program returns control to the command, the system delays for the specified number of seconds.

When the delay time has expired, the system then checks whether any additional journal entries are available to be passed to the exit program. Any additional entries are passed to the exit program sequentially, until there are no more available. When there are no further journal entries available, the exit program is called, and a '0' is passed as the first byte of the second exit program parameter, indicating there are no more journal entries currently available. When the exit program returns control to the command, the system again delays for the specified number of seconds.

If there are no new journal entries to pass to the exit program after the delay, the exit program is called, and a '0' is passed as the first byte of the second exit program parameter to indicate that no further journal entries are available. The exit program then passes the value '9' for the first byte of the second parameter, indicating that this command is to end.

- If *NEXTENT is specified for the first element in the list, then additional journal entries are passed to the exit program as they become available. When this option is used, the second element in the list indicates the maximum number of seconds between calls to the exit program. If there are no additional journal entries to pass after the specified maximum delay time, the exit program is called, and a '0' is passed to the first byte of the second exit program parameter, indicating that no additional journal entries are currently available.

The maximum delay time can be either of the following:

- The time between a call to the exit program passing the last currently available journal entry, and a subsequent call to the exit program indicating that no new journal entries are available.
- The time between calls to the exit program indicating that no additional journal entries are available.

If the exit program is called after the maximum delay has expired, it then can pass the value '9' for the first byte of the second parameter, indicating that this command should be ended.

Note: The previous description of the DELAY parameter assumes that the journal receiver that is currently attached at the beginning of the RCVJRNE command is still attached. If that journal receiver has been detached, the exit program is sent the reason code 3 after all journal entries have been received by the exit program and the RCVJRNE command ends.

Element 1: Delay time value

- 30 The command delays 30 seconds before checking whether additional journal entries are available to be passed to the exit program.

***NEXTENT**

A fixed delay time is not used. Additional journal entries are passed to the exit program as they become available.

Note: If the RCVJRNE exit program causes any additional calls of the RCVJRNE command, those additional calls cannot specify DELAY(*NEXTENT) if a preceding call specified TOENTLRG(*NONE) or TOENT(*NONE).

Note: INCENT(*ALL) and DELAY(*NEXTENT) cannot be specified at the same time.

seconds

Specify the number of seconds that the command delays before checking whether additional journal entries are available to be passed to the exit program. Valid values range from 1 through 99999.

Element 2: Maximum delay time value

This element indicates the maximum number of seconds between calls to the exit program when a fixed delay time is not specified on the first element. This element is valid only if *NEXTENT is specified for the first element.

***CLS** The process default wait time is used as the maximum number of seconds between calls to the exit program.

seconds

Specify the maximum length of time between calls to the exit program, in seconds. Valid values range from 1 through 99999.

Top

Block length (BLKLEN)

Specifies whether the system will be sending one or more journal entries to the exit program and specifies the block length of the buffer passed to the exit program. The EXITPGM parameter has further details about block mode semantics.

***NONE**

At most one journal entry will be sent to the exit program.

***CALC**

One or more journal entries will be passed to the exit program in a block. The length of the block passed (the first parameter passed to the exit program) is determined by the system and will be optimal.

block-length

Specify the length in kilobytes of the buffer passed to the exit program (EXITPGM parameter). Valid values range from 32 to 4000.

Top

Journal entry format (JRNENTMFT)

Specifies the format of the journal entries received by the exit program. The formats are described in the Retrieve Journal Entries (QjoRetrieveJournalEntries) API. See the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for the details associated with these formats.

This parameter is only valid if ENTTFMT(*JRNENTTFMT) is also specified.

RJNE0100

Received journal entries are in RJNE0100 format.

RJNE0200

Received journal entries are in RJNE0200 format. While in block mode, a single block of entries will not be received from multiple receivers.

Top

Return pointers (RTNPTR)

Specifies whether the journal entries received include entry specific data that could contain pointers for specific journal entry types.

This parameter is only valid if ENTFMT(*JRNENTFMT) is also specified.

***NONE**

Received journal entries will not include pointers.

***SYSMNG**

Specifies that journal entries received include entry specific data that could contain pointers for specific journal entry types. The system will manage the releasing of the resources related to pointers that are received in the journal entries. The pointers, their associated pointer handles and any related storage will be released when the exit program returns control to the system.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

***USRMNG**

Specifies that journal entries received include entry specific data that could contain pointers for specific journal entry types. The user will manage the releasing of the resources related to pointers that are received in the journal entries. Neither the pointers, their associated pointer handles or any related storage will be released when the exit program returns control to the system. The user must use the Delete Pointer Handle (QjoDeletePointerHandle) API to release these resources. See the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for the details associated with this API.

Note: NULLINDLEN(*VARLEN) and RTNPTR(*SYSMNG) or RTNPTR(*USRMNG) cannot be specified at the same time.

Top

Include entries (INCENT)

Specifies whether only the confirmed or both the confirmed and unconfirmed, journal entries are received. This parameter only applies when receiving journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

*CONFIRMED

Only those journal entries which have been confirmed are received.

***ALL** All confirmed and unconfirmed journal entries are received.

Note: INCENT(*ALL) and DELAY(*NEXTENT) cannot be specified at the same time.

Top

Starting sequence number (FROMENT)

Specifies the first journal entry considered for reception.

Note: You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

*FIRST

The first journal entry in the specified journal receiver range is the first entry considered for reception.

starting-sequence-number

Specify the sequence number of the first journal entry considered for reception. The possible range is 1 to 9,999,999,999.

Top

Ending sequence number (TOENT)

Specifies the last journal entry considered for reception.

Note: You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

*NONE

No journal entry is specified. Journal entries are passed to the exit program until the command is canceled (by a cancel request or a cancel job command) or until an end reason code (9) is set by the exit program. If there are no more entries to pass, the RCVJRNE command waits the number of seconds indicated on the DELAY parameter before trying to find more entries to pass.

Note: TOENT(*NONE) is valid only if the RCVRNG parameter specifies a receiver that is currently attached when starting to receive journal entries.

*LAST

The last journal entry in the journal receiver range specified is the last entry considered for reception.

ending-sequence-number

Specify the sequence number of the final journal entry considered for reception. The possible range is 1 to 9,999,999,999.

Note: The values specified for the from and to prompts can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

Top

Commit cycle identifier (CMTCYCID)

Specifies the journal entries considered for reception based on their associated commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

***ALL** The journal entries received are not limited to a specified commit cycle identifier.

commit-cycle-identifier

Specify the commit cycle identifier of the journal entries to be considered for reception. The possible range is 1 to 9,999,999,999.

Top

Examples

Example 1: Receiving Journal Entries

```
RCVJRNE  JRN(APPLIB/JRN1)  EXITPGM(MYLIB/RCVPGM)
          FILE(APPLIB/FILE3) TOENT(*LAST)  ENTFMT(*TYPE3)
          NULLINDLEN(*ENTFMT)
```

This command receives journal entries from the journal receiver that is currently attached (when journal entries are starting to be received) to the journal JRN1 in library APPLIB and passes them one at a time to program RCVPGM in library MYLIB. Only entries with file-level information for the first member of file FILE3 in library APPLIB are received. The format of each entry passed to the exit program is shown in the "NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*TYPE3)" table shown within the NULLINDLEN parameter description.

Example 2: Receiving Journal Entries

```
RCVJRNE  JRN(JRNLIB/MYJRN) EXITPGM(RCVLIB/PGMA)
          FILE(FILELIB/PFILEB MBRONE)
          TOENT(*LAST)  ENTFMT(*TYPE3)  NULLINDLEN(*VARLEN 30)
```

This command receives journal entries with file-level information for member MBRONE of file PFILEB in library FILELIB from the journal receiver currently attached (when journal entries are starting to be received) to journal MYJRN in library JRNLIB and sends them one at a time to program PGMA in library RCVLIB. The format of each entry passed to the exit program is shown in the "NULLINDLEN(*VARLEN field-length) Journal Entry Format for ENTFMT(*TYPE3)" table at the end of the NULLINDLEN parameter description. The null value indicators portion of each received entry is 30 characters in length.

