



iSeries

Operating System/400 Commands
Starting with ENDMSF (End Mail Server Framework)

Version 5 Release 3





@server

iSeries

Operating System/400 Commands
Starting with ENDMSF (End Mail Server Framework)

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 495.

First Edition (May 2004)

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

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

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

Contents

End Mail Server Framework (ENDMSF)	1	Examples	19
Parameters	1	Error messages	20
How to end (OPTION)	1		
Controlled end delay time (DELAY)	1	End Performance Trace (ENDPFRTRC)	21
Examples	2	Parameters	21
Error messages	2	Dump the trace (DMPTRC)	21
		Member (MBR)	21
End NFS Server (ENDNFSSVR)	3	Library (LIB)	22
Parameters	3	Text 'description' (TEXT)	22
Server daemon (SERVER)	4	Examples	22
Timeout for end of daemon (ENDJOBTIMO).	4	Error messages	22
Examples	4		
Error messages	4	End Program (ENDPGM)	23
		Parameters	23
End NetWare Connection (ENDNTWCNN)	7	Examples	23
Parameters	7	Error messages	23
Server (SERVER)	7		
Connection number (C>NNNBR)	7	End Program Export List (ENDPGMEXP)	25
Examples	7	Parameters	25
Error messages	7	Examples	25
		Error messages	25
End Network Interface Recovery (ENDNWIRCY)	9		
Parameters	9	End Program Profiling (ENDPGMPRF)	27
Network interface description (NWI)	9	Parameters	27
Examples	9	Examples	27
Error messages	9	Error messages	27
End Pass-Through (ENDPASTHR)	11	End Prestart Jobs (ENDPJ)	29
Parameters	11	Parameters	29
Job log (LOG)	11	Subsystem (SBS)	29
Examples	12	Program (PGM)	30
Error messages	12	How to end (OPTION)	30
		Controlled end delay time (DELAY)	30
End Performance Explorer (ENDPEX)	13	Delete spooled files (SPLFILE)	31
Parameters	13	Maximum log entries (LOGLMT)	31
Session ID (SSNID)	14	Examples	31
Option (OPTION)	14	Error messages	32
Data option (DTAOPT)	15		
Data library (DTALIB)	15	End Printer Emulation (ENDPRTEML)	33
Data member (DTAMBR)	15	Parameters	33
Management collection (MGTCOL)	16	Emulation device (EMLDEV)	33
Replace data (RPLDTA)	16	Emulation location (EMLLOC)	34
Number of threads (NBRTHD)	16	Print device (PRTDEV)	34
Text 'description' (TEXT)	17	Examples	34
Examples	17	Error messages	34
Error messages	17		
		End Receive (ENDRCV)	35
End Performance Collection (ENDPFCOL)	19	Parameters	35
Parameters	19	Display device (DEV)	35
Force end (FRCCOLEND)	19	Open file identifier (OPNID)	35
		Examples	36
		Error messages	36

(ENDRDBRQS)	37
Parameters	37
Examples	37
Error messages	37
End Reader (ENDRDR)	39
Parameters	39
Reader (RDR).	39
When to end reader (OPTION).	39
Examples	40
Error messages	40
End Remote Support (ENDRMTSPT)	41
Parameters	41
Delete library (DLTLIB)	41
How to end (OPTION)	41
Examples	42
Error messages	42
End RPC Binder Daemon (ENDRPCBIND)	43
Parameters	43
Examples	43
Error messages	43
End Request (ENDRQS)	45
Parameters	45
Request level (RQSLVL)	45
Examples	46
Error messages	46
End S/36 Session (ENDS36)	47
Parameters	47
Examples	47
Error messages	47
End Subsystem (ENDSBS)	49
Parameters	49
Subsystem (SBS).	50
How to end (OPTION)	50
Controlled end delay time (DELAY)	51
End subsystem option (ENDSBSOPT).	51
Batch time limit (BCHTMLMT)	52
Examples	52
Error messages	52
End Select (ENDSELECT).	55
Parameters	55
Examples	55
Error messages	55
End Service Agent (ENDSRVAGT)	57
Parameters	57
Type (TYPE)	57
Examples	57
Error messages	57
End Service Job (ENDSRVJOB)	59

Parameters	59
Examples	59
Error messages	59
End System (ENDSYS)	61
Parameters	61
How to end (OPTION)	61
Controlled end delay time (DELAY)	62
End subsystem option (ENDSBSOPT).	62
Confirm (CONFIRM)	63
Examples	63
Error messages	63
End TCP/IP (ENDTCP)	65
Parameters	65
How to end (OPTION)	66
Controlled end delay time (DELAY)	66
End application servers (ENDSVR)	66
Examples	67
Error messages	67
End TCP/IP Abnormal (ENDTCPABN)	69
Parameters	69
Examples	69
Error messages	69
End TCP/IP Connection (ENDTCPCNN)	71
Parameters	71
Protocol (PROTOCOL).	71
Local internet address (LCLINTNETA)	71
Local port (LCLPORT).	72
Remote internet address (RMTINTNETA)	72
Remote port (RMTPORT).	72
Examples	72
Error messages	73
End TCP/IP Interface (ENDTCPIFC)	75
Warning: Temporary Level 2 Header	75
Parameters	75
Internet address (INTNETADR).	75
Examples	76
Error messages	76
End Point-to-Point TCP/IP (ENDTCPPTP).	77
Parameters	77
Configuration profile (CFGPRF)	77
Operating mode (OPRMODE)	78
Examples	78
Error messages	78
End TCP/IP Server (ENDTCPSVR)	81
Parameters	81
Server application (SERVER).	81
HTTP server (HTTPSVR)	84
DNS server (DNSSVR).	84
TCM server (TCMSVR)	84
ASFTOMCAT server (TOMCATSVR)	84
Examples	85

Error messages	85
End TIE Session (ENDTISSN)	87
Parameters	87
Examples	87
Error messages	87
End Trace (ENDTRC).	89
Parameters	89
Session ID (SSNID).	89
Data option (DTAOPT)	90
Data library (DTALIB).	90
Replace data (RPLDTA)	90
Print trace data (PRTTRC)	90
Examples	91
Error messages	91
End Trap Manager (ENDTRPMGR)	93
Parameters	93
Examples	93
Error messages	93
End Writer (ENDWTR)	95
Parameters	95
Writer (WTR).	95
When to end writer (OPTION)	95
Examples	96
Error messages	96
Remove Link (ERASE)	97
Parameters	98
Object link (OBJLNK)	98
Examples	98
Error messages	99
Export a Program Symbol (EXPORT)	101
Parameters	101
Exported symbol name (SYMBOL)	101
Examples	101
Error messages	101
Change NFS Export (EXPORTFS).	103
Parameters	103
NFS export options (OPTIONS)	104
Directory (DIR).	107
Host name (HOSTOPT)	107
Examples	108
Error messages	109
Extract Program Information (EXTPGMINF)	111
Error messages for EXTPGMINF	111
Parameters	111
Program (PGM).	111
File to receive information (FILE).	112
Extract record options (OPTION)	112
Create the file (CRTFILE)	113
Library name to record (RECLIB).	113
Consistency Check (CHECK)	113

Examples	113
Error messages	113
File Document (FILDOC).	115
Parameters	115
Information to be filed (TYPE).	116
To document (TODOC)	117
To folder (TOFLR)	117
Sensitivity (SENSITIV)	117
User authority (USRAUT)	118
Authorization list (AUTL)	119
Access code (ACC)	119
Allow replacement (ALWRPL).	119
Profile file (IDPFILE)	119
Profile member (IDPMBR)	120
User identifier (USRID)	120
Document file (DOCFILE)	121
Document member (DOCMBR)	121
Distribution identifier (DSTID)	121
Distribution ID extension (DSTIDEXN).	122
Keep in mail log (KEEP).	122
Document type (DOCTYPE)	122
System code (SYSCOD)	123
Document description (DOCD)	123
Author (AUTHOR)	124
Document class (DOCCLS).	124
Keyword (KWD)	124
Subject (SUBJECT).	125
Document date (DOCDATE)	125
File cabinet location (FILCAB).	125
Copy list (CPYLST)	126
Expiration date (EXPDATE)	126
Reference (REFERENCE)	126
Action due date (ACTDATE)	127
Document status (STATUS).	127
Completion date (CMPDATE)	127
Project (PROJECT).	128
Document character identifier (DOCCHRID)	128
Language ID (DOCLANGID)	128
Country or region ID (DOCCNTRYID)	129
Personal (PERSONAL)	129
Distribution expiry indicator (DSTEXPDATE).	129
Command character identifier (CMDCHRID).	130
Examples	131
Error messages	131
Format Data (FMTDTA)	133
Parameters	133
Input file (INFILE)	133
Output file (OUTFILE)	134
Source file (SRCFILE).	134
Source member (SRCMBR)	135
Print file (PRTFILE)	135
Options: (OPTION)	135
Program date: (PGMDATE).	136
Examples	136
Error messages	136
Generate Message Catalog (GENCAT)	137
Parameters	137

Message catalog name (CLGFILE)	137
Source file path name (SRCFILE)	138
Text 'description' (TEXT)	138
Message catalog CCSID (CLGCCSID)	138
Source file CCSID (SRCCCSID)	138
Public authority for data (DTAAUT).	139
Public authority for object (OBJAUT)	140
Examples.	140
Examples for MRGMSGCLG	140
Error messages	140

Generate Command Documentation (GENCMDDOC) 141

Parameters	141
Command (CMD)	142
To directory (TODIR)	142
To stream file (TOSTMF)	142
Replace file (REPLACE)	143
Generation options (GENOPT)	143
Examples.	144
Error messages	144

Go to Menu (GO) 147

Parameters	147
Menu (MENU)	147
Return point (RTNPNT)	148
Examples.	149
Error messages	149

Go To (GOTO) 151

Parameters	151
Command label (CMDLBL)	151
Examples.	151
Error messages	152

Grant Access Code Authority (GRTACCAUT) 153

Parameters	153
Document access code (ACC)	153
User profile (USER)	154
Reference user profile (REFUSER)	154
Examples.	154
Error messages	154

Grant Object Authority (GRTOBJAUT) 157

Parameters	158
Object (OBJ)	159
Object type (OBJTYPE)	159
ASP device (ASPDEV)	160
Users (USER)	160
Authority (AUT)	160
Authorization list (AUTL)	162
Reference object (REFOBJ)	162
Reference object type (REFOBJTYPE)	162
Reference ASP device (REFASPDEV)	163
Replace authority (REPLACE)	163
Examples.	163
Error messages	164

Grant User Authority (GRTUSRAUT) 167

Parameters	167
User (USER).	167
Referenced user (REFUSER)	168
Examples.	168
Error messages	168

Grant User Permission (GRTUSRPMN) 169

Parameters	169
To user profile (TOUSER)	169
For user profile (FORUSER)	169
Examples.	170
Error messages	170

Grant Workstation Object Aut (GRTWSOAU). 171

Parameters	171
Workstation object type (WSOTYPE).	172
Users (USER)	173
Authority (AUT)	174
Authorization list (AUTL)	175
Reference workstation object (REFWSO)	175
Examples.	175
Error messages	176

Hold Communications Device (HLDCMNDEV) 177

Parameters	177
Device (DEV)	177
Option (OPTION)	178
Examples.	178
Error messages	178

Hold Distribution Queue (HLDDSTQ) 181

Parameters	181
Distribution queue (DSTQ)	181
Priority (PTY)	181
Examples.	182
Error messages	182

Hold Job (HLDJOB). 185

Parameters	185
Job name (JOB).	185
Hold spooled files (SPLFILE)	186
Duplicate job option (DUPJOB OPT)	186
Examples.	186
Error messages	187

Hold Job Queue (HLDJOBQ). 189

Parameters	189
Job queue (JOBQ)	189
Examples.	189
Error messages	190

Hold Job Schedule Entry (HLDJOBSCDE) 191

Parameters	191
Job name (JOB).	191

Entry number (ENTRYNBR)	192
Examples	192
Error messages	192
Hold Output Queue (HLDOUTQ)	195
Parameters	195
Output queue (OUTQ)	195
Examples	195
Error messages	196
Hold Reader (HLDRDR)	197
Parameters	197
Reader (RDR)	197
Examples	197
Error messages	197
Hold Spooled File (HLDSPLF)	199
Parameters	199
Spooled file (FILE)	200
Job name (JOB)	200
Spooled file number (SPLNBR)	200
Job system name (JOBSYSNAME)	201
Spooled file created (CRTDATE)	201
Select files for (SELECT)	202
ASP device (ASPDEV)	203
When to hold file (OPTION)	203
Examples	203
Error messages	204
Hold Writer (HLDWTR)	207
Parameters	207
Writer (WTR)	207
When to hold writer (OPTION)	207
Examples	208
Error messages	208
If (IF)	209
Parameters	210
Condition (COND)	210
Command (THEN)	210
Examples	211
Error messages	212
Install Program Temporary Fix (INSPTF)	213
Parameters	213
Product description (LICPGM)	214
Device (DEV)	214
PTF apply type (INSTYP)	214
PTF omit list (OMIT)	215
HIPER PTFs only (HIPER)	215
End of media option (ENDOPT)	216
Restart type (RESTART)	216
Prompt for media (PMTMED)	217
Examples	217
Error messages	217
Initialize Diskette (INZDKT)	219
Parameters	219

Diskette device (DEV)	220
New volume identifier (NEWVOL)	220
New owner identifier (NEWOWNID)	220
Diskette format (FMT)	220
Sector size (SCTSIZ)	221
Check for active files (CHECK)	222
Code (CODE)	222
Examples	222
Error messages	222
Initialize DLFM (INZDLFM)	225
Parameters	225
Clear existing databases (CLEARDB)	225
Examples	225
Error messages	225
Initialize Distribution Queue (INZDSTQ)	227
Parameters	227
Distribution queue (DSTQ)	228
Clear queue entries (CLEAR)	228
Examples	228
Error messages	229
Initialize Optical (INZOPT)	231
Parameters	231
Volume identifier (VOL)	231
New volume identifier (NEWVOL)	232
Device (DEV)	232
Volume full threshold (THRESHOLD)	232
Check for an active volume (CHECK)	232
End of media option (ENDOPT)	233
Clear (CLEAR)	233
Volume type (TEXT)	233
Volume type (TYPE)	233
Coded character set ID (CCSID)	234
Media format (MEDFMT)	234
Examples	234
Error messages	235
Initialize Client Access/400 (INZPCS)	237
Error messages for INZPCS	237
Parameters	237
Keyboard type (KBDTYPE)	238
ASCII code page number (ASCII)	240
EBCDIC code page number (EBCDIC)	240
Language feature code (LANGUAGE)	241
Examples	242
Error messages	243
Initialize Physical File Mbr (INZPFM)	245
Parameters	245
Physical file (FILE)	245
Member (MBR)	246
Initialize records as (RECORDS)	246
Total number of records (TOTRCDS)	246
Examples	246
Error messages	247

Initialize System (INZSYS) 249

Parameters 249
Message queue (MSGQ) 249
Examples 250
Error messages 250

Initialize Tape (INZTAP) 251

Parameters 251
Device (DEV) 251
New volume identifier (NEWVOL) 252
New owner identifier (NEWOWNID) 252
Volume identifier (VOL) 252
Check for active files (CHECK) 253
Tape density (DENSITY) 253
Code (CODE) 257
End of tape option (ENDOPT) 257
Clear (CLEAR) 257
Examples 258
Error messages 258

Iterate (ITERATE) 261

Parameters 261
Command label (CMDLBL) 261
Examples 261
Error messages 262

Leave (LEAVE) 263

Parameters 263
Command label (CMDLBL) 263
Examples 264
Error messages 264

Link/Unlink Data Definition (LNKDTADFN) 265

Parameters 265
Option (OPTION) 265
Data base file (FILE) 266
Data dictionary (DTADCT) 266
File definition (DFN) 266
Creation date (CRTDATE) 266
Examples 267
Error messages 267

Load or Unload Image Catalog (LODIMGCLG) 269

Parameters 269
Image catalog (IMGCLG) 269
Virtual optical device (DEV) 270
Option (OPTION) 270
Examples 270
Error messages 270

Load/Unload/Mount IMGCLG Entry (LODIMGCLGE) 271

Parameters 271
Image catalog (IMGCLG) 271
Image catalog index (IMGCLGIDX) 272
Option (OPTION) 272
Examples 272

Error messages 272

Load Program Temporary Fix (LODPTF) 275

Parameters 275
Product (LICPGM) 275
Device (DEV) 276
PTF numbers to select (SELECT) 276
PTF numbers to omit (OMIT) 276
Superseded PTFs (SPRPTF) 276
Release (RLS) 277
Sequence number (SEQNBR) 277
End of media option (ENDOPT) 277
Path identifier (PATHID) 278
Save file (SAVF) 278
Copy PTF cover letter (COVER) 279
Examples 279
Error messages 279