Example 3: Receiving Journal Entries Using DELAY(*NEXTENT)

```
RCVJRNE  JRN(JRNLIB/MYJRN) EXITPGM(RCVLIB/PGMA)
          RCVRNG(*CURCHAIN) TOENT(*NONE)  DELAY(*NEXTENT)
```

This command receives all available journal entries from the chain of journal receivers, which includes the journal receiver that is attached at the start of receiving journal entries, associated with the journal

MYJRN in the library JRNLIB. These journal entries are sent sequentially to exit program PGMA in library RCVLIB, as they become available. The maximum length of time between calls to the exit program is equal to the process default wait time value.

Example 4: Receiving Journal Entries for Data Area, Data Queue, and Integrated File System Objects

```
RCVJRNE  JRN(LIBPROD/PRODJRN) EXITPGM(RCVLIB/PGMA)
         OBJ((APPLIB/D1 *DTAARA) (APPLIB/D2 *DTAQ)
         (APPLIB/D3 *FILE *NONE))
         OBJPATH('/mydirectory') SUBTREE(*YES)
         TOENT(*LAST) ENTFMT(*JRNENTFMT) JRNENTFMT(RJNE0200)
```

This command receives journal entries from the journal receiver currently attached (when journal entries are starting to be received) to journal PRODJRN in library LIBPROD and sends them one at a time to program PGMA in library RCVLIB. The format of each entry passed to the exit program is shown in the "NULLINDLEN(*ENTFMT) Journal Entry Format for ENTFMT(*JRNENTFMT) when JRNENTFMT(RJNE0200) is specified" table at the end of the NULLINDLEN parameter description. Only entries associated with the specified objects are converted. These objects are a data area in library APPLIB called D1, a data queue in library APPLIB called D2, and a database file D3 in library APPLIB, as well as the directory '/mydirectory' and all directories, stream files, and symbolic links within that directory or one of its subdirectories.

Top

Error messages

*ESCAPE Messages

CPF7002

File &1 in library &2 not a physical file.

CPF7006

Member &3 not found in file &1 in &2.

CPF7007

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

CPF701B

Journal recovery of an interrupted operation failed.

CPF705C

INCENT(*ALL) not allowed for a local journal.

CPF7053

Values for RCVRNG parameter not correct; reason code &1.

CPF7054

FROM and TO values not valid.

CPF7055

Maximum number of objects exceeded.

CPF7057

*LIBL not allowed with *ALL or *ALLLIB or a generic name.

CPF706A

Significant null value indicator truncated.

CPF706D

RCVJRNE exit program &1 in &2 failed.

CPF7060
Object not journaled in specified receiver range.

CPF7061
Conversion of journal entries failed.

CPF7062
No entries converted or received from journal &1.

CPF7065
Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF707B
DELAY(*NEXTENT) not allowed.

CPF7074
RCVRNG for specified SEARCH not valid.

CPF708C
DELAY(*NEXTENT) not allowed with INCENT(*ALL).

CPF708D
Journal receiver found logically damaged.

CPF709C
JOB, PGM, and USRPRF not valid for receiver range.

CPF7096
RCVRNG and TOENTLRG or TOENT parameters do not match.

CPF70A9
OBJPATH parameter not valid for a remote journal.

CPF70AC
Object for file identifier &1 not found.

CPF70AE
Member *FIRST not allowed for a 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.

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

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

CPF9825
Not authorized to device &1.

Top

Receive Message (RCVMSG)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

Parameters
Examples
Error messages

The Receive Message (RCVMSG) command is used by a program to receive a message previously sent to a message queue.

The RCVMSG command receives messages from a job message queue (a message queue associated with a call stack entry or the external message queue (*EXT)), or from a named message queue. The program can receive a message from a message queue associated with its own call stack entry or from a message queue associated with another call stack entry.

This command copies a message received in the specified message queue into control language (CL) variables within the program. The message and its attributes are copied into the CL variables specified by the parameters KEYVAR through DTACCSID.

You can specify the message being received by indicating the message type, the reference key of the message, or both. The program receiving the message can also specify, on the RCVMSG command, whether a message is removed from the message queue or left there as an old message. If the specified message queue is not allocated to the job in which this command is entered, or to any other job, the message queue is implicitly allocated by this command for the duration of the command's processing.

If a message of the specified type does not exist on the queue, the requesting program can either wait for a message to arrive or continue with other processing. This allows a set of message queues to be polled.

If the message received is an unhandled exception message, the program can specify whether this command should handle the exception. An unhandled exception message is an escape, status, or notify message that has been sent to an Integrated Language Environment (ILE) procedure. When this command is run, the ILE procedure has not yet taken action to tell the system that the exception is handled. One action the ILE procedure can take is to call a CL program that receives the message using this command. More information on actions that can be taken is in the ILE Concepts book, SC41-5606.

Top

Parameters

Keyword	Description	Choices	Notes
PGMQ	Call stack entry message queue	Single values: *EXT Other values: <i>Element list</i>	Optional, Positional 1
	Element 1: Relationship	*SAME, *PRV	
	Element 2: Call stack entry identifier	<i>Element list</i>	
	Element 1: Call stack entry	<i>Character value</i> , *	
	Element 2: Module	<i>Name</i> , *NONE	
	Element 3: Bound program	<i>Name</i> , *NONE	
MSGQ	Message queue	Single values: *PGMQ Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	

Keyword	Description	Choices	Notes
MSGTYPE	Message type	*ANY, *NEXT, *PRV, *INFO, *INQ, *RPY, *FIRST, *COPY, *COMP, *DIAG, *EXCP, *RQS, *LAST	Optional, Positional 3
MSGKEY	Message key	Character value, *NONE, *TOP	Optional, Positional 4
WAIT	Wait time	Integer, 0, *MAX	Optional, Positional 5
RMV	Remove message	*YES, *NO, *KEEPEXCP	Optional
CCSID	Coded character set ID	1-65535, *HEX, *JOB	Optional
RJTDFTRPY	Reject default reply	*NOALWRJT, *ALWRJT	Optional
KEYVAR	CL var for KEYVAR (4)	Character value	Optional
MSG	CL var for 1st level text	Character value	Optional
MSGLEN	CL var for MSGLEN (5 0)	Decimal number	Optional
SECLVL	CL var for 2nd level text	Character value	Optional
SECLVLEN	CL var for SECLVLEN (5 0)	Decimal number	Optional
MSGDTA	CL var for msg data	Not restricted	Optional
MSGDTALEN	CL var for MSGDTALEN (5 0)	Decimal number	Optional
MSGID	CL var for MSGID (7)	Character value	Optional
SEV	CL var for SEV (2 0)	Decimal number	Optional
SENDER	CL var for SENDER (80)	Character value	Optional
SENDERFMT	Sender format	*SHORT, *LONG	Optional
RTNTYPE	CL var for RTNTYPE (2)	Character value	Optional
ALROPT	CL var for ALROPT (9)	Character value	Optional
MSGF	CL var for MSGF (10)	Character value	Optional
MSGFLIB	CL var for MSGFLIB (10)	Character value	Optional
SNDMSGFLIB	CL var for SNDMSGFLIB (10)	Character value	Optional
TXTCSSID	CL var for text CCSID (5 0)	Decimal number	Optional
DTACSSID	CL var for data CCSID (5 0)	Decimal number	Optional

Top

Call stack entry message queue (PGMQ)

Specifies the call stack entry message queue from which a message is received. The call stack entry message queue can be the *EXT queue or it can be a message queue that is associated with a call stack entry for a program or an ILE procedure.