Load Q/A Database (LODQSTDB) 283

Parameters 283
Q/A database (QSTDB) 283
Lib containing Q/A database (LIB) 284
Examples 284
Error messages 284

Load and Run (LODRUN) 285

Parameters 287
Device (DEV) 287
Sequence number (SEQNBR) 287
Volume identifier (VOL) 288
Directory (DIR) 288
Examples 288
Error messages 289

Create Directory (MD) 291

Parameters 292
Directory (DIR) 292
Public authority for data (DTAAUT) 292
Public authority for object (OBJAUT) 293
Auditing value for objects (CRTOBJAUD) 294
Scanning option for objects (CRTOBJSCAN) 294
Restricted rename and unlink (RSTDRNMUNL) 295
Examples 295
Error messages 296

Create Directory (MKDIR) 297

Parameters 298
Directory (DIR) 298
Public authority for data (DTAAUT) 298
Public authority for object (OBJAUT) 299
Auditing value for objects (CRTOBJAUD) 300
Scanning option for objects (CRTOBJSCAN) 300
Restricted rename and unlink (RSTDRNMUNL) 301
Examples 301
Error messages 302

Monitor Message (MONMSG) 303

Parameters 304

Message identifier (MSGID)	304
Comparison data (CMPDTA)	305
Command to execute (EXEC)	305
Examples.	305
Error messages	306

Add Mounted FS (MOUNT). 307

Parameters	307
Type of file system (TYPE)	308
File system to mount (MFS)	308
Directory to mount over (MNTOVRDIR)	308
Mount options (OPTIONS)	309
Coded character set ID (CCSID)	310
Code page (CODEPAGE)	311
Examples.	312
Error messages	313

Move Object (MOV). 315

Parameters	316
Object (OBJ)	316
To directory (TODIR)	316
To object (TOOBJ)	316
From CCSID (FROMCCSID)	317
To CCSID (TOCCSID)	317
Data Format (DTAFMT)	318
From Code Page (FROMCODPAG)	318
To Code Page (TOCODEPAGE)	319
Examples.	319
Error messages	320

Move Document (MOVDOC) 321

Parameters	321
From document (FROMDOC)	321
From folder (FROMFLR)	322
To folder (TOFLR)	322
Rename (RENAME)	322
System object name (SYSOBJNAM)	323
Examples.	323
Error messages	323

Move Object (MOVE) 325

Parameters	326
Object (OBJ)	326
To directory (TODIR)	326
To object (TOOBJ)	326
From CCSID (FROMCCSID)	327
To CCSID (TOCCSID)	327
Data Format (DTAFMT)	328
From Code Page (FROMCODPAG)	328
To Code Page (TOCODEPAGE)	329
Examples.	329
Error messages	330

Move Object (MOV OBJ) 331

Parameters	332
Object (OBJ)	332
Object type (OBJTYPE)	333
To library (TOLIB)	333
From ASP device (ASPDEV)	333
To ASP device (TOASPDEV)	334

Examples.	334
Error messages	335

Merge Message Catalog (MRGMSGCLG) 339

Parameters	339
Message catalog name (CLGFILE)	339
Source file path name (SRCFILE)	340
Message catalog CCSID (CLGCCSID)	340
Text 'description' (TEXT)	340
Source file CCSID (SRCCCSID)	340
Public authority for data (DTAAUT)	341
Public authority for object (OBJAUT)	342
Examples.	342
Error messages	342

Merge Message File (MRGMSGF) 343

Parameters	343
From message file (FROMMSGF)	344
To message file (TOMSGF)	344
Replaced message file (RPLMSGF)	344
Message IDs to select (SELECT)	345
Message IDs to omit (OMIT)	345
Examples.	345
Error messages	346

Merge TCP/IP Host Table (MRGTCPTH) 349

Parameters	349
From file (FROMFILE)	349
From member (FROMMBR)	350
File format (FILEFMT)	350
Replace host table (REPLACE)	350
Examples.	350
Error messages	351

Work with TCP/IP Network Sts (NETSTAT). 353

Parameters	353
Option (OPTION)	353
Examples.	353
Error messages	354

Start DNS Query (NSLOOKUP) 355

Parameters	358
Domain Name Server (DMNNAMSVR)	359
Examples.	359
Error messages	359

Open Data Base File (OPNDBF) 361

Parameters	361
File (FILE)	362
Open option (OPTION)	362
Member to be opened (MBR)	362
Open file identifier (OPNID)	362
Access path to use (ACCPH)	363
Limit to sequential only (SEQONLY)	363
Commitment control active (COMMIT)	363
Open scope (OPNSCOPE)	364

Duplicate key check (DUPKEYCHK)	364
Type of open (TYPE)	364
Examples	364
Error messages	365

Open Query File (OPNQRYF) 367

Parameters	368
File specifications (FILE)	370
Open options (OPTION)	371
Format specifications (FORMAT)	371
Query selection expression (QRYSLT)	372
Key field specifications (KEYFLD)	373
Unique key fields (UNIQUEKEY)	374
Join field specifications (JFLD)	375
Join with default values (JDFTVAL)	377
Join file order (JORDER)	378
Grouping field names (GRPFLD)	378
Group selection expression (GRPSLT)	379
Mapped field specifications (MAPFLD)	379
Ignore decimal data errors (IGNDECERR)	382
Open file identifier (OPNID)	383
Limit to sequential only (SEQONLY)	383
Commitment control active (COMMIT)	383
Open scope (OPNSCOPE)	384
Duplicate key check (DUPKEYCHK)	384
Allow copy of data (ALWCPYDTA)	384
Performance optimization (OPTIMIZE)	385
Optimize all access paths (OPTALLAP)	386
Sort sequence (SRTSEQ)	386
Language ID (LANGID)	387
Final output CCSID (CCSID)	387
Type of open (TYPE)	387
Examples	388
Error messages	391

Otherwise (OTHERWISE) 395

Parameters	395
Command (CMD)	395
Examples	396
Error messages	396

Override with Data Base File (OVRDBF) 397

Parameters	397
File being overridden (FILE)	398
Overriding to data base file (TOFILE)	399
Overriding member (MBR)	399
Starting position in file (POSITION)	400
Record format lock (RCDFMTLCK)	401
Records to force a write (FRCRATIO)	402
Rcd format selector program (FMTSLR)	402
Maximum file wait time (WAITFILE)	403
Maximum record wait time (WAITRCD)	403
Records retrieved at once (NBRRCD)	403
EOF retry delay in sec (EOFDLY)	404
Record format level check (LVLCHK)	404
Check expiration date (EXPCHK)	405
Inhibit write (INHWRIT)	405
Secure from other overrides (SECURE)	405
Override scope (OVRSCOPE)	405

Share open data path (SHARE)	406
Open scope (OPNSCOPE)	406
Limit to sequential only (SEQONLY)	406
Distributed Data (DSTDTA)	408
Examples	408
Error messages	409

Override with Diskette File (OVRDKTF) 411

Parameters	411
File being overridden (FILE)	412
Overriding to diskette file (TOFILE)	412
Device (DEV)	413
Volume identifier (VOL)	413
File label (LABEL)	413
Diskette file exchange type (EXCHTYPE)	414
Code (CODE)	414
Creation date (CRTDATE)	414
File expiration date (EXPDATE)	415
Spool the data (SPOOL)	415
Output queue (OUTQ)	415
Max spooled output records (MAXRCDS)	416
Spooled output schedule (SCHEDULE)	416
Hold spooled file (HOLD)	416
Save spooled file (SAVE)	417
Output priority (on OUTQ) (OUTPTY)	417
User data (USRDTA)	417
User specified DBCS data (IGCDTA)	417
Maximum file wait time (WAITFILE)	418
Secure from other overrides (SECURE)	418
Override scope (OVRSCOPE)	418
Share open data path (SHARE)	418
Open scope (OPNSCOPE)	419
Examples	419
Error messages	419

Override with Display File (OVRDSPF) 421

Parameters	421
File being overridden (FILE)	422
Overriding to display file (TOFILE)	422
Device (DEV)	422
Character identifier (CHRID)	423
Decimal format (DECfmt)	424
SFLEND text (SFLENDTXT)	424
User specified DBCS data (IGCDTA)	424
DBCS extension characters (IGCEXNCHR)	424
Maximum file wait time (WAITFILE)	425
Maximum record wait time (WAITRCD)	425
Record format level check (LVLCHK)	426
Secure from other overrides (SECURE)	426
Override scope (OVRSCOPE)	426
Data queue (DTAQ)	427
Share open data path (SHARE)	427
Open scope (OPNSCOPE)	427
Examples	428
Error messages	428

Override ICF Pgm Device Entry (OVRICFDEVE) 429

Parameters	429
Program device (PGMDEV)	430

Remote location (RMTLOCNAME)	430	Overriding to printer file (TOFILE)	456
Communication type (CMNTYPE)	431	Device (DEV)	456
Device (DEV)	432	Printer device type (DEVTYPE)	456
Local location (LCLLOCNAME)	432	Page size (PAGESIZE)	457
Mode (MODE)	432	Lines per inch (LPI)	458
Remote network identifier (RMTNETID)	433	Characters per inch (CPI)	458
Format select (FMTSLT)	433	Front margin (FRONTMGN)	459
Application identifier (APPID)	433	Back margin (BACKMGN)	460
Batch activity (BATCH)	434	Overflow line number (OVRFLW)	460
Host type (HOST)	434	Fold records (FOLD)	461
End session with host (ENDSSNHOST)	434	Unprintable character action (RPLUNPRT)	461
Special host application (SPCHOSTAPP)	434	Align page (ALIGN)	462
Initialize self (INZSELF)	435	Source drawer (DRAWER)	462
Header processing (HDRPROC)	435	Output bin (OUTBIN)	462
Message protection (MSGPTC)	435	Font specifications (FONT)	462
Emulation device (EMLDEV)	435	Form feed (FORMFEED)	463
Conversation type (CNVTYPE)	436	Print quality (PRTQLTY)	463
Blocking type (BLOCK)	437	Control character (CTLCHAR)	464
Record length (RCDLEN)	438	Channel values (CHLVAL)	465
Block length (BLKLEN)	438	Fidelity (FIDELITY)	465
Transmit in transparent mode (TRNSPY)	438	Character identifier (CHRID)	466
Data compression (DTACPR)	439	Decimal format (DECfmt)	466
Truncate trailing blanks (TRUNC)	439	Font character set (FNTCHRSET)	466
Overflow data (OVRFLWDTA)	439	Coded font (CDEFNT)	467
Group separator type (GRPSEP)	439	Page definition (PAGDFN)	468
Remote BSCCEL (RMTBSCCEL)	440	Form definition (FORMDF)	468
Initial connection (INLCNN)	440	AFP Characters (AFPCHARS)	469
Secure from other overrides (SECURE)	440	Table Reference Characters (TBLREFCHR)	470
Override scope (OVRSCOPE)	441	Degree of page rotation (PAGRTT)	470
Examples	441	Pages per side (MULTIUP)	470
Error messages	442	Reduce output (REDUCE)	471
Override ICF File (OVRICFF)	443	Print text (PRTTXT)	471
Parameters	443	Hardware justification (JUSTIFY)	471
File being overridden (FILE)	444	Print on both sides (DUPLEX)	471
Overriding to ICF file (TOFILE)	444	Unit of measure (UOM)	472
Acquire program device (ACQPGMDEV)	444	Front side overlay (FRONTOVL)	472
Maximum record length (MAXRCDLEN)	445	Back side overlay (BACKOVL)	473
Maximum file wait time (WAITFILE)	445	Convert line data (CVTLINDTA)	474
Maximum record wait time (WAITRCD)	445	IPDS pass through (IPDSPASTHR)	474
Record format level check (LVLCHK)	446	User resource library list (USRRSCLIBL)	475
Secure from other overrides (SECURE)	446	Corner staple (CORNERSTPL)	475
Override scope (OVRSCOPE)	446	Edge stitch (EDGESTITCH)	476
Data queue (DTAQ)	446	Saddle stitch (SADLSTITCH)	477
Share open data path (SHARE)	447	Font resolution for formatting (RNTRSL)	478
Open scope (OPNSCOPE)	447	Defer write (DFRWRT)	479
Examples	447	Spool the data (SPOOL)	479
Error messages	448	Output queue (OUTQ)	479
Override Message File (OVRMSGF)	449	Form type (FORMTYPE)	480
Parameters	449	Copies (COPIES)	480
Message file being overridden (MSGF)	449	Page range to print (PAGERANGE)	480
Overriding to message file (TOMSGF)	450	Max spooled output records (MAXRCDSD)	480
Secure from other overrides (SECURE)	450	File separators (FILESEP)	481
Examples	450	Spooled output schedule (SCHEDULE)	481
Error messages	450	Hold spooled file (HOLD)	481
Override with Printer File (OVRPRTF)	451	Save spooled file (SAVE)	482
Parameters	451	Output priority (on OUTQ) (OUTPTY)	482
File being overridden (FILE)	455	User data (USRDTA)	482
		Spool file owner (SPLFOWN)	482
		User Defined Option (USRDFNOPT)	483
		User Defined Data (USRDFNDTA)	483
		User Defined Object (USRDFNOBJ)	484

Spool file name (SPLFNAME)	485
User specified DBCS data (IGCDTA)	485
DBCS extension characters (IGCEXNCHR)	485
DBCS character rotation (IGCCHRRTT).	485
DBCS characters per inch (IGCCPI)	485
DBCS SO/SI spacing (IGCSOSI)	486
DBCS coded font (IGCCDEFNT)	486
Maximum file wait time (WAITFILE)	487
Record format level check (LVLCHK)	487
Secure from other overrides (SECURE)	487
Override scope (OVRSCOPE)	488
Share open data path (SHARE)	488
Open scope (OPNSCOPE)	488
Examples	488
Error messages	489
Override with Save File (OVRSAVF)	491
Parameters	491

File being overridden (FILE)	491
Save file (TOFILE).	492
Extend file (EXTEND)	492
Starting position in file (POSITION).	492
Maximum file wait time (WAITFILE)	493
Secure from other overrides (SECURE)	493
Override scope (OVRSCOPE)	493
Share open data path (SHARE)	494
Open scope (OPNSCOPE)	494
Examples	494
Error messages	494

Appendix. Notices	495
Trademarks	496
Terms and conditions for downloading and printing publications	497
Code disclaimer information	497

End Mail Server Framework (ENDMSF)

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

Parameters
Examples
Error messages

The End Mail Server Framework (ENDMSF) command ends the mail server framework jobs in the system work subsystem (QSYSWRK).

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	*CNTRLD, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	1-999999, <u>30</u>	Optional, Positional 2

Top

How to end (OPTION)

Specifies whether the mail server framework jobs that are in the system work subsystem (QSYSWRK) end immediately or in a controlled manner.

The possible values are:

*CNTRLD

All mail server framework jobs are ended in a controlled manner. This allows each framework job a chance to complete processing the current mail server framework messages before it ends.

*IMMED

All mail server framework jobs are ended immediately. Any mail server framework messages being processed at the time the job ended are processed when the mail server framework is restarted.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the mail server framework jobs to complete their processing during a controlled end. This parameter is ignored if OPTION(*IMMED) is specified. If the jobs do not end before the end of the delay time, they are then immediately ended.

The possible values are:

30 A maximum delay time of 30 seconds is allowed before the mail server framework jobs are ended.

delay-time

Specify the maximum amount of delay time in seconds before the jobs are ended. Valid values range from 1 through 999999.

Top

Examples

Example 1: Ending Mail Server Framework in a Controlled Manner

```
ENDMSF OPTION(*CNTRLD) DELAY(60)
```

This command ends the mail server framework jobs in the system work subsystem in a controlled manner and has 60 seconds to complete processing any mail server framework messages currently being handled.

Example 2: Ending Mail Server Framework Immediately

```
ENDMSF OPTION(*IMMED)
```

This command ends the mail server framework jobs in the system work subsystem immediately. The mail server framework jobs do not complete processing any mail server framework messages currently being handled.

Top

Error messages

*ESCAPE Messages

CPFAFAB

ENDMSF did not complete successfully.

CPFAFAC

ENDMSF completed successfully; however errors occurred.

CPFAFFF

Internal system error in program &1.

Top

End NFS Server (ENDNFSSVR)

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

Parameters
Examples
Error messages

The End Network File System Server (ENDNFSSVR) command ends one or all of the NFS server daemons. For more information about these daemon jobs, see OS/400 Network File System book, SC41-5714.

SERVER(*ALL) should be specified, which will end the daemons in the following order. (This order is the recommended order for ending the Network File System daemons.)