If values are specified for this parameter, specifying *PGMQ for the **Message queue (MSGQ)** parameter is allowed.

Single values

***EXT** The message is received from the external message queue of the job. The external message queue is used to communicate with the external requester of the job, such as a display station user.

Element 1: Relationship

Element 1 of this parameter specifies whether the message queue is associated with the program or procedure identified by Element 2, or if it is associated with the caller of the program or procedure.

*SAME

The message is received from the message queue of the program or procedure identified by element 2.

***PRV** The message is received from the message queue of the program or procedure that called the program or procedure identified by element 2 of this parameter.

Note: If the message queue previous to the one identified by element 2 is for an ILE program entry procedure (PEP), the message will be received from the message queue immediately previous to the PEP message queue; effectively this would be *two* message queues previous to the one identified by element 2.

Element 2: Call stack entry identifier

The second element of this parameter has three elements. Element 1 specifies an OPM program or ILE procedure name or a special value. Element 2 specifies an ILE module name which is used as a qualifier for the value specified in element 1. Element 3 can specify either an OPM program name or an ILE program name or a service program name, depending on what is specified in element 1. Element 3 is also used as a qualifier for what is specified in element 1.

Element 1: Call stack entry

***** Specifies the OPM program or ILE procedure running this command.

name Specify the name of the OPM program or ILE procedure used to identify the call stack entry.

If this element identifies an OPM program, the name specified can be a maximum of 10 characters. If this element identifies an ILE procedure, the name specified can be a maximum of 256 characters.

Nested procedure names can be specified by separating each procedure name with a colon (:). When specifying nested procedure names, the outermost procedure name is identified first, followed by its contained procedures. The innermost procedure name is identified last in the string.

Partial names of programs or procedures can be specified by placing three less-than symbols (<<<) at the beginning of the name or by placing three greater-than symbols (>>>) at the end of the name. If both the greater-than symbols and the less-than symbols are used, the program or procedure name specified is limited to 250 characters.

The system begins its search for the specified program or procedure name with the most recently called program or procedure.

When searching for a partial program or procedure name:

- The less-than symbols (<<<) are truncated when specified only at the beginning of a program or procedure name and the remaining character string is right-justified. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the last position of the program or procedure name and comparing backward.
- The greater-than symbols (>>>) are truncated when specified only at the end of a program or procedure name. The remaining characters in the specified string are compared to the current program or procedure on the call stack, starting with the first position of the program or procedure name.
- The less-than symbols (<<<) and the greater-than symbols (>>>) are truncated when both are specified for a program or procedure name. The remaining characters are used to scan and compare the entire length of the specified string with the current program or procedure on the call stack.

Element 2: Module

*NONE

No ILE module qualifier is provided.

name Specify the ILE module name to be used to identify the message queue.

Element 3: Program

*NONE

No program qualifier is provided.

name Specify the program name to be used to identify the message queue.

Top

Message queue (MSGQ)

Specifies the message queue (not a program message queue) from which a message is to be received.

Single values

*PGMQ

The program message queue specified for the **Call stack entry message queue (PGMQ)** parameter is the only queue from which a message is received.

Qualifier 1: Message queue

name Specify the name of the message queue from which a message is to be received. If a message queue name is specified, the **Call stack entry message queue (PGMQ)** parameter cannot be specified.

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 library is specified as the current library for the job, QGPL is used.

name Specify the library where the message queue is located.

Top

Message type (MSGTYPE)

Specifies the type of message received by this program.

*ANY Any type of message (except a sender's copy) is received. To receive a sender's message, MSGTYPE(*COPY) must be specified.

*NEXT

The message that follows the one specified for the **Message key (MSGKEY)** parameter is received. If there is not another message available, blanks are returned in all CL variables.

When a message is received from a message queue associated with a call stack entry, *NEXT works only for one call stack entry. *NEXT cannot be used to receive messages for multiple call stack entries of the same program.

- *PRV** The message previous to the message specified for the **Message key (MSGKEY)** parameter is received.
- *INFO**
An informational message is received.
- *INQ** An inquiry message is received.
- *RPY** A reply message is received. This program has sent an inquiry message to a message queue and expects a reply.
- *FIRST**
The first message currently on the message queue or program queue is received.
- *COPY**
A copy of an inquiry message that was previously sent is received by this program. The message queue specified for the **Call stack entry message queue (PGMQ)** parameter or the **Message queue (MSGQ)** parameter must be the same queue that was specified for the **Message queue to get reply (RPYMSGQ)** parameter when the INQ message was sent.
- *COMP**
A completion message is received. This type of message can only be received from a program message queue.
- *DIAG**
A diagnostic message is received. This type of message can only be received from a program message queue.
- *EXCP**
An exception message is received. Exception messages (escape, notify, status) are received by the program in last-in first-out (LIFO) order. The receiving program can monitor for exception messages by using the MONMSG command.

Note: Non-exception messages are received in first-in first-out (FIFO) order.

If an exception message is received from a message queue for a procedure, the related exception may not be handled at the time the RCVMSG command is run. The RMV parameter can be used to specify whether the exception is to be handled by the RCVMSG command.
- *RQS** A request message is received. This type of message can only be received from a program message queue.
- *LAST**
The last message currently on the message queue or program queue is received.

Top

Message key (MSGKEY)

Specifies the message reference key of the message that is received.

***NONE**

No message reference key is specified.

- *TOP** The top of the message queue is used. *TOP can be used only when *NEXT is specified for the **Message type (MSGTYPE)** parameter. It causes the first message on the message queue to be received. For program message queues, this is the message following the last request message that was received, if any.

name Specify the name of the CL variable that contains the message reference key of the message to be used by this receive function. The variable must be a character variable having a length of 4 characters.

Top

Wait time (WAIT)

Specifies, in seconds, the length of time that the program waits for a message of the specified type to arrive in the message queue if it is not there when this command is processed. If the message does not arrive in the specified time, the control language (CL) variables named to receive message fields are filled with blanks (or zeros, if they are decimal variables).

The program cannot wait for a message from a program message queue unless it is receiving a reply.

If a wait time is specified (not zero), the message queue is implicitly allocated to the first user whose message is received, and it is not released until the request has been handled by the program.

If a message is sent to a message queue in the same job, and the message queue is in break delivery mode, this parameter is ignored (that implies WAIT(0), which is the default value for the WAIT parameter).

If the value specified for MSGKEY refers to an inquiry message, and MSGTYPE(*RPY) has been specified, the program ignores the WAIT parameter (value for Wait is 0).

0 The program does not wait for the arrival of a message.

***MAX** The program waits indefinitely for the arrival of the specified message.

number-of-seconds

Specifies the number of seconds that the program waits for the arrival of a message.

Top

Remove message (RMV)

Specifies whether the message received by the program is removed from the message queue. For messages that are unhandled exceptions, this parameter also specifies whether the exception is to be handled. If *INQ is specified for the **Message type (MSGTYPE)** parameter, then *NO must also be specified for this parameter so a reply to the inquiry message can be sent, otherwise the default reply will be sent before the unanswered inquiry is removed.

*YES The message is removed from the message queue. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

***NO** The message is not removed from the message queue. It is left on the message queue as an old message. If the message is an unhandled exception, the exception is handled by running the RCVMSG command.

Note: Old messages are messages that have been received but not deleted. An old message can be received again in one of the following ways:

1. The message reference key of the message is specified for the MSGKEY parameter.
2. A message type of *FIRST, *LAST, *NEXT, or *PRV is specified for the **Message type (MSGTYPE)** parameter.