- The network lock manager (NLM) daemon
- The network status monitor (NSM) daemon
- The mount (MNT) daemon
- The server (SVR) daemon(s)
- The block input/output (I/O) (BIO) daemon(s)
- The Remote Procedure Call (RPC) RPCBind daemon

If just one daemon is to be ended, be sure the appropriate order for ending NFS daemons and the possible consequences of ending daemons in an order other than that specified above are understood. For more information about ending NFS daemons, see the OS/400 Network File System book, SC41-5714.

If the user attempts to end a daemon or daemons that are not running, they will not cause the command to fail, and it will continue to end other daemons that were requested to end.

To determine if an NFS daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following jobs:

QNFSRPCD The RPCBind daemon
QNFSBIOD The block I/O (BIO) daemon
QNFSNFSD The server (SVR) daemon
QNFSMNTD The mount (MNT) daemon
QNFSNSMD The network status monitor (NSM) daemon
QNFSNLMD The network lock manager (NLM) daemon

Restrictions:

1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
2. The user must have job control (*JOBCTL) special authority to end any daemon jobs that were started by someone else.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server daemon	*ALL, *RPC, *BIO, *SVR, *MNT, *NSM, *NLM	Required, Positional 1
ENDJOBTIMO	Timeout for end of daemon	1-3600, <u>30</u> , *NOMAX	Optional, Positional 2

Server daemon (SERVER)

Specifies the Network File System (NFS) daemon jobs to be ended.

- *ALL** All NFS daemons will be ended.
- *RPC** The NFS Remote Procedure Call (RPC) RPCBind daemon will be ended.
- *BIO** All NFS block input/output (I/O) daemons that are running will be ended.
- *SVR** All NFS server daemons that are running will be ended.
- *MNT** The NFS mount daemon will be ended.
- *NSM** The NFS network status monitor daemon will be ended.
- *NLM** The NFS network lock manager daemon will be ended.

This is a required parameter.

Timeout for end of daemon (ENDJOBTIMO)

Specifies the number of seconds to wait for each daemon to successfully end. If a daemon has not ended within the timeout value, the command will fail.

30 Wait 30 seconds for the daemon job to end.

***NOMAX**
Wait forever for daemons to end; do not timeout.

1-3600 Specify the number of seconds to wait for daemons to end before timing out and failing the command. Timeout values less than 30 seconds are rounded up to 30 seconds.

Examples

Example 1: End All Daemons

```
ENDNFSSVR  SERVER(*ALL)
```

This command ends all NFS daemon jobs that are running.

Example 2: End a Single Daemon

```
ENDNFSSVR  SERVER(*MNT)  ENDJOBTIMO(*NOMAX)
```

This command ends the NFS mount daemon, and waits forever for it to end. The mount daemon was previously running, and other daemons have been ended in the appropriate order.

Error messages

None

End NetWare Connection (ENDNTWCNN)

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

Parameters
Examples
Error messages

The End NetWare Connection (ENDNTWCNN) command allows a user to end active NetWare connections. This command can be used to end a specific NetWare connection on a particular NetWare server. This connection may or may not have originated from an iSeries.

Restrictions: You need *JOBCTL special authority to end a connection other than your own.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server	<i>Character value</i>	Required, Positional 1
CNNNBR	Connection number	1-65534	Required, Positional 2

Top

Server (SERVER)

Specifies the NetWare server whose connection or connections are to be ended.

name Specify the name of an active server defined for the network.

Top

Connection number (CNNNBR)

Specifies the connection number for the active NetWare connection that is to be ended.

1-65534

Specify the connection number for the NetWare connection to be ended.

Top

Examples

None

Top

Error messages

*ESCAPE Messages

FPE0103

NetWare connection &1 not ended.

[Top](#)

End Network Interface Recovery (ENDNWIRCY)

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

Parameters
Examples
Error messages

The End Network Interface Recovery (ENDNWIRCY) command ends automatic error recovery procedures for a network interface description.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
NWI	Network interface description	<i>Name</i>	Required, Positional 1

[Top](#)

Network interface description (NWI)

Specifies the name of the network interface description whose recovery is to be ended.

This is a required parameter.

[Top](#)

Examples

```
ENDNWIRCY  NWID(ISDNNET)
```

This command ends automatic error recovery procedures for the network interface named ISDNNET.

[Top](#)

Error messages

*ESCAPE Messages

CPF591A

Not authorized to network interface description &1.

CPF593A

Network interface &1 not varied on.

CPF593B

Network interface description &1 not found.

CPF593C

Cannot access network interface &1.

[Top](#)

End Pass-Through (ENDPASTHR)

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

Parameters
Examples
Error messages

The End Pass-Through (ENDPASTHR) command ends a pass-through session. The ENDPASTHR command signs you off the target system, and ends the advanced program-to-program communications (APPC) session. This releases the virtual display device from the subsystem and returns it to the vary-on pending condition. The job at each intermediate node for the pass-through session also ends. Control returns to the source system for the next command following the Start Pass-Through (STRPASTHR) command.

Note: The ENDPASTHR command uses the SIGNOFF command as part of its processing. If the system has a SIGNOFF command that appears in the library list before QSYS/SIGNOFF, the SIGNOFF command is used by ENDPASTHR. The SIGNOFF command should not use the ENDPASTHR command. It sends the system into a loop when you end your pass-through session.

The ENDPASTHR command does not end the passthrough session when there is a secondary interactive job at the target system. One of the jobs must be ended (by using SIGNOFF or ENDJOB) before the ENDPASTHR command can be entered.

If the ENDPASTHR command is entered and there is not a pass-through session, an error message is sent.

More information about pass-through is in the Remote Work Station Support book, SC41-5402.

Top

Parameters

Keyword	Description	Choices	Notes
LOG	Job log	<u>*NOLIST</u> , *LIST	Optional, Positional 1

Top

Job log (LOG)

Specifies whether the job log is saved at the target system.

*NOLIST

The information in the job log is deleted when the job ends.

***LIST** A job log is saved at the target system.

Top

Examples

ENDPASTHR LOG(*LIST)

This command ends a pass-through session and prints a job log.

[Top](#)

Error messages

*ESCAPE Messages

CPF8914

ENDPASTHR command not allowed.

CPF8915

ENDPASTHR not allowed. System request job active.

[Top](#)

End Performance Explorer (ENDPEX)

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

Parameters
Examples
Error messages

The End Performance Explorer (ENDPEX) command instructs the Performance Explorer tool to stop collecting data. The command expects a session name to accompany the request which identifies which instance of the Performance Explorer session to end.

The user can either end the data collection session or suspend the data collection session. If the user chooses to end the session, the collected data is put into an object of type *MGTCOL or into a set of database files, or it is deleted, based on the value specified for the DTAOPT parameter.

If the user chooses to suspend the collection of performance data, the session remains active. To resume data collection for a suspended session, the user can specify OPTION(*RESUME) on a subsequent call of the STRPEX (Start Performance Explorer) command.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The user must have *ADD and *EXECUTE authority to the specified DTALIB and MGTCOL libraries.
3. The user must have *OBJMGMNT, *OBJEXIST, and use authorities to the management collection object if replacing an existing management collecton object.
4. To use this command you must have *SERVICE special authority, or be authorized to the Service Trace function of Operating System/400 through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
5. The following user profiles have private authorities to use the command:
 - QPGMR
 - QSRV
6. If running ENDPEX from a secondary thread, the QAYPE* files must already exist in the DTALIB library. These files must be created in the primary thread by running ENDPEX DTAOPT(*LIB).
7. If running ENDPEX from a secondary thread, DTAOPT(*MGTCOL) object cannot be specified.
8. Two threads within the same job will not be allowed to run ENDPEX at the same time. The thread that issued ENDPEX first will run the command to completion while the second ENDPEX waits.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	Name, * <u>SELECT</u>	Optional, Positional 1
OPTION	Option	* <u>END</u> , * <u>SUSPEND</u> , * <u>STOP</u>	Optional
DTAOPT	Data option	* <u>LIB</u> , * <u>MGTCOL</u> , * <u>DLT</u>	Optional
DTALIB	Data library	Name, <u>QPEXDATA</u>	Optional
DTAMBR	Data member	Name, * <u>SSNID</u>	Optional

Keyword	Description	Choices	Notes
MGTCOL	Management collection	<i>Qualified object name</i>	Optional
	Qualifier 1: Management collection	<i>Name, *SSNID</i>	
	Qualifier 2: Library	<i>Name, QPEXDATA</i>	
RPLDTA	Replace data	*NO, *YES	Optional
NBRTHD	Number of threads	1-256, *CALC	Optional
TEXT	Text 'description'	<i>Character value, *BLANK</i>	Optional

Top

Session ID (SSNID)

Specifies which Performance Explorer session to end. This is the session identifier that was specified on the STRPEX (Start Performance Explorer) command.

*SELECT

A list panel of all active Performance Explorer data collection sessions will be displayed with an option to select which session to end. *SELECT is only valid if the ENDPEX command is being run interactively. If the command is being run in batch, a session identifier must be specified.

session-identifier

Specify the Performance Explorer data collection session to end.

Top

Option (OPTION)

Specifies whether to end the data collection session or just suspend collection of performance data for the session.

*END The Performance Explorer session is ended. The user is prompted for a choice of three methods to handle the collected data:

1. Save the collected data to a set of database files.
2. Save the data to a single file (used for sending data to IBM for analysis).
3. Discard the data.

*SUSPEND

The Performance Explorer session is suspended, and the session remains active but no additional data is collected for this session. Once a session is suspended, the user can use STRPEX with OPTION(*RESUME) to resume data collection, end the suspended session by specifying ENDPEX with OPTION(*END), or stop the suspended session by specifying ENDPEX with OPTION(*STOP).

*STOP

The Performance Explorer session is ended and the jobs are removed from the collection. The session cannot be started up again. Addresses are not resolved to object names, and no database files are created. The address data and database files can be created at a later time with the OPTION(*END) and DTAOPT(*LIB or *MGTCOL) options of ENDPEX. However, Performance Explorer may not be able to resolve some of the addresses if objects get deleted. The longer the time between *STOP and *END, the greater the chance the resolved address data will be incomplete.

Top

Data option (DTAOPT)

Specifies how to handle the collected data. The collected data can be stored in a set of database files or a management collection object (*MGTCOL), or both. The temporary management collection object used to hold the collected data will be deleted. To delete the temporary management collection object without storing the collected data, specify *DLT.

Note: This parameter is valid only if OPTION(*END) is specified.

***LIB** Indicates to store all of the collected performance data for the session into a set of database files located in the library specified on the DTALIB parameter. The Performance Explorer tool creates all the necessary files if this is the first time that a library is being used to store performance data. The member name for each of the files where the session data is stored can be controlled through the DTAMBR parameter, but defaults to be the same name as the session identifier.

***MGTCOL**

Indicates to store all of the collected data in a management collection object (type *MGTCOL). No database files will be created. This option can be used if the data is to be shipped to another system or to your service provider for analysis.

***DLT** The collected performance data for the session is to be deleted from the system.

Top

Data library (DTALIB)

Specifies the name of the library that contains the set of database files where the collected performance data is stored.

Note: This parameter is valid only if the user specified DTAOPT(*LIB).

QPEXDATA

The QPEXDATA library is the recommended library for storing data collected with the Performance Explorer tool. The first time the Performance Explorer tool is used, this library is created for the user, and a set of database files to store the information is created in that library.

library-name

Specifies the name of the library in which to store the collected data. If the library does not exist, the command ends in an error condition. After the library is created, retry the command. If the specified library does not already have the performance database files, they are created and the data is stored.

Top

Data member (DTAMBR)

Specifies the name to be used for the database file members where the collected performance data is stored. If a member does not exist by the specified name, it is created.

Note: This parameter is valid only when DTAOPT(*LIB) is specified.

***SSNID**

The member name is the same as the value specified for the SSNID parameter.

member-name

Specify the member name to use when storing the collected data in Performance Explorer database files.

Management collection (MGTCOL)

Specifies the name of a management collection object to store the collected performance data.

Note: This parameter is valid only if DTAOPT(*MGTCOL) is specified.

The *MGTCOL object name can be qualified by one of the following library values:

-

QPEXDATA

The QPEXDATA library is the recommended library for storing data collected by the Performance Explorer tool. The first time the Performance Explorer tool is used, this library is created for the user.

data-library-name

Specify the name of the library to store the collected data. If the library does not exist, the command ends in an error condition. After the library is created, retry the command.

*SSNID

The name specified for the SSNID parameter is used when creating the management collection object to contain the collected performance data.

management-collection-name

Specify the name to use when creating the management collection object to contain the collected performance data.

Top

Replace data (RPLDTA)

Specifies whether to replace the data in an existing file member or management collection object with the new performance data. If DTAMBR was specified and a member with the same name already exists in any of the Performance Explorer database files in the specified library (DTALIB parameter), this parameter controls whether the member data is replaced. If MGTCOL was specified and an object already exists with the same name, this parameter controls whether the data in that object is replaced.

***NO** If a member already exists with the same name, an error message is sent to the user. This prevents the user from inadvertently writing over existing data.

***YES** If a member already exists with the same name, the old data is lost and is replaced by the new data.

Top

Number of threads (NBRTHD)

Specifies the number of concurrent threads that the ENDPEX command uses to process the data in the session being ended. Specifying a number greater than 1 allows the ENDPEX 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 ENDPEX command to a higher number. You should also verify that the disk subsystem can handle the additional threads.

*CALC

The system calculates a reasonable number of threads to do the command processing which does not use excessive CPU or disk resources.

***MAX** The system calculates a maximum number of threads to do the command processing. An attempt will be made to maximize utilization on all resources to minimize processing time. This may cause severe degradation for all other jobs on the system.

number-of-threads

Specify the number of threads for the ENDPEX command to use to process the collected data.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the type of data collected.

*BLANK

Text is not specified.

'description'

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

Top

Examples

Example 1: End a Session and Save the Database Files

```
ENDPEX  SSNID(TEST3)  OPTION(*END)  DTAOPT(*LIB)
        DTAMBR(SYS1DATA)
```

This command ends the performance explorer session named TEST3 and saves the data in a set of database files in library QPEXDATA. The member name to be used for each file is SYS1DATA.

Example 2: End a Session and Delete the Data

```
ENDPEX  SSNID(TESTRUN)  OPTION(*END)  DTAOPT(*DLT)
```

This command ends the performance explorer session named TESTRUN and deletes the collected performance data.

Example 3: End a Session and Save the *MGTCOL

```
ENDPEX  SSNID(TEST3)  OPTION(*END)  DTAOPT(*MGTCOL)
        MGTCOL(MYLIB/SYS1DATA)  NBRTHD(*CALC)
```

This command ends the performance explorer session named TEST3 and saves the data in a management collection object in library MYLIB in the management collection object named SYS1DATA. ENDPEX will calculate a number of threads to process this request. This number of threads will do the ENDPEX processing as quickly as possible without disrupting the rest of the system.

Top

Error messages

*ESCAPE Messages

CPFAF06

ENDPEX command was not successful. Reason code is &1. See details for more information.

End Performance Collection (ENDPFRCOL)

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

Parameters
Examples
Error messages

The End Performance Collection (ENDPFRCOL) command stops the system-level collection. If there are no other client applications using Collection Services, the Collection Services server job (QYPSPFRCOL) will also end.

If client applications are using Collection Services, the server job will continue to run unless you also specify the Force end parameter. Forcing the server job to end when it is being used by client applications (for example, iSeries Navigator monitors or Performance Collector APIs) will cause those clients to experience data collection failure.

Other system functions are capable of starting Collection Services. Even though this command ends the current instance of the server job, it does not prevent the server job from being restarted. Functions which can start the server job include PM/400, the Management Central server, and the Performance Collector APIs.

Top

Parameters

Keyword	Description	Choices	Notes
FRCCOLEND	Force end	<u>*NO</u> , *YES	Optional, Positional 1

Top

Force end (FRCCOLEND)

Determines whether the Collection Services server job (QYPSPFRCOL) should be forced to end.

***NO** The QYPSPFRCOL job will be ended only if it is not being used by a client application.

***YES** The QYPSPFRCOL job will be forced to end immediately.

Top

Examples

Example 1: Ending the Performance Collection

```
ENDPFRCOL
```

This command will end the system-level collection of performance data. If no client applications are using Collection Services, this command will also end the Collection Services server job (QYPSPFRCOL). If client applications are using Collection Services, the QYPSPFRCOL job will continue to run.

Example 2: Forcing the Performance Collection to End

```
ENDPFRCOL FRCCOLEND(*YES)
```

This command will force the Collection Services server job (QYPSPFRCOL) to end, even if it is being used by client applications.

[Top](#)

Error messages

***ESCAPE Messages**

CPF3CF2

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

CPF94A

Collector communications error. Reason code &1.