***KEEPEXCP**

If the message is an exception message and the exception has not been handled, the exception is left unhandled and the message is left on the message queue as a new message. It can be

received again by using the RCVMSG command to receive an *EXCP message. If the message is not an exception message, or if it is but the exception has already been handled, the message is left on the message queue as an old message.

To handle an exception after the RCVMSG has been run, the command can be run a second time by specifying RMV(*YES) or RMV(*NO).

Top

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that you want the message text returned in. This only applies to text returned in the MSG, SECLVL and MSGDTA parameters. When replacement data is returned in the MSGDTA parameter or substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement text that is defined as a character that can be converted (*CCHAR) is converted. The rest of the replacement data is not converted. For more information about the *CCHAR field, see the ADDMSGD command.

***JOB** The received message is converted to the CCSID of the job before being returned.

***HEX** The received message is not converted before being returned.

coded-character-set-identifier

Specify the CCSID that you want your message converted to before being returned. Valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter> for a list of valid values. Only CCSID values that a job can be changed to are accepted.

For more information on the message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Reject default reply (RJTDFTRPY)

Removing an unanswered inquiry message causes the default reply to be sent to the inquiry message. This value indicates whether a reply handling exit program will be allowed to reject a default reply that is sent as a result of using this command. A reply handling exit program can be registered via the system registration facility for exit point QIBM_QMH_REPLY_INQ. This parameter is only applicable when *YES is specified for the RMV keyword.

***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

CL var for KEYVAR (4) (KEYVAR)

Specifies the name of the control language (CL) character variable, if any, that contains the message reference key identifying the message received by the program containing this command. At the time the RCVMSG command is processed, the system returns the message reference key to the variable specified by KEYVAR in this command and changes the received message to an old message. The message reference key can then be used in the MSGKEY parameter in a subsequent RCVMSG command to receive the old message. If the message is not found, blanks are returned for the KEYVAR variable. For reply type messages, use the MSGKEY parameter on this command in conjunction with the KEYVAR parameter on the SNDPGMMSG command. The message reference key can also be used by this program for building message subfiles. The CL variable is the name of the field for which the SFLMSGKEY keyword is specified in the DDS for the message subfile.

Note: For message queues not associated with call stack entries, message reference keys can be used again after a message has been received and then removed (by specifying *YES for the RMV parameter).

The variable must be a character variable having a length of 4 characters.

Note: When using the message reference key (obtained from the CL variable specified by the KEYVAR parameter of the Send Program Message (SNDPGMMSG) command) to receive the reply to an inquiry message, note that the message reference key refers to the sender's copy. The sender's copy message is located on the reply message queue (which defaults to the program message queue that sent the inquiry message), not the message queue to which the inquiry message was sent.

Top

CL var for 1st level text (MSG)

Specifies the name of the control language (CL) character variable, if any, that contains the message when it is received by the program. This includes the message data fields that were substituted for substitution variables in the text before the message was sent (replies and immediate messages contain no message data fields). This is a variable-length field, but most message text is less than 132 characters in length.

Top

CL var for MSGLEN (5 0) (MSGLEN)

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message text available to be received. The variable must be a decimal variable having a length of 5 positions.

Top

CL var for 2nd level text (SECLVL)

Specifies the name of the CL character variable, if any, that contains the message help received by the program. This includes the message data fields that were substituted for any substitution variables in the text before the message was sent (replies and immediate messages do not have second-level messages). This is a variable-length field, but most online message help is designed to be less than 3000 characters in length.

Top

CL var for SECLVLEN (5 0) (SECLVLEN)

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message help available to be received. The variable must be a decimal variable having a length of 5 positions.

Top

CL var for msg data (MSGDTA)

Specifies the name of the control language (CL) character variable, if any, that contains the message data record received by the program as part of the message. The message data record contains the substitution values (in a single character string) that are used in the text of the received message. The amount of data returned and its format depend on the message. Pointers contained in system messages are invalidated.

Note: If you use data that has an invalidated pointer in it an error message can occur.

Top

CL var for MSGDTALEN (5 0) (MSGDTALEN)

Specifies the name of the control language (CL) decimal variable, if any, that contains the total length of the message data record available to be received. The variable must be a decimal variable having a length of 5 positions.

Top

CL var for MSGID (7) (MSGID)

Specifies the name of the control language (CL) character variable, if any, that contains the message identifier of the message received by the program. If the message being received is an immediate message, the message identifier is returned as blanks. The minimum length of the variable is 7 characters.

Top

CL var for SEV (2 0) (SEV)

Specifies the name of the control language (CL) decimal variable, if any, that contains the severity code of the message received by the program. If the message being received is an immediate message, the message severity is not returned. The variable must be a decimal variable having a length of 2 positions.

Top

CL var for SENDER (80) (SENDER)

Specifies the name of the control language (CL) character variable, if any, that contains the identification of the sender of the message received through the RCVMSG command. The length of the CL variable depends on the SENDERFMT specification. If SENDERFMT(*SHORT) is specified, the variable must be a minimum of 80 characters. If the CL variable is longer than 80 characters, additional information will be returned. If SENDERFMT(*LONG) is specified, the variable must be a minimum of 720 characters.

Top

Sender format (SENDERFMT)

Specifies which format of the sender identification is returned. This parameter is valid only when the SENDER parameter is specified.

*SHORT

The short format of the sender information is returned. The short format is a minimum of 80 characters, If the CL variable is longer than 80 characters, additional information will be returned. Positions in the CL variable beyond the last returned field will be set to blanks. The following information is returned:

- The first 26 characters identify the sending job
 - Job name (10)
 - User name (10)
 - Job number (6)
- The next 16 characters identify the sending program
 - Program name (12) (for an ILE procedure, this is the bound program name); if the sender type is 3, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name
 - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 13 characters are the date and time
 - Date (7) (in the format 0yymmdd)
 - Time (6) (in the format hhmmss)
- The next 14 characters identify the sent-to call stack entry if the message is sent to a program message queue
 - Program name (10) (for an ILE procedure, this is the bound program name)
 - Instruction number (4) (for an ILE procedure, this field is set to blanks)
- The next 1 character identifies the sender type
 - "0" if the sender is an OPM program or a SLIC program with 12 characters or less
 - "1" if the sender is an ILE procedure and the name is 256 characters or less
 - "2" if the sender is an ILE procedure and the name is more than 256 characters
 - "3" if the sender is a SLIC program with more than 12 characters
- The next 1 character identifies the sent-to type
 - "0" if the receiver is an OPM program
 - "1" if the receiver is an ILE procedure and the name is 256 characters or less
 - "2" if the receiver is an ILE procedure and the name is more than 256 characters
- The next 6 characters are the microseconds
- The last 10 characters are the name of the user profile that the thread was running under when the message was sent, and is returned if the length of the CL variable is at least 87

*LONG

The long format of the sender information is returned. The long format is 720 characters, with the last 30 characters set to blanks. The following information is returned:

- The first 26 characters identify the sending job
 - Job name (10)
 - User name (10)
 - Job number (6)
- The next 13 characters are the date and time
 - Date (7) (in the format 0yymmdd)
 - Time (6) (in the format hhmmss)

- The next 1 character identifies the sender type
 - "0" if the sender is an OPM program or a SLIC program with 12 characters or less
 - "1" if the sender is an ILE procedure and the name is 256 characters or less
 - "2" if the sender is an ILE procedure and the name is more than 256 characters
 - "3" if the sender is a SLIC program with more than 12 characters
- The next 1 character identifies the sent-to type
 - "0" if the receiver is an OPM program
 - "1" if the receiver is an ILE procedure and the name is 256 characters or less
 - "2" if the receiver is an ILE procedure and the name is more than 256 characters
- The next 12 characters are the sender's program name (for an ILE procedure, this is the bound program name); if the sender type is 3 and the program name is greater than 12 characters in length, the first three characters of this field are less than symbols (<<<) followed by the last nine characters of the program name
- The next 10 characters are the sender's module name (if the sender is not an ILE procedure, this field is set to blanks)
- The next 256 characters are the sender's procedure name (if the sender is not an ILE procedure, this field is set to blanks)
 - For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name, and ending with the inner-most procedure name
 - For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCVPM API can be used to obtain the entire procedure name
- The next 1 character is blank
- The next 4 characters are the number of statement numbers available

Note: A statement number represents a point in the sending program at which the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.

- The next 30 characters return a maximum of 3 statement numbers, 10 characters each
- The next 320 characters return program or procedure information if the message being received was originally sent to a message queue associated with a call stack entry (otherwise, this field is set to blanks)
 - Sent-to program name (10) (for an ILE procedure, this is the bound program name)
 - Sent-to module name (10) (if the sender is not an ILE procedure, this field is set to blanks)
 - Sent-to procedure name (256) (if the sender is not an ILE procedure, this field is set to all blanks)
 - For a nested procedure name, each procedure name is separated by a colon (:) starting with the outer-most procedure name, and ending with the inner-most procedure name.
 - For a procedure name that is longer than 256 characters, three less than symbols (<<<) are returned followed by the last 253 characters of the procedure name; the QMHRCVPM API can be used to obtain the entire procedure name
 - Blanks (10)
 - Number of statements available for the receiving call stack entry (4)

Note: A statement number represents a point at which the sent-to program was suspended (for example, due to a call operation) at the time the message was sent. For programs and non-optimized procedures, this count is always 1. For optimized procedures, this count can be greater than 1, and each statement number represents a point at which the message could have been sent. If it is not possible to return statement numbers, this count will be 0.

- - Statement numbers (30) (a maximum of 3 statement numbers, 10 characters each)
- The next 6 characters are the microseconds
- The last 10 characters are the name of the user profile that the thread was running under when the message was sent

Top

CL var for RTNTYPE (2) (RTNTYPE)

Specifies the name of the control language (CL) variable, if any, that contains the type code for the message received by the program. The variable must be a character variable having a length of 2 positions.

The following values are returned to indicate the message type:

Value	Message Type
01	Completion
02	Diagnostic
04	Information
05	Inquiry
06	Copy
08	Request
10	Request with prompting
14	Notify (exception already handled at time of RCVMSG)
15	Escape (exception already handled at time of RCVMSG)
16	Notify (exception not handled at time of RCVMSG)
17	Escape (exception not handled at time of RCVMSG)
21	Reply, not checked for validity
22	Reply, already checked for validity
23	Reply, message default used
24	Reply, system default used
25	Reply, from System Reply List
26	Reply, from exit program

Top

CL var for ALROPT (9) (ALROPT)

Specifies the name of the control language (CL) variable, if any, used to return the alert option of the message received by the program. The variable must be a character variable 9 positions in length.

Top

CL var for MSGF (10) (MSGF)

Specifies the name of the control language (CL) variable, if any, used to return the message file name of the message received by the program. If the message received is a stored message, the message file name of the file containing the stored message is returned. If the received message is not a stored message, the message file name is returned as blanks. The variable must be a character variable 10 positions in length.

Note: The message file name returned on this parameter is the message file specified or defaulted on the send function, not the overriding message file. If an override was specified when sending the message, the same override should be used when receiving the message.

Top

CL var for MSGFLIB (10) (MSGFLIB)

Specifies the name of the control language (CL) variable, if any, used to return the message file library name of the message received by the program. If the message received is a stored message, the message file library name of the library containing the message file of the stored message is returned. If *LIBL was specified on the send command, *LIBL is returned. If the received message is not a stored message, the message file library name is returned as blanks. The variable must be a character variable 10 positions in length.

Note: The message file library name returned on this parameter is the message file specified or defaulted on the send function, not the overriding message file library. If an override was specified when sending the message, the same override should be used when receiving the message.

Top

CL var for SNDMSGFLIB (10) (SNDMSGFLIB)

Specifies the name of the control language (CL) variable, if any, used to return the message file library name used to send the message. If the message received is a stored message, the message file library name of the library containing the message file of the stored message is returned. If *LIBL was specified on the send command, this parameter would have the actual name of the library. If the received message is not a stored message, the message file library name is returned as blanks. If the received message is a stored message and the original message file is destroyed, the message file library name is returned as blanks. If the received message is not a stored message, this parameter is returned as blanks. The variable must be a character variable of 10 positions in length.

Top

CL var for text CCSID (5 0) (TXTCCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLVL parameters. If a conversion error occurs or if the CCSID you requested the text to be converted to is 65535, the CCSID that the message description or the text for an immediate message is stored in is returned. Otherwise, the CCSID you wanted the text converted to is returned. If you do not want the text converted before it is returned to you but you do want to know the CCSID that the message description or the text for an immediate message is stored in, specify 65535 for the CCSID parameter, and the CCSID is returned in the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred. The variable must be a decimal variable having a length of 5 positions.

CL var for data CCSID (5 0) (DTACCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the replacement data defined as *CCHAR. All other replacement data is not converted before it is returned. If a conversion error occurs or if the CCSID you requested the text to be converted to is 65535, the CCSID message data is returned. If there is no *CCHAR replacement data in the data, 65535 is returned. Otherwise, the CCSID you wanted the text converted to is returned. For immediate messages, 0 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535, a conversion error occurred. The variable must be a decimal variable having a length of 5 positions.

Top

Examples

Example 1: Receiving a Message

```
RCVMSG MSGQ(SMITH) MSGKEY(&KEY) MSG(&WORK)
```

This command receives the message having the message reference key specified by the program variable &KEY from the message queue SMITH. The text of the message is copied into the CL variable &WORK.

Example 2: Receiving a New Message

```
RCVMSG MSGQ(INV) WAIT(120) MSG(&WORK)
```

This command receives a new message from the message queue named INV into the CL variable &WORK. The program waits no more than 120 seconds for the arrival of a new message if there are no new messages in the message queue. If there is more than one new message in the queue, the first message in the queue is the message received by the program.

Example 3: Receiving a Message From a Procedure

```
RCVMSG PGMQ(*SAME CURRENT_MONTH_TOTALS) MSGTYPE(*EXCP)
      RMV(*KEEPEXCP) MSGID(&MID) MSG(&MTEXT)
```

This command receives an exception message from the procedure CURRENT_MONTH_TOTALS. Since the specified name is more than 10 characters, the system does not search for any programs. If the message is an unhandled exception, the message is left on the call message queue as a new message and the exception is not handled by the RCVMSG command. The message ID is returned in the CL variable &MID and the message text in the CL variable &MTEXT. To handle the exception and remove the message, run the following RCVMSG command:

```
RCVMSG PGMQ(*SAME CURRENT_MONTH_TOTALS) MSGTYPE(*EXCP)
      RMV(*YES)
```

Example 4: Receiving a Message from a Program or Procedure

```
RCVMSG  PGMQ(*SAME TARGETPGM) MSGTYPE(*EXCP) RMV(*NO)
        MSGID(&MID)  MSG(&MTEXT)
```

This command receives an exception message from the message queue of the program or procedure named TARGETPGM. Since the specified name is only 9 characters, the system searches both programs and procedures. Because RMV(*NO) is specified, if the message is an unhandled exception, the exception is handled by the RCVMSG command. The message is left on the message queue as an old message.