[Top](#)

End Performance Trace (ENDPFRTTRC)

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

Parameters
Examples
Error messages

The End Performance Trace (ENDPFRTTRC) command will stop the collection of performance trace data in the QPM_STRPFRTTRC trace table and optionally write performance trace data to a data base file. The QPM_STRPFRTTRC trace table will be deleted whether or not the data is written to a data base file.

This command is intended to be used to end a performance trace started via the Start Performance Trace (STRPFRTTRC) command. However, it will end and try to process any active trace in the QPM_STRPFRTTRC trace table.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The following user profiles have private authorities to use the command:
 - QSRV

Top

Parameters

Keyword	Description	Choices	Notes
DMPTRC	Dump the trace	*NO, *YES	Optional, Positional 1
MBR	Member	Name	Optional
LIB	Library	Name, QPFRDATA	Optional
TEXT	Text 'description'	Character value, *BLANK	Optional

Top

Dump the trace (DMPTRC)

Specifies whether the trace data is to be dumped to the performance database file QAPMDMPT. If the data is not dumped, it will be lost when the trace table is deleted.

***YES** The trace data, if any, is dumped.

***NO** The trace data is not dumped.

Top

Member (MBR)

Specifies the member within the QAPMDMPT database file where the trace table data is to be dumped. A value must be specified for this parameter if *YES is specified for the **Dump the trace (DMPTRC)** parameter.

name Specify the name of the database file member to be used.

Library (LIB)

Specifies the library where the database file for trace data is located. If the file is not found in the specified library, the system automatically creates it in that library.

QPFRDATA

IBM-supplied performance data library QPFRDATA is to be used to locate the database file for trace data.

name Specify the name of the library to be searched.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the database member.

*BLANK

No text is specified.

character-value

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

Top

Examples

Example 1: Ending Performance Trace

```
ENDPFRTRC DMPTRC(*YES) MBR(MYDATA)
```

In this example, the current trace is ended, the data is written to member MYDATA of file QAPMDMPT in library QPFRDATA, and the trace table is deleted, releasing the storage used by the trace.

Top

Error messages

*ESCAPE Messages

Refer to the TRCINT and DMPTRC commands for messages.

Top

End Program (ENDPGM)

Where allowed to run:

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

Threadsafe: Yes

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

The End Program (ENDPGM) command specifies the end of a CL procedure. When the command is processed, it performs the same function as a RETURN command. That is, control is returned to the command immediately following the CALL command in the calling program.

The ENDPGM command is not required at the end of a CL procedure. If the last statement in a CL procedure source file is reached and no ENDPGM command is found, an ENDPGM command is assumed by the compiler.

Restrictions: This command is valid only within a CL procedure.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

```
PGM
:  
ENDPGM
```

This CL procedure is identified by a PGM command that contains no parameters and is ended by the ENDPGM command.

[Top](#)

Error messages

None

[Top](#)

End Program Export List (ENDPGMEXP)

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

The End Program Export List (ENDPGMEXP) binder definition statement ends a list of exports in a service program export block.

There are no parameters for this statement.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDPGMEXP

This binder definition statement marks the end of a list of exported variables or procedures for a service program.

[Top](#)

Error messages

None

[Top](#)

End Program Profiling (ENDPGMPRF)

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

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

The End Program Profiling (ENDPGMPRF) command ends collection of program profiling data for programs or service programs that have been enabled to collect profiling data using the PRFDTA(*COL) option on the CHGPGM (Change Program), CHGSRVPGM (Change Service Program) CL command, or when the modules are created using the CHGMOD (Change Module) CL command.

Restrictions:

- This command is shipped with no public (*EXCLUDE) authority, and QPGMR user profile having use (*USE) authority to the command.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDPGMPRF

This command ends program profile data collection.

[Top](#)

Error messages

*ESCAPE Messages

CPF5CAA

Unexpected error occurred during program profiling.

[Top](#)

End Prestart Jobs (ENDPJ)

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

Parameters
Examples
Error messages

The End Prestart Jobs (ENDPJ) command ends all jobs and any associated inline data files for a prestart job entry in an active subsystem.

Jobs can be waiting for a request or can already be associated with a request. Spooled output files associated with the jobs being ended can also be ended or allowed to remain on the output queue. The limit on the number of messages being written to each of the job logs can also be changed. **Restrictions:**

1. This command is restricted to a user with job control (*JOBCTL) special authority.
2. Spooled output files on output queues in independent auxiliary storage pools (ASPs 33-255) are not deleted.

Top

Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	<i>Name</i>	Required, Positional 1
PGM	Program	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OPTION	How to end	<i>*CNTRLD, *IMMED</i>	Optional, Positional 3
DELAY	Controlled end delay time	1-999999, <u>30</u>	Optional, Positional 4
SPLFILE	Delete spooled files	<i>*NO, *YES</i>	Optional, Positional 5
LOGLMT	Maximum log entries	<i>Integer, *SAME, *NOMAX</i>	Optional

Top

Subsystem (SBS)

Specifies the name of the active subsystem that contains the prestart job entry.

This is a required parameter.

name Specify the name of the active subsystem that contains the prestart job entry.

Top

Program (PGM)

Specifies the qualified name of the program that identifies the prestart job entry.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program that identifies the prestart job entry.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the library where the program is located.

Top

How to end (OPTION)

Specifies whether the jobs are ended in a controlled manner, which lets the application program perform end-of-job processing, or the jobs end immediately.

***CNTRLD**

The jobs are ended in a controlled manner. This allows the program that is running to perform 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. The application has the amount of time specified on the DELAY parameter to complete cleanup before the job is ended.

***IMMED**

The jobs end 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 QENDJOBLMT system value specifies the time limit. Other than by handling the SIGTERM signal, the program that is running is not allowed to perform end-of-job processing.

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.

Top

Controlled end delay time (DELAY)

Specifies the time (in seconds) allowed for the program to complete end-of-job processing during a controlled end. If the end-of-job processing is not completed before the end of the delay time, the job is immediately ended. Only system cleanup is performed. This parameter is not valid if *IMMED is specified for the **How to end (OPTION)** parameter.

30 A maximum delay time of 30 seconds is allowed for end-of-job processing before each prestart job is ended.

1-999999

Specify the maximum delay time (in seconds) before each prestart job is ended.

Top

Delete spooled files (SPLFILE)

Specifies whether spooled output files created by the jobs are retained for normal processing by a writer or deleted.

- *NO** The spooled output files created by the jobs being ended are retained for normal processing by a writer. When the job ends, the spooled file action (SPLFACN) job attribute determines whether spooled files are detached from the job or kept with the job.
- *YES** The spooled output files created by the jobs being ended and which are on output queues in the system auxiliary storage pool (ASP 1) or in a basic user ASP (ASPs 2-32) are deleted. Spooled output files on output queues in independent ASPs (ASPs 33-255) are not deleted. The job log is not deleted.

Top

Maximum log entries (LOGLMT)

Specifies the maximum number of entries in the message queue of the jobs being ended that are written to the job log. This parameter can be used to limit the number of messages written to the job log printer file, QPJOBLOG, for each job that is ended.

*SAME

The message logging limit does not change. If the logging limit was not changed for these prestart jobs on a previous command, *NOMAX is the value used by the system.

***NOMAX**

There is no limit to the number of messages being logged. All messages on each job message queue are written to the job log for each job.

integer-number

Specify the maximum number of messages being written to the job log for each job. This value is the maximum only if it is entered before the job log contains that number of messages. Otherwise, the limit only stops the process of writing any more messages to the job log. If 0 is specified before any messages are written to the log, no job log is produced.

Top

Examples

Example 1: Ending a Job Immediately

```
ENDPJ  SBS(SBS1)  PGM(PJLIB/PJPGM)  OPTION(*IMMED)
        SPLFILE(*YES)
```

This command ends all jobs associated with prestart job entry PJPGM in subsystem SBS1 immediately. Spooled output produced by these prestart jobs is deleted and the job log is saved.

Example 2: Delaying a Job End

```
ENDPJ  SBS(SBS2)  PGM(PJPGM2)  OPTION(*CNTRLD)
        DELAY(50)  SPLFILE(NO)
```

This command ends all the jobs associated with prestart job entry PJPGM2 in subsystem SBS2. Spooled output for these prestart jobs is saved for normal processing by the spooling writer. The jobs have 50 seconds to perform any cleanup routines, after which they are immediately ended.

Top

Error messages

*ESCAPE Messages

CPF0922

End Prestart Jobs command not allowed now.

CPF1083

Prestart jobs already are ending controlled.

CPF1084

Prestart jobs are already ending immediately.

CPF1227

No authority has been granted to use command.

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1351

Function check occurred in subsystem for job &3/&2/&1.

CPF1834

Prestart job entry for program &1 in &2 does not exist.

Top

End Printer Emulation (ENDPRTEML)

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

Parameters
Examples
Error messages

The End Printer Emulation (ENDPRTEML) command ends printer emulation without ending the job. If there is another request in the job, that request is then processed.

This command closes the file to the host system, then writes the last data received from the host system out to the spooled file or printer by closing the printer file.

In some cases, the request does not take effect immediately. The request is delayed while any of the following conditions exist in the printer emulation request:

- Printing a block sent from the host system.
- Waiting for a printer error to be cleared (for example, a paper jam).
- Waiting for a reply to a PA1 or PA2 inquiry message.
- Waiting for error recovery to be done to the host system or printer device.
- The job has been held by using the HLDJOB command. When the condition is cleared, the End Printer Emulation request takes effect, and the printer emulation request ends.

Top

Parameters

Keyword	Description	Choices	Notes
EMLDEV	Emulation device, or	<i>Name</i>	Optional, Positional 1
EMLLOC	Emulation location	<i>Communications name</i>	Optional, Positional 2
PRTDEV	Print device	<i>Name</i>	Optional, Positional 3

Top

Emulation device (EMLDEV)

Specifies the name of a printer emulation device that receives data from the host system. This device must be a 3287 Printer (EMLDEV(3287)) or a 3289 Printer (EMLDEV(3289)), and must currently be operating as an LU1 unit. The printer emulation job or session that is using this device will be informed of the request. If the LU1 session is between brackets, printer emulation starts a bracket and sends the PA key signal to the host system with a Change Direction (CD) request. If the LU session is in receive condition, a signal (request for CD) is sent to the host system, and printer emulation waits for the CD. When the CD is received, the PA key signal is sent to the host system with the CD. If the LU session is in send condition, the PA key signal is sent to the host system with the CD.

Either this parameter, or the **Emulation location (EMLLOC)** parameter and the **Print device (PRTDEV)** parameter is required.

Top

Emulation location (EMLLOC)

Specifies the remote location name associated with this session. The location name is defined during device description configuration and refers to the remote location where communication takes place. This value must be the same as the value specified for the Emulation location (EMLLOC) parameter on the Start Printer Emulation (STRPRTEML) command.

Either this parameter and the **Print device (PRTDEV)** parameter, or the **Emulation device (EMLDEV)** parameter is required.

[Top](#)

Print device (PRTDEV)

Specifies the name of a printer device that is used to print the spooled output. This value must be the same as the value specified for the Printer device (PRTDEV) parameter on the Start Printer Emulation (STRPRTEML) command. This parameter must be specified when the EMLLOC parameter is specified.

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

[Top](#)

Examples

```
ENDPRTEML  EMLDEV(HOSTPRT3)
```

This command ends the printer emulation request that is using the device HOSTPRT3.

[Top](#)

Error messages

*ESCAPE Messages

CPF8599

End printer emulation function not performed.

[Top](#)

End Receive (ENDRCV)

Where allowed to run:

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

Threadsafe: No

Parameters
Examples
Error messages

The End Receive (ENDRCV) command is used to end (cancel) a request for input made by a previously issued RCVF or SNDRCVF command that had WAIT(*NO) specified. The ENDRCV command ends an input request even if the user enters the requested data at the display station at the same time that the command is processed. If the requested data is entered and is being sent to the program when the end receive operation is performed, the entered data is lost. If there is no outstanding input request, the command is ignored.

Restrictions:

- This command is valid only for display files within CL procedures. It cannot be used for database files.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Display device	<i>Name</i> , <u>*FILE</u>	Optional, Positional 1
OPNID	Open file identifier	<i>Simple name</i> , <u>*NONE</u>	Optional

Top

Display device (DEV)

Specifies the name of the display device for which the request for input is to be ended.

*FILE The name of the device whose response is to be ended. This device is in the device file that is declared in the **File (FILE)** parameter of the Declare File (DCLF) command. If the device file has more than one device name specified in it, *FILE cannot be specified.

name Specify the name of the display device from which a response is to be ended.

Top

Open file identifier (OPNID)

Specifies the open file identifier that was declared on a preceding Declare File (DCLF) command in the same CL procedure. A CL variable cannot be specified for this parameter value.

*NONE

No open file identifier is provided. This command will use the file associated with the DCLF command that had *NONE specified for the OPNID parameter. Only one file can be declared in a CL procedure with *NONE as the open file identifier.

simple-name

Specify a name that matches the OPNID parameter value on a preceding DCLF command in the same CL procedure.

Top

Examples

Example 1: Ending Previous Receive

```
ENDRCV  DEV(MYDISPLAY)
```

Assume that a RCVF command with WAIT(*NO) was issued earlier in the CL procedure to request input from the device file declared earlier in the DCLF command and from the display device MYDISPLAY. When this ENDRCV command is processed, that request for input from MYDISPLAY is ended.

Example 2: Using an Open File Identifier

```
DCLF  FILE(MYLIB/MYDSPFILE)  RCFMT(FMT1)  OPNID(DSPFILE1)
:
SNDRCVF  DEV(DSP02)  RCFMT(FMT1)  OPNID(DSPFILE1)  WAIT(*YES)
:
ENDRCV  DEV(DSP02)  OPNID(DSPFILE1)
```

This command ends the previous SNDRCVF (Send/Receive File) command's request for input from a workstation display device DSP02.

Top

Error messages

*ESCAPE Messages

CPF0883

*FILE not valid in DEV parameter for file &1.

CPF4101

File &2 in library &3 not found or inline data file missing.

Top

(ENDRDBRQS)

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

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

Parameters

None

[Top](#)

Examples

None

[Top](#)

Error messages

Unknown

[Top](#)

End Reader (ENDRDR)

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

Parameters
Examples
Error messages

The End Reader (ENDRDR) command ends the specified diskette or database reader and makes its associated input device available to the system. The reader can be stopped either immediately, without completing the current job being read, or at the end of the current job. If the reader is in a hold state when this command is issued, the reader is stopped immediately.

Top

Parameters

Keyword	Description	Choices	Notes
RDR	Reader	<i>Name</i>	Required, Positional 1
OPTION	When to end reader	<u>*CNTRLD</u> , *IMMED	Optional, Positional 2

Top

Reader (RDR)

This is a required parameter.

Specifies the name of the diskette or database reader to be ended. The reader's associated input device is made available to the system.

reader-name

Specify the name of the reader to be ended.

Top

When to end reader (OPTION)

Specifies when the ended reader should stop processing.

The possible values are:

***CNTRLD**

The reader stops processing after the current job is read and an entry for the job is placed on the job queue.

***IMMED**

The reader stops processing immediately. The job being read is not placed on the job queue.

Top

Examples

ENDRDR RDR(DISKETTE)

This command stops the reader DISKETTE as soon as the current job is completely read in and releases that device to the system.

Top

Error messages

*ESCAPE Messages

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1352

Function not done. &3/&2/&1 in transition condition.

CPF3312

Reader &1 neither active nor on job queue.

CPF3330

Necessary resource not available.

CPF3490

Not authorized to specified reader.

Top

End Remote Support (ENDRMTSPT)

Where allowed to run:

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

Parameters
Examples
Error messages

Threadsafe: No

The End Remote Support (ENDRMTSPT) command varies off and deletes the line, controller's and device descriptions created by the Start Remote Support (STRRMTSPT) command. This command optionally deletes the QTILIB library created by the (STRRMTSPT) command.

Restriction: This command is not valid when you are signed-on the remote support work station.

Top

Parameters

Keyword	Description	Choices	Notes
DLTLIB	Delete library	*YES, <u>*NO</u>	Optional, Positional 1
OPTION	How to end	<u>*CNTRLD</u> , *IMMED	Optional, Positional 2

Top

Delete library (DLTLIB)

Specifies if the remote service library (QTILIB) should be deleted when running the (ENDRMTSPT) command.

*NO The remote service library (QTILIB) is not deleted.

*YES Delete the remote service library (QTILIB).

Top

How to end (OPTION)

Specifies how the remote support connection is ended.

*CNTRLD

The remote support connection ends when the connection timeout is reached.

*IMMED

The remote support connection ends immediately.