Example 5: Receiving a Message Using Qualifiers

```
RCVMSG  PGMQ(*SAME PRINT_RPT_FMT1 DEPTRPTS AREARPTS)
        MSGTYPE(*EXCP)  RMV(*YES)
        MSGID(&MID)  MSG(&MTEXT)
```

This command receives an exception message from the message queue of the procedure named PRINT_RPT_FMT1. The procedure must have been compiled into the module DEPTRPTS and have been bound into the bound program AREARPTS. Since RMV(*YES) is specified, the exception is handled if the exception message is for an unhandled exception. The message is always removed from the message queue.

Example 6: Receiving a Message Using a Partial Procedure Name

```
RCVMSG  PGMQ(*SAME 'HANDLE_FORM_NUM>>>') MSGID(&MID)
        MSG(&MTEXT)
```

This command receives a new message from the most recent procedure whose name begins with HANDLE_FORM_NUM.

Top

Error messages

*ESCAPE Messages

CPF2401

Not authorized to library &1.

CPF2403

Message queue &1 in &2 not found.

CPF2407

Message file &1 in &2 not found.

CPF2408

Not authorized to message queue &1.

CPF2411

Not authorized to message file &1 in &2.

CPF241C

Variable for SENDER parameter is too small.

CPF2410

Message key not found in message queue &1.

CPF2415
End of requests.

CPF2422
Reply not valid.

CPF2423
Variable specified in SENDER parameter less than &1 bytes.

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

CPF2449
Message that should be a reply, is not a reply.

CPF2450
Work station message queue &1 not allocated to job.

CPF2451
Message queue &1 is allocated to another job.

CPF247A
Call stack entry not found.

CPF247E
CCSID &1 is not valid.

CPF2471
Length of field not valid.

CPF2477
Message queue &1 currently in use.

CPF2479
Call stack entry not found.

CPF2482
Message type &1 not valid.

CPF24A3
Value for call stack counter parameter not valid.

CPF24A8
Value for wait time not valid.

CPF24B3
Message type &1 not valid.

CPF2531
Message file &1 in &2 damaged for &3.

CPF2532
Job message queue is damaged. Job log ended.

CPF2548
Damage to message file &1 in &2.

CPF2551
Message key and message type combination not valid.

CPF36F7
Message queue QSYSOPR is allocated to another job.

CPF8127
&8 damage on message queue &4 in &9. VLIC log-&7.

CPF8176

Message queue for device description &4 damaged.

[Top](#)

Receive Network File (RCVNETF)

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

Parameters
Examples
Error messages

The Receive Network File (RCVNETF) command receives a network file and copies the records into a physical database file or a save file. Once the file has been received, it is removed from the queue of network files.

If the original file is a save file, it must be received into a save file. Before a file can be received, the file specified by the TOFILE parameter must already exist.

When a source physical file is sent, the source sequence number and change date in positions 1 through 12 of the record are sent with the file. These are kept if the file is received into a source physical file, and are truncated if the file is received into a nonsource physical file. When a file that was originally a nonsource physical file is received into a source physical file, the source sequence numbers are created and placed in front of the records.

If the file is a physical file, the record length of the to-file must be at least as large as the record length of the original file. If the record length of the to-file is larger than that of the original file, the records are padded to the end with the default record value for the to-file.

This command does not perform any CCSID translation on the contents of the file. However, the user ID and address of both the recipient and the originator are translated from the multinational character set 697/500 to the current job CCSID.

Restrictions:

1. A user with security officer authority can receive the files sent to any user. Users with other than security officer authority can receive only files sent to them or to their group profile.
2. The user must have read authority to the library containing the to-file, and use and add authority to the to-file. The following additional authority may be required:
 - Object management authority, if a member is added to the file.
 - Object management authority and delete authority, if a save file or existing physical file member is cleared.

Top

Parameters

Keyword	Description	Choices	Notes
FROMFILE	From file	<i>Character value</i>	Required, Positional 1
TOFILE	To file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: To file	<i>Name, *FROMFILE</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
FROMMBR	Member to be received	<i>Character value, *ONLY</i>	Optional, Positional 3
TOMBR	To member	<i>Name, *FROMMBR, *FIRST</i>	Optional, Positional 4

Keyword	Description	Choices	Notes
MBROPT	Replace or add records	<u>*REPLACE</u> , *ADD	Optional
NBR	File number	<i>Integer</i> , <u>*LAST</u> , *ONLY, *FIRST	Optional
USER	User	<i>Name</i> , <u>*CURRENT</u>	Optional
FROMTYPE	From file type	<u>*NETFILE</u> , *SRC	Optional

Top

From file (FROMFILE)

Specifies the name of the file that is received. This is the name of the file on the sending system.

This is a required parameter.

Top

To file (TOFILE)

Specifies the name of the file into which the network file is received. Overrides to this file are ignored.

The possible values are:

*FROMFILE

The network file is received into a file of the same name as the file sent.

file-name

Specify the name and library of the receiving file.

The possible library values are:

*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.

library-name

Specify the library where the file is located.

Top

Member to be received (FROMMBR)

Specifies the name of the file member that is received.

The possible values are:

*ONLY

Only one member is received for this file. If *ONLY is specified on the **File number** prompt (NBR parameter), only one member per file is on the arrived file queue.

member-name

Specify the name of the member that is received. A member name cannot be specified if the file is a save file.

Top

To member (TOMBR)

Specifies the database file member that receives the data.

The possible values are:

*FROMMBR

The data is received into a member with the same name as the member specified on the **Member to be received** prompt (FROMMBR parameter).

*FIRST

The first member in the file receives the output.

member-name

Specify the name of the member that receives the records. A member name cannot be specified if the file is a save file.

Top

Replace or add records (MBROPT)

Specifies whether the new records replace or are added to the existing records.

The possible values are:

*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

File number (NBR)

Specifies the number of the file member that is received. This number is used to identify the member that is received when there is more than one member of the same name available for the file.

The possible values are:

*LAST

The last network file member with the specified member name is received. The last member is determined as the last member to arrive at your system.

Note: The file member that arrived last at the user's system may not have been the last one sent by the sending user. The network does not guarantee the arrival sequence of separately sent files.

*ONLY

Only one file member of the specified file name is received.

member-number

Specify the number of the member that is received.

Top

User (USER)

Specifies the user to whom the file was sent.

The possible values are:

*CURRENT

The files sent to the current user are received.

user-name

Specify the name of the user to whom the files were sent. Only users with security officer authority can specify a name other than their own or their group profile.

Top

From file type (FROMTYPE)

Specifies the type of file that is received. This option should be used mainly when the file is a System i5 or System/38 source file which was sent by a System/370 VM or MVS user. Since VM or MVS cannot identify whether the file is a source file, you can indicate whether the file is a source file or a non-source file.

The possible values are:

*NETFILE

The network file type is used to determine whether file type conversion is needed.

If the file is a non-source file and is:

- Received into a non-source file, the file is received unchanged.
- Received into a source file, the sequence numbers and date fields are added.

If the file is a source file and is:

- Received into a non-source file, the sequence numbers and date fields are removed (the first 12 bytes of each record).
- Received into a source file, the file is received unchanged.

***SRC** The file being received is a source file. The sequence numbers and date fields are in the file. If the file is received into another source file, the sequence numbers and date fields are not added to the file being received. If the file is received into a non-source file, the sequence numbers and date fields are removed (the first 12 bytes of each record).

Note: *SRC must **not** be specified if the network file does not contain sequence numbers and date fields in the first 12 bytes of each record.

Top

Examples

Example 1: Receiving a Member

```
RCVNETF FROMFILE(FILEA) TOFILE(FILEB/FILEA)
        FROMMBR(PAYROLL)
```

This command receives member PAYROLL of file FILEA into member PAYROLL of file FILEA in library FILEB. If there is an existing member of that name, the records in the member are replaced. If multiple members of that name are available, the last one to arrive at the destination system is received.