Top

Examples

```
ENDRMTSPT DLTLIB(*NO) OPTION(*IMMED)
```

This command immediately ends the remote support connection and deletes the configuration objects that have been created.

[Top](#)

Error messages

None

[Top](#)

End RPC Binder Daemon (ENDRPCBIND)

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

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

The End RPC Binder Daemon (ENDRPCBIND) command ends the Remote Procedure Call (RPC) RPCBind daemon. The RPC binder daemon job must be running to use and run Network File System (NFS) daemons and commands and some of the TI-RPC APIs.

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

- ENDNFSSVR SERVER(*RPC)

If the user attempts to end this daemon and it is not running, it will not cause the command to fail.

To determine if the RPCBind daemon is running, use the Work with Active Jobs (WRKACTJOB) command and look in the subsystem QSYSWRK for existence of the following job:

QNFSRPCD The RPCBind daemon

Restrictions

- The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.

[Top](#)

Parameters

None

[Top](#)

Examples

Example 1: Ending the RPC Binder Daemon

ENDRPCBIND

This command ends the RPC binder daemon job, if it is running.

[Top](#)

Error messages

None

[Top](#)

End Request (ENDRQS)

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

Parameters
Examples
Error messages

The End Request (ENDRQS) command ends (cancels) a previously requested operation (command). One common use of the End Request (ENDRQS) command is to cancel a request that is currently stopped at a breakpoint. This command function is also available as an option on the System Request menu.

If the End Request (ENDRQS) command cannot be processed immediately because a system function that cannot be interrupted is currently running, the command is delayed until interruption is allowed.

When a request is ended, an escape message is sent to the request processing program that is currently called at the request level being canceled. Request processing programs can be canceled. Request processing programs can monitor for the escape message so that cleanup processing can be done when the request is canceled. The static storage and open files are reclaimed for any program that was called by the request processing program. None of the programs called by the request processing program is notified of the cancel, so they have no opportunity to stop processing.

To become a request processing program, the program must receive a request message.

If the ENDRQS command is in a program, that program must become a request processor before it issues this command.

More information on how to set up a program to become a request processor is in the CL Programming book, SC41-5721.

Note: External objects that are locked by the Allocate Object (ALCOBJ) command are not unlocked (deallocated) by the canceled request.

Top

Parameters

Keyword	Description	Choices	Notes
RQSLVL	Request level	Integer, <u>*PRV</u>	Optional, Positional 1

Top

Request level (RQSLVL)

Specifies the (command) request level at which the command being canceled was entered.

***PRV** The command entered at the immediately previous level is being canceled.

request-level

Specify the request level at which the command being canceled was entered. All request levels from the level specified to the current level are canceled.

Top

Examples

Example 1: Ending a Command

```
CALL PROGA      (This is level 1)
:
Breakpoint occurs
CALL PROGB      (This is level 2)
:
Breakpoint occurs
ENDRQS          (This is level 3)
```

In this example, because RQSLVL(*PRV) is the default, the request made at level 2 is canceled. The user can then enter another command at level 2 or press F3 to show the PROGA breakpoint display again.

Example 2: Ending a Command

```
CALL PROGA      (This is level 1)
:
Breakpoint occurs
CALL PROGB      (This is level 2)
:
Breakpoint occurs
ENDRQS RQSLVL(1) (This is level 3)
```

In this example, the request made at the highest level (CALL PROGA) is canceled. Consequently, any requests made between level 1 and level 3 are also canceled.

[Top](#)

Error messages

None

[Top](#)

End S/36 Session (ENDS36)

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

Parameters
Examples
Error messages

The End System/36 (ENDS36) command allows the user to end the System/36 environment session that was started with a Start System/36 (STRS36) command.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDS36

This command immediately ends the System/36 Environment session and any programs or procedures that are running in the System/36 Environment. If the ENDS36 command is in a procedure or in a program, the statements following the command are ignored.

[Top](#)

Error messages

None

[Top](#)

End Subsystem (ENDSBS)

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

Parameters
Examples
Error messages

The End Subsystem (ENDSBS) command ends the specified subsystem (or all active subsystems) and specifies what happens to active work being processed. No new jobs or routing steps are started in the subsystem or subsystems after this command is run.

Interactive jobs that have been transferred to a job queue by the Transfer Job (TFRJOB) command are ended as part of ending the subsystem. If an initial program load (IPL) occurs while either a batch or interactive job is on a job queue (because of the TFRJOB command), that job is removed from the job queue during IPL and its job log is produced.

You can specify that the application programs running in the subsystem are given time to control end-of-job processing. If no time is given or if cleanup cannot be performed within the given time, the system performs minimal end-of-job processing, which can include:

- Closing the database files.
- Spooling the job log to an output queue.
- Cleaning up internal objects in the operating system.
- Showing the end-of-job display (for interactive jobs).
- Completing commitment control processing.

Restrictions:

1. To use this command, you must have:
 - job control (*JOBCTL) special authority.
 - object operational (*OBJOPR) and read (*READ) authority to the subsystem description associated with the specified subsystem.
2. If the controlling subsystem is being ended, because either its name or *ALL is specified for the **Subsystem** (SBS) parameter, then this command can be run only in
 - an interactive job that is in the controlling subsystem and only from a work station (associated with the interactive job) whose work station entry in the controlling subsystem description specifies *SIGNON for the **Allocation** (AT) parameter. For more information, see the Add Work Station Entry (ADDWSE) command.
 - or a batch job running in the controlling subsystem, initiated from a job queue, with the BCHTIMLMT parameter and SBS(*ALL) specified.

ENDSBS SBS(*ALL) is not allowed in a TELNET job, pass-through job, or in a workstation function job.

3. ENDSBS SBS(*ALL) is not allowed in a batch job that allows multiple threads.

Top

Parameters

Keyword	Description	Choices	Notes
SBS	Subsystem	Name, *ALL	Required, Positional 1

Keyword	Description	Choices	Notes
OPTION	How to end	<u>*CNTRLD</u> , *IMMED	Optional, Positional 2
DELAY	Controlled end delay time	0-99999, <u>*NOLIMIT</u>	Optional
ENDSBSOPT	End subsystem option	Single values: <u>*DFT</u> Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL	Optional
BCHTIMLMT	Batch time limit	5-9999, *NOMAX	Optional

Top

Subsystem (SBS)

Specifies the name of the subsystem to be ended, or it specifies that all active subsystems are to be ended.

This is a required parameter.

***ALL** All the subsystems that are currently active are being ended. All jobs are ended except the job in which this command is entered. When this value is specified, the QSYSOPR message queue should be in break delivery mode in the job issuing the end subsystem command.

name Specify the simple name of the subsystem to be ended. If the subsystem specified is the controlling subsystem, the interactive job from which the command was issued remains active. Also, if the subsystem specified is the controlling subsystem and the job that issues this command is one of two secondary jobs that are active at the work station, neither of the jobs is forced to end. The controlling subsystem does not end until you end one of the jobs (either by signing off in one job or by ending one job from the other).

Top

How to end (OPTION)

Specifies whether jobs in the subsystem are ended in a controlled manner (ending jobs in a controlled manner lets the application programs perform end-of-job processing) or immediately.

*CNTRLD

The jobs are ended in a controlled manner. This allows the programs that are running 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. The application has the amount of time specified for the DELAY parameter to complete cleanup before the job is ended.

***IMMED**

The jobs are ended 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 QENDJOBLMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs that are running are not allowed to perform any cleanup.

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.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) that is allowed to complete the controlled subsystem end operation. If this amount of time is exceeded and the end operation is not complete, any jobs still being processed in the subsystem are ended immediately.

*NOLIMIT

The amount of time in which to complete a controlled end is not limited.

0-99999

Specify the number of seconds in which the end operation is allowed to complete.

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 ENDSBS 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 ENDSBS 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.

*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 subsystem 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 subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Note: Specifying *CHGPTY and *CHGTSL will reduce the impact on other active jobs on the system, but this may cause undesirable delays if there are active workstations that were allocated to the ending subsystem. It may take longer for those workstations to have their signon screens re-displayed since the job using the display must end before the workstation is ready to be allocated to another subsystem.

Top

Batch time limit (BCHTMLMT)

Specifies the amount of time (in minutes) that the system will run in batch restricted state. This parameter is only valid when ending all subsystems from a batch job running in the controlling subsystem. Under this condition, a parameter value must be specified. When this parameter is specified, the system will be ended to the restricted state, with only the batch job running the ENDSBS command remaining active. While the system is in this restricted state, system reference code A900 3C70 is displayed. If the specified time limit is reached, the batch job will be ended and the controlling subsystem restarted.

Note: This parameter is recommended only for an application that requires no operator interaction.

*NOMAX

There is no time limit for the batch restricted function. The system will remain in the restricted state until the job ends, the Start Subsystem (STRSBS) command is used, or the Dedicated Service Tools (DST) option to end batch restricted state is used.

5-9999 Specify the time limit (in minutes) that the batch restricted function is allowed to run.

Top

Examples

```
ENDSBS SBS(QBATCH) OPTION(*CNTRLD) DELAY(60)
```

This command ends all active jobs in the QBATCH subsystem and ends the subsystem. The active jobs are allowed 60 seconds to perform application-provided end-of-job processing.

Top

Error messages

*ESCAPE Messages

CPF1001

Wait time expired for system response.

CPF1032

System ending with *CNTRLD option.

CPF1033

System ending with *IMMED option.

CPF1034

All subsystems ending with *CNTRLD option.

CPF1035

Subsystems ending with *IMMED option.

CPF1036

System powering down with *CNTRLD option.

CPF1037

System powering down with *IMMED option.

CPF1038

No authority to use command.

CPF1052

ENDSBS *ALL not allowed in current environment.

- CPF1053**
Ending controlling subsystem &1 not allowed.
- CPF1054**
No subsystem &1 active.
- CPF1055**
Subsystem &1 ending with *CNTRLD option.
- CPF1056**
Subsystem &1 already ending with *IMMED option.
- CPF1081**
Controlling subsystem already ending to a single job.
- CPF1091**
Function check occurred in system arbiter.
- CPF18C3**
Exit Point Program &1 cannot enter restricted state.

Top

End Select (ENDSELECT)

Where allowed to run:

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

Threadsafe: Yes

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

The End Select (ENDSELECT) command is used with the SELECT command to select a group of commands that are to be processed. The ENDSELECT command specifies the end of the select group that is started with an associated SELECT command. The ENDSELECT command must be specified after the last WHEN or or OTHERWISE command in the select group.

When select groups are nested, each group must have its own ENDSELECT command at its end. Every ENDSELECT command must be associated with a SELECT command; if too many ENDSELECT commands occur in the CL procedure source, a message is issued and the program is not created.

Restrictions:

- This command is valid only within a CL procedure.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

```
DCL  VAR(&NAME)  TYPE(*CHAR)  LEN(10)
:
SELECT
  WHEN  COND(&NAME *EQ *CMD)  THEN(DO)
    :  (group of CL commands)
  ENDDO
  :  (other WHEN or OTHERWISE commands)
ENDSELECT
```

The ENDSELECT command ends an active SELECT command group.

[Top](#)

Error messages

None

[Top](#)

End Service Agent (ENDSRVAGT)

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

Parameters
Examples
Error messages

The End Service Agent (ENDSRVAGT) command allows a user to end an aspect of Service Agent. The aspect to be ended is specified by the **Type (TYPE)** parameter.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type	<u>*SBSJOB</u>	Optional, Positional 1

Top

Type (TYPE)

Specifies the aspect of Service Agent to be ended.

*SBSJOB

All Service Agent monitoring jobs running in the QSYSWRK subsystem are to be ended immediately. This option will have no effect if the QSYSWRK subsystem has not been started or the Service Agent monitoring jobs have not been started in the QSYSWRK subsystem.

Top

Examples

```
ENDSRVAGT TYPE(*SBSJOB)
```

This command will end immediately all Service Agent monitoring jobs running in the QSYSWRK subsystem.

Top

Error messages

*ESCAPE Messages

CPF9899

Error occurred during processing of command.

Top

End Service Job (ENDSRVJOB)

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

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

The End Service Job (ENDSRVJOB) command ends the remote job service operation. This command stops the service operation that began when the Start Service Job (STRSRVJOB) command was entered.

Restrictions:

- If tracing or debugging is active in the serviced job when this command is entered, the remote service operation is *not* ended.
- The following user profiles have private authorities to use the command:
 - QPGMR
 - QSYSOPR
 - QSRV
 - QSRVBAS

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDSRVJOB

This command stops the service operation of the job currently being serviced.

[Top](#)

Error messages

None

[Top](#)

End System (ENDSYS)

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

Parameters
Examples
Error messages

The End System (ENDSYS) command ends most activity on the system and leaves the system in a condition in which only the console is active in the controlling subsystem. This is done so that the operator can do things like backing up the system or loading new programs. This condition is called the restricted state and is required for operations like saving the system or reclaiming storage. If two jobs are active in the controlling subsystem at the console, neither of the jobs is forced to end. The ENDSYS command cannot complete running until you end one of the jobs (either by signing off in one job or by ending one job from the other).

All active subsystems are notified that an end system operation is in process. No new jobs or routing steps can be accepted by the subsystems. This command also specifies what happens to all active work.

Interactive jobs that are transferred to a job queue by the Transfer Job (TFRJOB) command are ended as part of subsystem ending. If an initial program load (IPL) occurs while either a batch or interactive job is on a job queue (because of the TFRJOB command), that job is removed from the job queue during IPL and its job log is produced.

Restriction: This command can be entered only in an interactive job in the controlling subsystem. To use this command, the user must have job control (*JOBCTL) authority. This command is not allowed in a pass-through job or in a workstation function job.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	*CNTRLD, *IMMED	Optional, Positional 1
DELAY	Controlled end delay time	0-99999, *NOLIMIT	Optional
ENDSBSOPT	End subsystem option	Single values: *DFT Other values (up to 3 repetitions): *NOJOBLOG, *CHGPTY, *CHGTSL	Optional
CONFIRM	Confirm	*ENVVAR, *YES, *NO	Optional

Top

How to end (OPTION)

Specifies whether all active jobs are ended in a controlled manner (which lets the application programs perform end of processing) or immediately. In either case, the system performs certain job cleanup functions.

*CNTRLD

The jobs are ended in a controlled manner. This allows the programs that are running 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. The application has the amount of time specified for the DELAY parameter to complete cleanup before the job is ended.

***IMMED**

The jobs are ended 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 QENDJOBMT system value specifies a time limit. Other than by handling the SIGTERM signal, the programs that are running are not allowed to perform any cleanup.

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.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) that the controlled end operation is allowed. If this amount of time is exceeded and the end operation is not complete, any jobs still being processed are ended immediately, except for those running long-running instructions.

***NOLIMIT**

The amount of time in which to complete a controlled end is not limited.

0-99999

Specify the number of seconds in which the end operation is allowed to complete.

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 ENDSYS 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 ENDSYS 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.

***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 subsystem 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 subsystem is ending controlled. But if the DELAY time limit expires, this option will take effect immediately.

Top

Confirm (CONFIRM)

Specifies whether the request should be confirmed before the system is ended.

*ENVVAR

The value in environment variable QIBM_ENDSYS_CONFIRM is used to determine whether the request should be confirmed. If the value is set to *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 there is no confirmation.

***YES** A confirmation panel is displayed when the ENDSYS command is issued.

***NO** There is no confirmation when the ENDSYS command is issued.

Top

Examples

Example 1: Ending System Activity

ENDSYS

This command ends the system activity after all active jobs in the system are allowed to perform their own end of processes. The amount of time the end can take is not limited.

Example 2: Ending System Activity After Jobs are Ended

ENDSYS OPTION(*IMMED)

This command ends system activity after all active jobs are immediately ended.

Top

Error messages

*ESCAPE Messages

CPF1001

Wait time expired for system response.

CPF1017

ENDSYS not allowed when console powered or varied off.

CPF1032

System ending with *CNTRLD option.

CPF1033

System ending with *IMMED option.

CPF1034

All subsystems ending with *CNTRLD option.

CPF1035

Subsystems ending with *IMMED option.

- CPF1036**
System powering down with *CNTRLD option.
- CPF1037**
System powering down with *IMMED option.
- CPF1038**
No authority to use command.
- CPF1051**
Command can only be run in controlling subsystem.
- CPF1082**
Controlling subsystem already ending to single job.
- CPF1091**
Function check occurred in system arbiter.
- CPF18C3**
Exit Point Program &1 cannot enter restricted state.

Top

End TCP/IP (ENDTCP)

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

Parameters
Examples
Error messages

The End TCP/IP (ENDTCP) command ends TCP/IP processing.

Attention:

There is no confirmation display shown when ENDTCP is entered. The ENDTCP command must be used carefully. When it is used, it ends all TCP/IP processing on the iSeries that you are working on.

If OPTION(*IMMED) is specified for the ENDTCP command, the following is true:

- All TCP/IP connections are ended. This affects all currently active applications using sockets or the Pascal API.
- Unless ENDSVR(*NO) is specified, TCP/IP server jobs are ended for TELNET, FTP, TFTP, SMTP, LPD, HTTP, POP, RouteD, DHCP, DNS, DDM, BOOTP, REXEC, SNMP, DIRSRV, NSLD, INETD, MGTC, ONDMD, NETSVR, DLFM, VPN, EDRSQL, HOD, ODP, NTP, QoS, TCM, DOMINO, WEBFACING, and CIMOM.
- Agents that are currently active in the QSYSWRK subsystem are ended. See the description of the **End application servers (ENDSVR)** parameter for more information.
- All active TCP/IP interfaces are ended.

If OPTION(*CNTRL) is specified for the ENDTCP command, the following is true:

- No new open operations are allowed to TCP, UDP, or raw sockets.
- A job is submitted to the QSYSWRK subsystem that will, after the time indicated in the DELAY parameter value has expired, do an ENDTCP *IMMED operation.
- An ENDTCP OPTION(*IMMED) can be submitted at any time after issuing ENDTCP OPTION(*CNTRL). This cancels the controlled end. TCP/IP processing is ended immediately when the ENDTCP OPTION(*IMMED) is issued.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	<u>*IMMED</u> , *CNTRL	Optional, Positional 1
DELAY	Controlled end delay time	1-86400, <u>30</u>	Optional, Positional 2
ENDSVR	End application servers	<u>*YES</u> , *NO	Optional

Top

How to end (OPTION)

Specifies whether TCP/IP processing is ended in an immediate or controlled manner.

*IMMED

TCP/IP processing is ended immediately.

Attention:

The ENDTCP OPTION(*IMMED) command should be used carefully. Partially updated data may result if an application is processing data and has not completed an operation when the ENDTCP *IMMED command is issued. It is suggested that you do the following:

- Notify all users before issuing the ENDTCP command so that they can end their applications.
- Issue the ENDTCP command at a time when you know no TCP/IP traffic is occurring on the iSeries. To display the current TCP/IP traffic on the iSeries, use option 3 on the Work with TCP/IP Status (WRKTCPSTS or NETSTAT) command.

*CNTRL

TCP/IP processing is ended in a controlled manner. Applications using TCP/IP are given time to complete their processing. New application processing is not allowed. After the specified period of time elapses, the processing for ENDTCP OPTION(*IMMED) is performed.

The controlled end processing does not do any of the following:

- It does not monitor to see if all TCP/IP processing has completed before the specified period of time has elapsed.
- It does not notify an application that is actively using a TCP/IP connection that TCP/IP processing will be ended.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed in which to complete a controlled end of TCP/IP processing. After this period of time all TCP/IP processing is ended immediately.

1-86400

Specify the number of seconds in which the end operation is completed.

Top

End application servers (ENDSVR)

Specifies whether or not all TCP/IP application server jobs are ended when the End TCP/IP (ENDTCP) command ends TCP/IP processing.

Attention:

Before specifying *NO for this parameter, please consider the following:

- It is not possible to end all the TCP/IP processing on your system without affecting the applications which use TCP/IP.
- If TCP/IP processing is ended and no form of TCP/IP emulation (such as AnyNet) is active, then TCP/IP applications which are not restarted will not function correctly.
- Applications that use the Pascal API must always be ended and restarted whenever TCP/IP processing is ended and restarted.

*YES The ENDTCP command ends all TCP/IP application servers prior to ending TCP/IP processing.

***NO** The ENDTCP command does not end any TCP/IP application server jobs when it ends TCP/IP processing.

Note: ENDTCP ENDSVR(*NO) can be used to end TCP/IP processing without disturbing the operation of jobs using AnyNet. TCP/IP processing will be ended, however TCP/IP application servers that are using AnyNet will continue to function.

If both TCP/IP and AnyNet are inactive, use the End TCP/IP Server (ENDTCPSVR) command to end TCP/IP application server jobs.

Top

Examples

Example 1: Ending TCP/IP Immediately

```
ENDTCP OPTION(*IMMED)
```

This command ends all TCP/IP processing on the iSeries system immediately.

Example 2: Ending TCP/IP in a Controlled Time

```
ENDTCP OPTION(*CNTRLD) DELAY(120)
```

This command ends all TCP/IP processing after 120 seconds have expired. During this time, new TCP/IP processing is not allowed.

Example 3: Ending TCP/IP Immediately Without Ending Application Servers

```
ENDTCP OPTION(*IMMED) ENDSVR(*NO)
```

This command ends all TCP/IP processing on the iSeries 400 immediately. However, any TCP/IP application servers (FTP, SMTP, and so on) that are active are not ended when TCP/IP processing is ended.

Top

Error messages

*ESCAPE Messages

TCP1A13

Another job is starting or ending TCP/IP or IP over SNA.

TCP1A70

&1 not active.

TCP1A72

TCP/IP already ending with *CNTRLD option.

TCP1A73

Internal object damaged.

TCP1A74

Error occurred submitting job.

TCP1A77

&1 completed successfully; however errors occurred.

TCP9999

Internal system error in program &1.

End TCP/IP Abnormal (ENDTCPABN)

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

Parameters
Examples
Error messages

The End TCP/IP Abnormal (ENDTCPABN) command is used to force TCP/IP processing to terminate. It may only be used after attempting to use the End TCP/IP (ENDTCP) command with OPTION(*IMMED) specified.

The ENDTCPABN command cannot be issued until either the ENDTCP command has completed or until 10 minutes have passed following the request for TCP/IP immediate ending. This allows sufficient time for normal TCP/IP ending functions to occur.

Successful completion of ENDTCPABN processing should permit TCP/IP to be restarted without a system IPL. Issuing the ENDTCPABN command does not directly affect system termination. The next system end will **not** be marked as ABNORMAL as a result of ENDTCPABN processing.

Restrictions:

- This command is shipped with public *EXCLUDE authority. The QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles are shipped with private authorities to use this command.
- Users cannot run the ENDTCPABN command until ten minutes *after* running the ENDTCP command with OPTION(*IMMED) specified.

Top

Parameters

None

Top

Examples

ENDTCPABN

This command forces TCP/IP processing to end.

Top

Error messages

*ESCAPE Messages

TCP1A66

ENDTCPABN not allowed at this time. Reason &1.

Top

End TCP/IP Connection (ENDTCPCNN)

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

Parameters
Examples
Error messages

The End TCP/IP Connection (ENDTCPCNN) command is used to end a Transmission Control Protocol/Internet Protocol (TCP/IP) connection. This command ends a connection immediately and should be used only when a normal end is not possible.

Note: The ENDTCPCNN command is usually used by specifying option 4 on the Work with TCP/IP Connection Status list of the WRKTCPCNN (NETSTAT) display. The ENDTCPCNN command is provided as a separate command to give system administrators control over this function. By limiting the authority to the ENDTCPCNN command, the system administrator limits which users can end TCP/IP connections without restricting access to the NETSTAT utility.

Top

Parameters

Keyword	Description	Choices	Notes
PROTOCOL	Protocol	*UDP, *TCP	Required, Positional 1
LCLINTNETA	Local internet address	<i>Character value, *</i>	Required, Positional 2
LCLPORT	Local port	1-65535	Required, Positional 3
RMTINTNETA	Remote internet address	<i>Character value, *</i>	Optional, Positional 4
RMTPORT	Remote port	1-65535, *	Optional, Positional 5

Top

Protocol (PROTOCOL)

Specifies the protocol used by the connection that is to be ended. The protocol value must be either *TCP or *UDP.

***UDP** The connection was created for use with the User Datagram Protocol (UDP).

***TCP** The connection was created for use with the Transmission Control Protocol (TCP).

Top

Local internet address (LCLINTNETA)

Specifies the local internet address of the connection to end. This parameter is required for both UDP and TCP.

* The remote port number was left unspecified when this connection was opened.

character-value

Specify the local internet address. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Local port (LCLPORT)

Specifies the local port number of the connection to end. This parameter is required for both UDP and TCP. A decimal number identifying a local port must always be specified for this command.

1-65535

Specify the local port number of the connection to end.

Attention:

Ports 1 through 1024 are reserved for use by system-supplied TCP/IP applications. If the user specifies ports 1 through 1024, it can affect the operation of those applications.

Top

Remote internet address (RMTINTNETA)

Specifies the remote internet address of the connection to end. This parameter is required for TCP.

* The remote port number was left unspecified when this connection was opened.

character-value

Specify the remote internet address. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Remote port (RMTPORT)

Specifies the remote port number of the connection to end.

This parameter is required for TCP.

* The remote port number was left unspecified when this connection was opened.

1-65535

Specify the remote port number of the connection to end.

Top

Examples

Example 1: Ending a TCP Connection

```
ENDTCPCNN  PROTOCOL(*TCP)  LCLINTNETA('9.5.1.109')
           LCLPORT(13054)
           RMTINTNETA('9.130.28.144')  RMTPORT(23)
```

This command ends the TCP connection between local port 13054 for local internet address 9.5.1.109 and remote port 23 for remote internet address 9.130.28.144. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Example 2: Closing a UDP Socket

```
ENDTCPCNN  PROTOCOL(*UDP)  LCLINTNETA('9.130.28.144')
            LCLPORT(596)
```

This command closes the UDP socket using local port 596 and local internet address 9.130.28.144. The TCP/IP protocol stack ends all activity on the connection and returns the resources to the free storage pools.

Example 3: Ending a LISTEN State TCP Socket

```
ENDTCPCNN  PROTOCOL(*TCP)  LCLINTNETA(*)  LCLPORT(5023)
            RMTINTNETA(*)  RMTPORT(*)
```

This command ends the TCP socket that is listening on local port 5023. The application that created this socket did not specify a local internet address. The socket is closed and the local port is made available for use by another application.

Top

Error messages

*ESCAPE Messages

TCP2670

Not able to complete request. TCP/IP services are not available.

TCP3B01

Not able to end TCP connection &3 &4, &5 &6.

TCP3B02

Not able to close UDP socket &3 &4.

TCP9999

Internal system error in program &1.

Top

End TCP/IP Interface (ENDTCPIFC)

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

Parameters
Examples
Error messages

The End TCP/IP Interface (ENDTCPIFC) command is used to end a Transmission Control Protocol/Internet Protocol (TCP/IP) interface. When an interface is ended with this command, datagrams addressed to the IP addresses associated with this interface will no longer be accepted. However, the operation of any other TCP/IP or IP over SNA interface that is using the same line description as the the interface being ended is not affected.

This command can be used to end an interface that was previously started by the Start TCP/IP Interface (STRTCPIFC) or Start TCP (STRTCP) command.

Use this command to end all TCP/IP interfaces prior to ending TCP/IP. Also, use this command to end all TCP/IP interfaces prior to varying off a device, controller, or line associated with TCP/IP. Failure to do so may cause unpredictable results.

Warning: Temporary Level 2 Header

Warning: Temporary Level 3 Header

Notes on Route to Interface Binding

Interfaces define direct routes to networks or subnetworks that this iSeries is directly attached to. Routes define the indirect routes. An indirect route defines the next hop on the path to a network or subnetwork that this iSeries is not directly attached to.

Indirect routes are bound to interfaces using a best match first algorithm. This algorithm is based on the state of the interface and on the type of service (TOS) specified for the route and interface. When ending an interface, the routes associated with the interface can move to another existing active interface. This provides the widest available level of connectivity.

Top

Parameters

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

Top

Internet address (INTNETADR)

Specifies the internet address of an interface that had previously been added to the TCP/IP configuration with the Add TCP/IP Interface (ADDTCPIFC) command and which had been previously started by the STRTCPIFC or STRTCP command. The internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An internet address is not valid if it has a value of

all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address. If the internet address is entered from a command line, the address must be enclosed in apostrophes.

Top

Examples

Example 1: Ending an X.25 Interface

```
ENDTCPIFC INTNETADR('9.5.11.125')
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the internet address 9.5.11.125.

Example 2: Ending a Token-Ring Interface

```
ENDTCPIFC INTNETADR('156.93.81.7')
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the internet address 156.93.81.7.

Top

Error messages

*ESCAPE Messages

TCP1B15

Line description &2 unusable. Internal errors encountered.

TCP1B61

Unable to determine if &1 interface ended.

TCP1B62

Cannot determine if &1 interface ended.

TCP1B65

&2 interface not ended. Reason &1.

TCP1B72

&1 interface not ended. &1 interface is not active.

TCP1B73

&1 interface not ended. &1 interface not defined in the TCP/IP configuration.

TCP1B74

&1 interface not ended. Line description &2 not found.

TCP1B85

Unable to submit request to end interface &1.

TCP265F

INTNETADR parameter value &2 not valid.

TCP265F

INTNETADR parameter value &2 not valid.

TCP9999

Internal system error in program &1.

Top

End Point-to-Point TCP/IP (ENDTCPPTP)

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

Parameters
Examples
Error messages

The End Point-to-Point TCP/IP (ENDTCPPTP) command is used to end a point-to-point TCP/IP session job. A session job operates in one of two possible modes:

1. Answer mode sessions (*ANS) allow a remote system to contact this iSeries and establish a point-to-point TCP/IP session.
2. Dial mode sessions (*DIAL) allow this iSeries to contact a remote system that supports point-to-point TCP/IP.

Note: Profiles of linetype *PPP can be ended with this command but any configuration of *PPP profiles must be done using the iSeries Navigator graphical user interface.

The TCP/IP point-to-point session jobs run in the QSYSWRK subsystem.

Top

Parameters

Keyword	Description	Choices	Notes
CFGPRF	Configuration profile	<i>Character value</i> , *ALL	Required, Positional 1
OPRMODE	Operating mode	* <u>ANY</u> , *ANS, *DIAL	Optional, Positional 2

Top

Configuration profile (CFGPRF)

Specifies which point-to-point TCP/IP sessions job or jobs should be ended.

This is a required parameter.

***ALL** All currently active point-to-point TCP/IP sessions jobs operating in the mode specified by the OPRMODE parameter are ended.

generic-name

Specify the generic name of the point-to-point TCP/IP configuration profile to be ended. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all profiles with names that begin with the generic name are ended. If an asterisk is not included, the name is assumed to be a complete point-to-point TCP/IP configuration profile name. All currently active point-to-point TCP/IP session jobs using the profiles indicated and operating in the mode specified by the OPRMODE parameter are ended.

name Specify the name of a TCP/IP point-to-point configuration profile. The active point-to-point session job using this profile is ended.

Top

Operating mode (OPRMODE)

Specifies the operating mode of the TCP/IP point-to-point session job to be ended.

***ANY** Any point-to-point TCP/IP session job that matches the configuration profile name specified on the CFGPRF parameter is ended, regardless of operating mode.

***ANS** The operating mode of the session to be ended is *ANS. All *ANS point-to-point TCP/IP session jobs that are currently active that match the specified CFGPRF parameter will be ended.

***DIAL** The operating mode of the session to be ended is *DIAL. All *DIAL point-to-point TCP/IP session jobs that are currently active that match the specified CFGPRF parameter will be ended.

Top

Examples

Example 1: End a TCP/IP Point-To-Point Session Job

```
ENDTCPPTP  CFGPRF(DIALPRF)
```

This command ends the point-to-point TCP/IP session job that is using configuration profile DIALPRF. The operating mode (OPRMODE) value will default to *ANY so the operating mode is not used in deciding whether to end the session job.

Example 2: End All Answer (*ANS) Mode TCP/IP Point-To-Point Session Jobs

```
ENDTCPPTP  CFGPRF(*ALL)  OPRMODE(*ANS)
```

This command ends all active or activating point-to-point answer mode (*ANS) TCP/IP session jobs.

Example 3: End All TCP/IP Point-To-Point Session Jobs

```
ENDTCPPTP  CFGPRF(*ALL)
```

This command ends all active or activating point-to-point TCP/IP session jobs.

Example 4: End All TCP/IP Point-To-Point Session Jobs Starting with XYZ.

```
ENDTCPPTP  CFGPRF(XYZ*)
```

This command ends all active or activating point-to-point TCP/IP session jobs that have profiles that begin with XYZ.

Example 5: End an Answer Mode TCP/IP Point-To-Point Session Job using a Specific Profile Name

```
ENDTCPPTP  CFGPRF(DIALPRF)  OPRMODE(*ANS)
```

This command will end the point-to-point TCP/IP session job using profile DIALPRF if this profile is defined to run in answer mode. If the profile is defined to run in dial mode then no action will be taken.

Top

Error messages

*ESCAPE Messages

TCP1A1F

Cannot process request while &3/&2/&1 using &6.

TCP8205

Required object &2/&1 type *&3 not found.