Example 2: Receiving a Network File

```
RCVNETF FROMFILE(PERSONNEL) NBR(*LAST) USER(USR1)
```

This command receives a network file named PERSONNEL, which was sent to user USR1, into a file with the same name. Because the FROMMBR parameter is not specified, there must be only one member name available for this file. Because USR1 is specified, only someone with a user profile of USR1, someone with a group profile of USR1, or someone with security officer authority can use this command.

Example 3: Receiving a Source File

```
RCVNETF  FROMFILE(FILEA)  TOFILE(FILEB/FILEA)
          FROMMBR(PAYROLL) FROMTYPE(*SRC)
```

This command specifies that the file being received is a source file and the sequence numbers and date fields are not added to the file being received.

Top

Error messages

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2207

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

CPF4101

File &2 in library &3 not found or inline data file missing.

CPF5715

File &1 in library &2 not found.

CPF8057

File &1 in &2 not a physical file or save file.

CPF8059

Member name not allowed for save file.

CPF8060

No files compare to the specified selection.

CPF8062

Record length of network file larger than receiving file.

CPF8063

Cannot assign necessary resource.

CPF8070

Not allowed to process files for user &1.

CPF8077

More than one file with same name found. See previously displayed messages.

CPF8080

MBROPT(*ADD) not allowed for save file.

CPF8081

File &5 member &6 number &7 already processed.

CPF8082

Cannot get network file &5 member &6 number &7.

- CPF9005**
System resource required to complete this request not available.
- CPF9006**
User not enrolled in system distribution directory.
- CPF9803**
Cannot allocate object &2 in library &3.
- CPF9807**
One or more libraries in library list deleted.
- CPF9808**
Cannot allocate one or more libraries on library list.
- CPF9810**
Library &1 not found.
- CPF9812**
File &1 in library &2 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9830**
Cannot assign library &1.
- CPF9843**
Object &1 in library &3 type &2 cannot be accessed.
- CPF9845**
Error occurred while opening file &1.
- CPF9846**
Error while processing file &1 in library &2.
- CPF9847**
Error occurred while closing file &1 in library &2.
- CPF9848**
Cannot open file &1 in library &2 member &3.
- CPF9849**
Error while processing file &1 in library &2 member &3.

Top

Receive TIE File (RCVTIEF)

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 Receive Technical Information Exchange File (RCVTIEF) command allows you to receive files transmitted from the remote support network.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i> , *LIBL, *CURLIB	Required, Positional 1
TYPE	File type	* <u>ALL</u> , *OTHER, *SAVE	Optional
OUTPUT	Output	* <u>NONE</u> , *PRINT	Optional
MAXRCDS	Maximum records	<i>Integer</i> , <u>10000</u> , *NOMAX	Optional

[Top](#)

Library (LIB)

Specifies the name and library where the files are stored.

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.

[Top](#)

File type (TYPE)

Specifies the types of files that are received.

***ALL** All available files are received.

***OTHER**

Files with unspecified contents are received.

***SAVF** Save files are received.

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.

***NONE**

The list of received files is not printed.

***PRINT**

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

Top

Maximum records (MAXRCDS)

Specifies the maximum size (number of records) of any file that can be received. If one or more files exceeds the maximum size, no files are received.

10000 The maximum size of file that can be received is 10000 records.

***NOMAX**

The system maximum of 500000 records is used.

number

Specify the maximum size (number of records) of file that can be received.

Top

Examples

```
RCVTIEF LIB(MAIL) TYPE(*OPEN) OUTPUT(*PRINT) MAXRCDS(1000)
```

This command receives from TIE all OPEN files (any file except a save file). A list of the received files is printed. If any of the received files are larger than 1000 records, the RCVTIEF command fails. If all OPEN files are received successfully, they are removed from the mailbox.

Top

Error messages

None

Top

Remove Directory (RD)

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

Parameters
Examples
Error messages

The Remove Directory (RD) command removes a specified directory from the system after all objects in the directory have been unlinked and the directory is no longer in use. If a directory to be removed contains objects, this command optionally unlinks all of the objects and then deletes the directory. If the user does not have the authority to unlink every object in the directory, only those objects for which the user has the authority are unlinked. When an object cannot be unlinked, the directory and all objects in the directory that cannot be unlinked are not removed.

This command can also be used to remove a directory tree, where the specified directory, its contents and the contents of all of its subdirectories are removed. If SUBTREE(*ALL) or RMVLNK(*YES) is specified, the command will attempt to remove as many objects as possible within the subtree. A diagnostic message will be sent for each object that cannot be removed. When all of the objects have been attempted, an escape message will be sent if there were any errors. If all of the objects have been removed with no errors, then a completion message will be sent.

If a symbolic link object is encountered, either specified in the **Directory (DIR)** parameter or encountered in the processing of a subtree, the symbolic link will not be followed.

Note: The symbolic link will be removed if SUBTREE(*ALL) was specified on the command.

This command is an alias for the Remove Directory (RMVDIR) command and can also be issued using the following alternative command names:

- RMDIR
- RMVDIR

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- In the "root" (/), QOpenSys and user-defined file systems, the user must have object existence (*OBJEXIST) authority for the specified directory, and *OBJEXIST authority for every object in it. If the user does not have *OBJEXIST authority for one or more objects in the directory, those objects are not unlinked and the directory is not removed.
- In the "root" (/), QOpenSys, and user-defined file systems, the user must have write and execute (*WX) authority to all of the non-empty directory objects to remove, including the parent directory.
- In the QDLS file system, the user must have all (*ALL) authority to the directory and execute (*X) authority to its parent directory.
- The user must have execute (*X) authority to the prefix directory.
- See the System i Security Reference, SC41-5302 book for the authority requirements for other file systems.
- A user cannot remove an object link within a "root" (/), QOpenSys, or user-defined file system directory that has the "restricted rename and unlink" attribute set on (this attribute is equivalent to the S_ISVTX mode bit) unless one or more of the following are true:
 - The user is the owner of the object link to be removed.
 - The user is the owner of the parent directory of the object link to be removed.

- The user has all object (*ALLOBJ) special authority.
- A directory cannot be removed if it is the current directory for a job.
- This command cannot be used to delete reserved directories or reserved libraries.
- When an object is open in QSYS.LIB, independent ASP QSYS.LIB, or QDLS, the object cannot be unlinked. When an object is open in QOpenSys or the "root" (/) file system, the object is successfully unlinked, and the object is deleted when it is closed.
- When doing subtree processing, the user must have read (*R) and execute (*X) authorities to the path name and all subdirectories within that path.
- If SUBTREE(*ALL) is specified, the restrictions and file system differences listed for Remove link (RMVLNK) command would also apply to this usage of this command.

Top

Parameters

Keyword	Description	Choices	Notes
DIR	Directory	<i>Path name</i>	Required, Positional 1
SUBTREE	Directory subtree	*NONE, *ALL	Optional
RMVLNK	Remove link	*NO, *YES	Optional

Top

Directory (DIR)

Specifies the path name of the directory or a pattern to match the path name or names of directories to be removed.

The object path name can be either a simple name or a name that is qualified with the name of the directory in which the object is located. A pattern can be specified in the last part of the path name. An asterisk (*) matches any number of characters and a question mark (?) matches a single character. If the path name is qualified or contains a pattern, it must be enclosed in apostrophes.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Top

Directory subtree (SUBTREE)

Specifies whether or not to remove all of the object links within the subtree if the object specified by the **Directory (DIR)** parameter is a directory.

You can specify either the **Directory subtree (SUBTREE)** parameter or the **Remove link (RMVLNK)** parameter, but not both.

*NONE

The directory specified by DIR is removed only if it is empty. A directory may contain entries for the directory (.) and for the parent directory (..) and still be treated as an empty directory.