TCP8209

ENDTCPPTP &1 &3 for &6/&5/&4 completed. &10 of &11 sessions ended.

Top

End TCP/IP Server (ENDTCPSVR)

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

Parameters
Examples
Error messages

The ENDTCPSVR command is used to end the TCP/IP application server jobs that are specified in the SERVER parameter. If the jobs have any current active connections, these connections are ended immediately. If the ENDTCPSVR command is used to end a server that is not active, a diagnostic message may be returned.

The End TCP/IP Server command can only be used when TCP/IP is fully operational. The interface server job QTCPIP must be available. When the iSeries is in restricted state, this command is not allowed.

Additional servers can automatically be added to the list of servers that ENDTCPSVR will support by using the ADDTCPSVR (Add TCP/IP Server) CL command.

Restrictions:

- This command is conditionally threadsafe. This command calls different programs to process each type of TCP/IP server. If the programs being called are threadsafe, this command is threadsafe.

Top

Parameters

Keyword	Description	Choices	Notes
SERVER	Server application	Single values: *ALL Other values (up to 300 repetitions): <i>Character value</i>	Optional, Positional 1
HTTPSVR	HTTP server	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Name</i> , *ADMIN	
DNSSVR	DNS server	Single values: *ALL Other values: <i>Element list</i>	Optional
	Element 1: Server instance	<i>Name</i>	
TCMSVR	TCM server	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Instance name	<i>Character value</i> , *ALL	
TOMCATSVR	ASFTOMCAT server	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Server instance name	<i>Character value</i> , *ALL	

Top

Server application (SERVER)

Specifies which of the TCP/IP application server jobs is to be ended by this command.

Additional TCP/IP servers could also be available if they were added by running the Add TCP/IP Server (ADDTCPSVR) command.

For a list of all supported values for this parameter, you can press F4 (Prompt) for this parameter when prompting this command.

Single values

***ALL** All of the TCP/IP server jobs are ended.

Other values (up to 300 repetitions)

*ASFTOMCAT

The Apache Software Foundation (ASF) Tomcat server is ended.

*BOOTP

The bootstrap protocol (BOOTP) server is ended.

*CIMOM

The Common Information Model Object Manager (CIMOM) server is ended.

***DBG** The Debug server is ended.

*DDM

The Distributed Data Management (DDM) server job is ended.

*DHCP

The Dynamic Host Configuration Protocol (DHCP) server job is ended.

*DIRSRV

The LDAP directory services (DIRSVR) server job is ended.

*DLFM

The DataLink File Manager (DLFM) server job is ended.

***DNS** The Domain Name System (DNS) server job is ended unless:

- You specify a specific server instance name on the DNSSVR parameter. When you specify a specific server instance, only that instance is ended.

To end all instances of the DNS server, specify one of the following:

```
ENDTCPSVR  SERVER(*DNS)
ENDTCPSVR  SERVER(*DNS)  DNSSVR(*ALL)
```

*DOMINO

The Lotus Domino (DOMINO) server is ended.

*EDRSQL

The Extended Dynamic Remote SQL (EDRSQL) server is ended.

***FTP** All File Transfer Protocol (FTP) server jobs are ended.

***HOD** The Host On Demand (HOD) server is ended.

*HTTP

All instances of the HyperText Transfer Protocol (HTTP) server are ended unless:

- You specify a specific server instance name on the HTTPSVR parameter. When you specify a specific server instance, only that instance is ended.
- You specify the *ADMIN value on the HTTPSVR parameter. When you specify HTTPSVR(*ADMIN), only the administration server is ended.

To end all instances of the HTTP server, specify one of the following:

```
ENDTCPSVR  SERVER(*HTTP)
ENDTCPSVR  SERVER(*HTTP)  HTTPSVR(*ALL)
```

The HTTP server is also known as the IBM HTTP Server.

- *INETD**
The Internet Daemon (INETD) server is ended.
- *LPD** All line printer daemon (LPD) servers are ended.
- *MGTC**
The Management Central (MGTC) server is ended.
- *NETSVR**
The NetServer (NETSVR) server is ended.
- *NSLD**
The Network Station Login Daemon (NSLD) server is ended.
- *NTP** All Simple Network Time Protocol (SNTP) services servers are ended.

Note: If both client and server SNTP services have been started, running the ENDTCPSPVR command specifying SERVER(*NTP) will end both client and server services. If you wanted to end just the client or just the server SNTP service, you will need to run the Start TCP/IP Server (STRTCPSPVR) command again, specifying SERVER(*NTP) and NTPSRV(*CLIENT) or NTPSRV(SERVER).
- *ODPA**
The On-Demand Platform Authentication (ODPA) server is ended.
- *ONDMD**
The OnDemand (ONDMD) server job is ended.
- *POP** All Post Office Protocol (POP3) mail server jobs are ended.
- *QOS** The Quality of Service (QOS) server is ended.
- *REXEC**
All Remote Execution (REXEC) servers are ended.
- *ROUTED**
The Router Daemon (ROUTED) server is ended.
- *SMTP**
All jobs associated with Simple Mail Transfer Protocol (SMTP) in the QSYSWRK subsystem are ended. The bridge job in the QSNADS subsystem is not ended.
- *SNMP**
All jobs associated with the Simple Network Management Protocol (SNMP) agent in the QSYSWRK subsystem are ended.
- *TCM** The Triggered Cache Manager (TCM) server is ended unless:
 - You specify a specific server instance name on the TCMSVR parameter. When you specify a specific server instance, only that instance is ended.

To end all instances of the TCM server, specify:

```
ENDTCPSPVR  SERVER(*TCM)  TCMSVR(*ALL)
```
- *TELNET**
All TELNET server jobs are ended.
- *TFTP** All Trivial File Transfer Protocol (TFTP) server jobs are ended.
- *VPN** The Virtual Private Network (VPN) server is ended.
- *WEBFACING**
The WebFacing server is ended.

Top

HTTP server (HTTPSVR)

Specifies the name of the HTTP server instance to end. The SERVER parameter specified must be *HTTP or this parameter is ignored.

If multiple HTTP server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***ALL** All instances of the HTTP server that are currently running are ended.

***ADMIN**

The Administration Server is ended. The Administration Server is an instance of the HTTP server that allows administration of certain iSeries functions using a Web browser.

name Specify the name of the HTTP server instance to be ended.

Top

DNS server (DNSSVR)

Specifies the name of the DNS server instance to end. The SERVER parameter specified must be *DNS or this parameter is ignored.

If multiple DNS server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***ALL** All instances of the DNS server that are currently running are ended.

name Specify the name of the DNS server instance to be ended.

Top

TCM server (TCMSVR)

Specifies the name of the TCM server instance to end. The SERVER parameter specified must be *TCM or this parameter is ignored.

If multiple TCM server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***NONE**

No instances of the TCM server that are currently running are ended.

***ALL** All instances of the TCM server that are currently running are ended.

name Specify the name of the TCM server instance to be ended.

Top

ASFTOMCAT server (TOMCATSVR)

Specifies the name of the Tomcat server instance to end. The SERVER parameter specified must be *ASFTOMCAT or this parameter is ignored.

If multiple Tomcat server instances have been defined, you can choose to end all instances, or end one specific instance by specifying the instance name to be ended.

***NONE**

No instances of the Tomcat server that are currently running are ended.

***ALL** All instances of the Tomcat server that are currently running are ended.

name Specify the name of the Tomcat server instance to be ended.

Top

Examples

Example 1: Ending All TCP/IP Servers

```
ENDTCPSVR SERVER(*ALL)
```

This command ends all active TCP/IP application server jobs.

Example 2: Ending the LPD Servers

```
ENDTCPSVR SERVER(*LPD)
```

This command ends the TCP/IP LPD application server jobs.

Example 3: Ending a Specific HTTP Server Instance

```
ENDTCPSVR SERVER(*HTTP) HTTPSVR(http1)
```

This command ends the TCP/IP HTTP application server instance named 'http1'.

Example 4: Ending a Specific DNS Server Instance

```
ENDTCPSVR SERVER(*DNS) DNSSVR('dns1')
```

This command ends the TCP/IP DNS application server instance named 'dns1'.

Top

Error messages

*ESCAPE Messages

CPF3894

Cancel reply received for message &1.

TCP1A0A

&1 ended abnormally. Reason code is &2.

TCP1A11

&1 failed.

TCP1A77

&1 completed successfully; however errors occurred.

Top

End TIE Session (ENDTISSN)

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 End Technical Information Exchange Session (ENDTISSN) command allows you to disconnect the communications line used for TIE batch commands. This command must follow other TIE batch commands.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDTISSN

This command ends the TIE function by disconnecting the communications line used for TIE batch commands.

[Top](#)

Error messages

None

[Top](#)

End Trace (ENDTRC)

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

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

The End Trace (ENDTRC) command ends a trace session that was started by a STRTRC (Start Trace) command.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service Trace function of OS/400 through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- If DTAOPT(*LIB) is specified, you must have authority to the library and the database files within that library where the trace data is stored.
- If PRTRC(*YES) is specified, you must have authority to the PRTRC (Print Trace) command.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	<i>Name</i> , *PRV	Required, Positional 1
DTAOPT	Data option	*LIB, *DLT	Optional
DTALIB	Data library	<i>Name</i> , *CURLIB	Optional
RPLDTA	Replace data	*YES, *NO	Optional
PRTRC	Print trace data	*NO, *YES	Optional

[Top](#)

Session ID (SSNID)

Specifies a session identifier for the trace to be ended. This name must match the session identifier of a trace that had been previously started and is still active.

This is a required parameter.

***PRV** The trace session most recently started by the same user who is running this ENDTRC command will be ended. For example, if the job running the ENDTRC command is running under user profile BOB, the last trace session started under user profile BOB is ended.

name Specify the session identifier of the trace to be ended.

[Top](#)

Data option (DTAOPT)

Specifies whether the trace data that has been collected is stored into database files or if the trace data is deleted.

***LIB** The trace data will be copied into database files. The PRTTRC parameter on this command or the Print Trace (PRTTRC) command can be used to format and print the data.

***DLT** The trace data will be deleted from the internal buffers where it was collected.

Top

Data library (DTALIB)

Specifies the name of the library in which the trace data will be stored. A set of database files will be created in this library to contain the trace data. The files will be created if they do not already exist.

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

***CURLIB**

The trace data is stored in 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 to contain the trace data files. The library must already exist when the ENDTRC command is run.

Top

Replace data (RPLDTA)

Specifies whether trace data that was collected by a previous trace session with the same session identifier is replaced with new trace data. This is determined by checking if the set of database files where the trace data is to be stored already have file members with the same name as the specified trace session identifier (SSNID parameter).

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

***YES** If trace data already exists with the specified session identifier, the old trace data is lost and replaced by the new trace data.

***NO** If trace data already exists for the specified session, an error message is sent to the user.

Top

Print trace data (PRTTRC)

Specifies whether trace data is formatted and printed after it is stored in the trace database files.

Note: This parameter is valid only if *LIB is specified for the **Data option (DTAOPT)** parameter.

***NO** The PRTTRC (Print Trace) command is not run as part of this command.

***YES** The PRTTRC (Print Trace) command is run after the trace data has been stored in the database files.

Top

Examples

Example 1: End Most Recently Started Trace

```
ENDTRC  SSNID(*PRV)
```

This command ends the trace session started most recently by the same user who is running the ENDTRC command. The trace data will be stored in a set of files in the current library of the job, or QGPL if there is no current library for the job.

Example 2: End a Trace and Delete Trace Data

```
ENDTRC  SSNID(DCG1)  DTAOPT(*DLT)
```

This command ends the trace session DCG1 and deletes the trace data.

Top

Error messages

*ESCAPE Messages

CPC3923

ENDTRC session ID &1 successfully saved into library &2.

CPC3924

ENDTRC session ID &1 successfully deleted.

CPF39CA

Trace session ID &1 not found.

CPF39CB

Trace session ID &1, in library &2, data exists. Specify RPLDTA(*YES).

CPF98A2

Not authorized to &1 command.

CPF39D3

Unable to start/end the trace.

Top

End Trap Manager (ENDTRPMGR)

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

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

Use the End Trap Manager (ENDTRPMGR) command to end the OS/400 Simple Network Management Protocol (SNMP) trap manager job.

[Top](#)

Parameters

None

[Top](#)

Examples

ENDTRPMGR

This command ends the OS/400 SNMP Manager Framework trap manager job.

[Top](#)

Error messages

*ESCAPE Messages

CPFA805

Trap manager job not active or being ended.

[Top](#)

End Writer (ENDWTR)

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

Parameters
Examples
Error messages

The End Writer (ENDWTR) command ends the specified spooling writer and makes its associated output device available to the system. The writer can be ended immediately or in a controlled manner. If ended immediately, the writer stops writing the file and the file is made available again on the output queue. If ended in a controlled manner, the writer finishes writing the current file (or a copy of a file), or it finishes printing a page of the file, before it is ended.

Top

Parameters

Keyword	Description	Choices	Notes
WTR	Writer	<i>Name</i> , *SYSVAL, *ALL	Required, Positional 1
OPTION	When to end writer	* <u>CNTRLD</u> , *IMMED, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

This is a required parameter.

Specifies the name of the spooling writer being stopped. The writer's output device will then be available to the system.

The possible values are:

***ALL** Specifies that all writers that are started are to stop.

***SYSVAL**

Specifies that the writer started to the system default printer is to stop.

writer-name

Specify the name of the writer to end.

Top

When to end writer (OPTION)

Specifies when the writer should stop processing.

The possible values are:

***CNTRLD**

The spooling writer stops processing in a controlled manner. Output stops at the end of the spooled file (or copy of a file) currently being written to an output device.

***IMMED**

The writer stops processing immediately. The spooled file that is currently printing remains on the output queue.

***PAGEEND**

The writer is stopped after processing of the current buffer. This value is valid only if the spooling writer is a printer writer.

Top

Examples

```
ENDWTR WTR(PRINTER)
```

This command stops the writer named PRINTER at the end of the spooled file whose output is being printed, and then releases the device to the system.

Top

Error messages

***ESCAPE Messages**

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1340

Job control function not performed.

CPF1352

Function not done. &3/&2/&1 in transition condition.

CPF1842

Cannot access system value &1.

CPF3313

Writer &1 not active nor on job queue.

CPF3330

Necessary resource not available.

CPF3331

Not authorized to control writer &3/&2/&1.

CPF3339

Previous end request to writer &3/&2/&1 pending.

CPF3438

*PAGEEND not valid for writer &3/&2/&1.

Top

Remove Link (ERASE)

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

Parameters
Examples
Error messages

The Remove link (ERASE) command removes the link to the specified object. If this is the only hard link to the object, the object is removed when no longer in use. The object can be removed even if a symbolic link to it exists. The symbolic link remains until it is removed.

This command is an alias for the Remove link (RMVLNK) command and can also be issued using the following alternative command names:

- DEL
- RMVLNK

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

Restrictions:

1. In the "root" (/), QOpenSys, and user-defined file systems, the user must have write, execute (*WX) authority to the directory containing the object. If a hard link is to be unlinked, the user must also have object existence (*OBJEXIST) authority to the object.
2. In the QDLS file system, the user must have all (*ALL) authority to the object and execute (*X) authority to the parent directory.
3. The user must have *X authority to each directory in the path.
4. See the iSeries Security Reference, SC41-5302 book for the authority requirements for other file systems.
5. A user cannot unlink 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:
 - a. The user is the owner of the object.
 - b. The user is the owner of the directory.
 - c. The user has all object (*ALLOBJ) special authority.
6. A directory cannot be unlinked.
7. The link to a file cannot be removed if the file is a DataLink column in an SQL table and where a row in that SQL table references this file.
8. The restrictions listed above are for the OS/400 objects of the types *DDIR, *DSTMF, *SOCKET, *STMF, and *SYMLNK.

QSYS.LIB and independent ASP QSYS.LIB File System Differences

1. If this command is to be used to remove links for an object that is in these file systems, additional restrictions may apply. To identify these restrictions, see the delete command for the object to be removed. In general, the name of this command is formed using the OS/400 object type value, from the character * is removed, and add the verb DLT to the beginning. For example, to delete an alert table, which has the object type value of *ALRTBL, see the Delete Alert Table (DLTALRTBL) command for any additional restrictions.

However, there are exceptions to this rule. For example, to delete a compiler unit, which has the object type value of *MODULE, see the Delete Module (DLTMOD) command for any additional restrictions.

For a description of the object types, see the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

2. In these file systems, libraries and database files cannot be deleted using the Remove Link (RMVLNK or alias DEL or ERASE) command. However, these objects can be deleted using the Remove Directory (RMVDIR or alias RMDIR or RD) command.
3. The following object types cannot be deleted using another command: *EXITRG, *IGCSRT, *JOBSCD, *PRDAVL, *QRYDFN, *RCT.