***ALL** The object links within the directory specified by DIR are removed. The directory's contents, as well as the contents of all of its subdirectories will be removed.

Note: Pattern matching on the DIR parameter only applies to the first level objects. If the first level object is a directory, the pattern matching does not apply to its contents or the contents of its subdirectories.

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Remove link (RMVLNK)

Specifies whether to unlink all objects in a directory or not allow the directory to be deleted if it contains objects.

You can specify either the **Directory subtree (SUBTREE)** parameter or the **Remove link (RMVLNK)** parameter, but not both.

***NO** Only an empty directory is removed. A directory may contain entries for the directory (.) and for the parent directory (..) and still be treated as an empty directory.

***YES** All object links within the specified directory are removed. If the file system that contains the directory does not support removal of links in the directory and the directory contains object links, error message CPFA0AC "Directory contains objects. Directory is &1." will be sent.

Note: The QDLS, QSYS.LIB and independent ASP QSYS.LIB file systems support removal of links using this parameter specification. For all other file systems, use the SUBTREE(*ALL) parameter specification.

Note: Pattern matching on the DIR parameter only applies to the first level objects. If the first level object is a directory, the pattern matching does not apply to its contents or the contents of its subdirectories.

Top

Examples

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

Example 1: Removing a Directory and Its Objects

```
RMVDIR DIR('/QSYS.LIB/JULIO.LIB') RMVLNK(*YES)
```

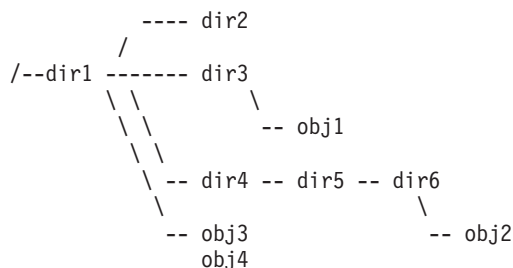
This command removes directory /QSYS.LIB/JULIO.LIB, after all of its objects have been unlinked. If directory /QSYS.LIB/JULIO.LIB contains objects, all of the objects are unlinked and directory /QSYS.LIB/JULIO.LIB is then removed.

Example 2: Removing an Empty Directory

```
RMVDIR DIR('/QSYS.LIB/EMPTY.LIB') RMVLNK(*NO)
```

This command will remove the empty directory /QSYS.LIB/EMPTY.LIB.

The examples below assume the following directory structure:



Example 3: Removing a Directory and Its Objects Using Subtree Processing

```
RMVDIR DIR('/dir1/dir4') SUBTREE(*ALL)
```

The command removes the **dir4** directory tree. This includes removing the **obj2,dir6,dir5,** and **dir4** objects.

Example 4: Removing a Directory With No Subtree Processing

```
RMVDIR DIR('/dir1/dir2') SUBTREE(*NONE)
```

This command will remove the directory /**dir1/dir2**, because it is empty.

Example 5: Removing a Group of Directories Using Pattern Matching and Subtree Processing

```
RMVDIR DIR('/dir1/d*') SUBTREE(*ALL)
```

Objects **obj1** and **obj2**, and directories **dir2, dir3, dir5, dir6,** and **dir4** will be removed. Objects **obj3** and **obj4** will **not** be removed because they do not match the specified name pattern and are not located in a directory that matches the name pattern.

Top

Error messages

*ESCAPE Messages

CPFA085
Home directory not found for user &1.

CPFA093
Name matching pattern not found.

CPFA09C
Not authorized to object. Object is &1.

CPFA09D
Error occurred in program &1.

CPFA09E
Object in use. Object is &1.

CPFA0A1
An input or output error occurred.

CPFA0A3
Path name resolution causes looping.

CPFA0A7
Path name too long.

CPFA0A9
Object not found. Object is &1.

CPFA0AA
Error occurred while attempting to obtain space.

CPFA0AB
Operation failed for object. Object is &1.

CPFA0AC
Directory contains objects. Directory is &1.

CPFA0AD
Function not supported by file system.

CPFA0B1
Requested operation not allowed. Access problem.

CPFA0B2
No objects satisfy request.

CPFA0B7
&3 object links removed. &4 object links not removed.

Top

Rename Object (REN)

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

Parameters
Examples
Error messages

The Rename Object (REN) command changes the name of an object in a directory.

This command is an alias for the Rename Object (RNM) command and can also be issued using the following alternative command names:

- RNM

For more information about integrated file system commands, see the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

- This command works on only one object. If a pattern is specified on the **Object (OBJ)** parameter and more than one object matches the pattern, the user can select the object from a list in an interactive job. If this is a batch job, the command fails with error message CPFA08E, "More than one name matches pattern."
- When renaming an object in the "root" (/), QOpenSys or user-defined file systems, the user must have object management (*OBJMGT) authority to the object to be renamed, and write and execute (*WX) authority to the directory that contains the object. If the object to be renamed is a directory, the user must also have write (*W) authority to the directory.
- The user must have execute (*X) authority to each directory in the path.
- A user cannot rename an object within a "root" (/), QOpenSys, or user-defined file system directory that has the "restricted rename and unlink" attribute set on (this attribute is equivalent to the S_ISVTX mode bit) unless one or more of the following are true:
 - The user is the owner of the object.
 - The user is the owner of the directory.
 - The user has all object (*ALLOBJ) special authority.
- The authority requirements and restrictions from the existing Rename Object (RNMOBJ) command and Rename Document Library Object (RNMDLO) command apply to objects in the QSYS.LIB, independent ASP QSYS.LIB, and QDLS file systems.
- In the QSYS.LIB and independent ASP QSYS.LIB file systems, the new name must contain the same object type suffix.
- Some objects cannot be renamed. An error is returned if an attempt is made to rename these objects.
- The file cannot be renamed if the file is a DataLink column in an SQL table and where a row in that SQL table references this file.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
NEWOBJ	New object	<i>Character value</i>	Required, Positional 2

Object (OBJ)

Specifies the path name of the object to be renamed.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

New object (NEWOBJ)

Specifies the new name of the object to be renamed. This name cannot contain any directory qualifiers and is in the same directory containing the existing object.

For more information on specifying path names, refer to "Object naming rules" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Examples

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

Example 1: Renaming an Object Link

```
RNM  OBJ('DECEMBER-1994-MONTHLY-PAYROLL-FILE')
      NEWOBJ('JANUARY-1995-MONTHLY-PAYROLL-FILE')
```

This command renames a file named DECEMBER-1994-MONTHLY-PAYROLL-FILE to a file named JANUARY-1995-MONTHLY-PAYROLL-FILE.

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA08E
More than one name matches pattern.

CPFA093
Name matching pattern not found.

CPFA09C
Not authorized to object. Object is &1.

CPFA09D
Error occurred in program &1.

CPFA0A1
An input or output error occurred.

CPFA0A3
Path name resolution causes looping.

CPFA0A6
Number of links exceeds maximum allowed for the file system.

CPFA0A7
Path name too long.

CPFA0A9
Object not found. Object is &1.

CPFA0AA
Error occurred while attempting to obtain space.

CPFA0AB
Operation failed for object. Object is &1.

CPFA0B1
Requested operation not allowed. Access problem.

CPFA0B2
No objects satisfy request.

CPFA0B4
NEWOBJ parameter cannot start with a slash.

CPFA0B5
The NEWOBJ parameter cannot contain path.

CPFA0C4
Object not a file. Object is &1.

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
Distributed Relational Database Architecture
Domino
DRDA
IBM
Infoprint
InfoWindow
i5/OS
iSeries
Integrated Language Environment
Lotus
LPDA
OfficeVision
Print Services Facility
RPG/400
System i
System x
SystemView
System/36
TCS
Tivoli
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.



Printed in USA