QDLS File System Differences

1. If this command is to be used to remove links for an object that is in this file system, additional restrictions may apply. To identify these restrictions, see the description of the Delete Document Library Object (DLTDLO) command.

Top

Parameters

Keyword	Description	Choices	Notes
OBJLNK	Object link	<i>Path name</i>	Required, Positional 1

Top

Object link (OBJLNK)

Specifies the path name of the object to unlink. Multiple links can be removed with a name pattern.

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 "CL concepts and reference" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

Examples

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

Example 1: Removing an Object Link

```
RMVLNK  OBJLNK('PAY')
```

This command removes a link named PAY.

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.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0A9

Object not found. Object is &1.

CPFA0AB

Operation failed for object. Object is &1.

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0B2

No objects satisfy request.

CPFA0BD

&1 links removed. &2 links failed.

Top

Export a Program Symbol (EXPORT)

Parameters
Examples
Error messages

The Export a Program Symbol (EXPORT) binder definition statement defines an export in a service program export block.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
SYMBOL	Exported symbol name	<i>Character value</i>	Required, Positional 1

[Top](#)

Exported symbol name (SYMBOL)

Specifies the symbol to be exported. The symbol can be enclosed in apostrophes, quotation marks, or expressed without the delimiting marks.

This is a required parameter.

character-value

Specify the name of the program external variable or procedure to be exported.

[Top](#)

Examples

```
EXPORT SYMBOL('ExtVar2')
```

This binder definition statement defines **ExtVar2** as an exported symbol in a service program export block.

[Top](#)

Error messages

None

[Top](#)

Change NFS Export (EXPORTFS)

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

Parameters
Examples
Error messages

The Change Network File System Export (EXPORTFS) command adds directory names to (exports) or removes directory names from (unexports) the list of directory trees that are currently exported to Network File System (NFS) clients. The flags in the OPTIONS list indicate what actions the EXPORTFS command should perform.

A list of directories and options for exporting the directory and its contents is stored in the */etc/exports* file. The EXPORTFS command allows the user to export all of the directory trees specified in the */etc/exports* file using the -A flag, or to export a single directory tree by specifying the directory name. When the directory tree to be exported exists in the */etc/exports* file, the user can export it with the options specified there, or one can use the -I flag to override the options, specifying the new options on the EXPORTFS command.

This user can also export a directory tree not previously defined in the */etc/exports* file by providing the options for it on the EXPORTFS command. The user can unexport directory trees by using the -U flag on the EXPORTFS command.

The user can also add, change, or remove export entries in the */etc/exports* file by using the -F flag.

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

- CHGNFSEXP

For more information about Network File System commands, see the OS/400 Network File System book, SC41-5714.

Restrictions:

1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
2. The user must have execute (*X) authority to each directory in the path name prefixes.
3. When the -F flag is specified and the */etc/exports* file does not exist, the user must have write, execute (*WX) authority to the */etc* directory.
4. When the -F flag is specified and the */etc/exports* file does exist, the user must have read, write (*RW) authority to the */etc/exports* file and *X authority to the */etc* directory.
5. Mixed CCSID encoding schemes are not supported. Specified CCSIDs must be single-byte character set (SBCS) or pure double-byte character set (DBCS).

Top

Parameters

Keyword	Description	Choices	Notes
OPTIONS	NFS export options	<i>Character value</i> , <u>*DFT</u>	Optional, Positional 1
DIR	Directory	<i>Path name</i>	Optional, Positional 2

Keyword	Description	Choices	Notes
HOSTOPT	Host options	Single values: *DFT Other values (up to 10 repetitions): <i>Element list</i>	Optional
	Element 1: Host name	<i>Character value</i>	
	Element 2: Data file CCSID	0-65535, *BINARY , *ASCII, *JOBCCSID	
	Element 3: Path name CCSID	0-65535, *ASCII , *JOBCCSID	
	Element 4: Force synchronous write	*SYNC , *ASYNC	

Top

NFS export options (OPTIONS)

The export options list contains some flags followed optionally by a list containing a character string of characteristics for the directory tree to be exported.

Each flag consists of a minus "-" followed by a character. The flags are separated by spaces. Only certain combinations of flags are allowed. If an invalid combination is detected, an error is returned.

Note: A value (other than *NONE) must be specified for either the OPTIONS or **Directory (DIR)** parameter. Both OPTIONS and DIR can be specified so long as '-A' is not part of the options list specified for the OPTIONS parameter.

***DFT** The default value for the options string is:

'-A'

options-flags

- A** This is the "all" flag and it indicates that all entries in the */etc/exports* file are to be processed. The following flag combinations have special significance:
 - A and not -U**
This will export every entry in the */etc/exports* file (making them available to NFS clients).
 - A and -U**
This will unexport every entry that is currently exported (making them unavailable to NFS clients). This makes no reference to the contents of the */etc/exports* file.
 - A and the DIR parameter**
This combination is not allowed.
 - A and (-I or -F or -O)**
These combinations are not allowed.
- I** This is the "ignore" flag and it indicates, for the directory tree specified in the DIR parameter, how the export characteristics are determined. The following flag combinations have special significance:
 - I and -O**
The export characteristics specified on the -O flag are used, and the definitions listed in the */etc/exports*, if they exist, are ignored.
 - not -I and not -O**
Either the export characteristics listed for the entry in the */etc/exports* file are used,

or, if there are no options in that file, the default options are assumed. See the -O flag description for the list of default options.

-I and (-A or -U)

These combinations are not allowed.

-U This is the "unexport" flag and it indicates that the specified directory tree entered in the DIR parameter is to be unexported (made unavailable to NFS clients). The following flag combinations have special significance:

-U and -A

This will unexport every entry that is currently exported (making them unavailable to NFS clients). This makes no reference to the contents of the */etc/exports* file.

-U and -F

The entry referenced in the DIR parameter is removed from the */etc/exports* file (if it exists) in addition to being unexported (making it unavailable to NFS clients).

-U and (-I or -O)

These combinations are not allowed.

-F This is the "file" flag and it requires the DIR parameter. The following flag combinations have special significance:

-F and -U

The entry referenced in the DIR parameter is removed from the */etc/exports* file (if it exists) in addition to being unexported (making it unavailable to NFS clients).

-F and not -U and not -O

The specified directory tree entered in the DIR parameter is to be exported (made available to NFS clients). In addition, an entry for this directory tree entered in the DIR parameter will be added to the */etc/exports* file. If the entry already exists in the file, it will be replaced with this new export entry. If the file does not exist, it will be created and the export entry will be added to it. Note that the "ignore" flag -I is implied when the "file" flag -F is specified without the "unexport" flag -U. Since the "options" flag -O is not specified, the default options are assumed. See the -O flag description for the list of default options.

-F and not -U and -O

The specified directory tree entered in the DIR parameter is to be exported (made available to NFS clients). In addition, an entry for this directory tree entered in the DIR parameter will be added to the */etc/exports* file. If the entry already exists in the file, it will be replaced with this new export entry. If the file does not exist, it will be created and the export entry will be added to it. Note that the "ignore" flag -I is implied when the "file" flag -F is specified without the "unexport" flag -U. All export characteristic options provided with the "options" flag -O are stored in the */etc/exports* file as given on the command.

-F and -A

This combination is not allowed.

Note: Successful use of the -F flag will cause the contents of the */etc/exports* file to be replaced completely such that it reflects the changes, additions, or deletions caused by the -F flag. Any unrelated existing entries are copied, however ALL comments in the */etc/exports* file will be lost as a result of using the -F flag.

- E This is the "escape message" flag and it indicates that an escape message should be issued if the command fails for any of the exports attempted.
- O This flag specifies the export characteristics for the directory tree that is to be exported (made available to NFS clients). The options list following the -O flag list consists of options separated by commas. Some options are followed by an equal '=' and a value (or list of values separated by colons ':'). The options list may contain spaces. If an option is not specified, the default value for that option will be used. The -O flag is only valid when either the "ignore" flag -I or the "file" flag -F is specified.

If options are required and the -O flag is not specified, the following are the default options.

- 'RW=' All host names have read-write access to the directory tree.
- ANON=UID associated with the profile *QNFSANON*.
- Requests to bits in the mode other than the permission bits are allowed.
- 'ROOT=' Root access is not allowed for any hosts.
- 'ACCESS=' All clients are allowed to mount the directory.

The following are the available options and their descriptions.

RO Specifies the protection for the exported directory tree. If RO is specified, the directory tree is exported allowing only read-only access to the directory and its contents. If it is not specified, read-write access is allowed to the directory and its contents.

RW=[HOSTNAME[:HOSTNAME]](...)

Specifies the host name or host names which will be allowed read-write access to the exported directory and its contents. For host names not specified, the directory and its contents will be exported allowing only read-only access.

If neither RO or RW is specified, then 'RW=' is assumed, and all host names have read-write access to the exported directory.

ANON=UID

If a request comes in from an unknown user, use this UID as the effective userid. Note that root users are considered unknown, unless specified on the ROOT option below. The default value for this option is the UID associated with the user profile *QNFSANON*.

If the user does not want to allow any requests from unknown users, use 'ANON=-1'.

NOSUID

Specifies that any attempt by the client to enable bits other than the permission bits will be ignored. If this option is not specified, attempt to set bits other than the permission bits will be carried out.

ROOT=[HOSTNAME[:HOSTNAME]](...)

Specifies the host name or host names for which root access is allowed to the exported directory tree. If root access is allowed for a host, an incoming UID of 0 is mapped to the user profile *QSECOFR*, and incoming requests from users with all object (*ALLOBJ) special authority are allowed. If root access is not allowed for a host, an incoming UID of 0 and incoming requests from users with *ALLOBJ special authority are mapped to the UID provided in the ANON option. If the ROOT option is not specified, no hosts will be granted root access.

ACCESS=[CLIENT[:CLIENT]](...)

Specifies the client or clients that are allowed to mount the exported directory tree. A client can be a host name or a netgroup. If no clients are specified, all clients will be allowed to mount the directory tree.

Directory (DIR)

Specifies the absolute path name of the existing directory to be exported (made available to NFS clients) or unexported (made unavailable to NFS clients). This directory can not be a subdirectory or a parent of an already exported directory (unless it is in a different file system). This parameter is not allowed when the -A flag is specified on the **NFS export options (OPTIONS)** parameter. This parameter is required when the -F flag is specified on the OPTIONS parameter.

Note: A value (other than *NONE) must be specified for either the OPTIONS or DIR parameter. Both OPTIONS and DIR can be specified so long as '-A' is not part of the options list specified for the OPTIONS parameter.

Host name (HOSTOPT)

The HOSTOPT parameter has four elements that specify additional information about the NFS clients that a directory tree is to be exported to. If the HOSTOPT parameter is not specified for a host name the user is exporting the directory tree to, the defaults for each of the elements of the HOSTOPT parameter are assumed for that host.

***DFT** *DFT specifies that the default values for the elements of the HOSTOPT parameter are used for all clients to which the directory tree or directory trees are to be exported. The network data file coded character set identifier (CCSID) is *BINARY, the network path name CCSID is *ASCII, and Force synchronous write is *SYNC.

Element 1: Host name

The name of the host for which additional options are to be specified. This host should be specified above in the OPTIONS -O list as a host that has access to the exported directory tree. Specify either a single host name that is an alias for an address of a single host or a netgroup name to be associated with these options.

The user can assign names to an internet address with the Work with TCP/IP host table entries option on the Configure TCP/IP menu (CFGTCP) command or via the OS/400 iSeries Navigator graphical user interface. Also, a remote name server can be used to map remote system names to internet addresses.

Element 2: Network data file coded character set identifier (CCSID)

The network data file CCSID is used for data of the files sent and received from the specified HOST NAME (or netgroup name). For any hosts not specified on a HOSTOPT parameter, the default network data file CCSID (*BINARY) is used. The CCSID may be one of the following:

***BINARY**

The default network 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 obtained from the default job CCSID is used.

1-65533

Specify a CCSID for data files.

Element 3: Network path name coded character set identifier (CCSID)

The network path name CCSID is used for the path name components of the files sent to and received from the specified HOST NAME (or netgroup name). For any hosts not specified on a HOSTOPT parameter, the default network path name CCSID (*ASCII) is used. The CCSID may be one of the following:

*ASCII

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

*JOBCCSID

The CCSID obtained from the default job CCSID is used.

1-65533

Specify a CCSID for path name components of files. 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.

Element 4: Write mode

Specifies whether write requests are handled synchronously or asynchronously for this HOST NAME (or netgroup name). The default value of *SYNC means that data will be written to disk immediately. *ASYNC does not guarantee that data is written to disk immediately, and can be used to improve server performance.

Note: The Network File System (NFS) protocol has traditionally used synchronous writes.

*SYNC

Write requests are performed synchronously.

*ASYNC

Write requests are performed asynchronously.

Top

Examples

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

Example 1: Exporting All Entries from /etc/exports

```
CHGNFSEXP  OPTIONS('-A')  
-or-  
CHGNFSEXP  '-A'
```

Both of these commands export all entries that exist in the */etc/exports* file.

Example 2: Exporting One Directory with Options

```
CHGNFSEXP  '-I -O R0,ANON=guest1,ACCESS=Roch1:9.7.431.2'  
           '/programs/public' HOSTOPT((MIAMI1 850 850))
```

This command exports the directory tree under the path name */programs/public* as read-only. It allows only two clients to mount this directory tree. It takes advantage of the positional parameters OPTIONS and DIR. It uses the HOSTOPT parameter to specify coded character set identifier (CCSID) for the host MIAMI1.

Example 3: Exporting One Directory with Options and Updating the /etc/exports File.

```
CHGNFSEXP '-I -F -O R0,ANON=guest1,ACCESS=Roch1:9.7.431.2'  
'/programs/public' HOSTOPT((MIAMI1 850 850))
```

This command exports the directory tree under the path name */programs/public* as read-only. It allows only two clients to mount this directory tree. The `OPTIONS` parameter value is specified positionally. It uses the `HOSTOPT` parameter to specify data and path name coded character set identifiers (CCSIDs) of 850 for host name MIAMI1.

In addition, it also adds an export entry for */programs/public*, along with the `OPTIONS` and `HOSTOPT` parameter values, to the */etc/exports* file.

[Top](#)

Error messages

None

[Top](#)

Extract Program Information (EXTPGMINF)

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

Parameters
Examples
Error messages

The Extract Program Information (EXTPGMINF) command extracts external linkage information from extended program model (EPM) program objects, and stores this information in a library information file. External linkage information, which includes external variables and entry points, can only be extracted from EPM program objects. The C/400*, FORTRAN/400*, and AS/400* Pascal compilers produce EPM program objects.

A library information file is a collection of the linkage information for a set of related programs. The library information file name is used on the LIBFILE parameter of the SETPGMINF command. For example, a library information file is provided for the set of programs that make up the C/400 run-time library.

The EXTPGMINF command lets you create a file to store the names of all the affected entry points in your application, instead of specifying each program object name on the SUBPGM parameter of the SETPGMINF command.

Error messages for EXTPGMINF

None

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Program	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *USRLIBL, *CURLIB</i>	
FILE	File to receive information	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: File to receive information	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OPTION	Extract record options	*REPLACE , *DELETE, *UPDATE	Optional, Positional 3
CRTFILE	Create the file	*NO , *YES	Optional
RECLIB	Library name to record	<i>Name, *LIBL, *FOUND</i>	Optional
CHECK	Consistency Check	*ALL , *ENTRY, *DATA, *NONE	Optional

Top

Program (PGM)

The PGM parameter specifies the name of the program and library that contains the linkage information you want to extract. This is a required parameter.

program-name

Enter the name of a program that contains the linkage information you want to extract.

*generic**

Enter a generic name for the programs that contain the linkage information you want to extract.

***ALL** Linkage information is extracted from all the programs that exist in the library.

The possible library values are:

***LIBL** The system searches the library list. You cannot specify *LIBL if you specify CRTFILE (*YES).

***USRLIBL**

The system searches the user portion of the library list.

***CURLIB**

The name of the current library is used. If you have not specified the current library, QGPL is used.

Top

File to receive information (FILE)

Specifies the name and library of the library information file. If the file does not exist, specify CRTFILE(*YES) to create it. If you do not, a message is issued.

file-name

Enter the name of the file where the linkage information will be stored.

The possible library values are:

***LIBL** The system searches the library list.

***CURLIB**

The name of the current library is used. If you have not specified a current library, QGPL is used.

library-name

Enter the name of the library where the linkage information file is located.

Top

Extract record options (OPTION)

Specifies the option of replacing, deleting, or updating data in the library information file.

***REPLACE**

Clears all data in the library information file, and replaces it with the information extracted from the program specified on the PGM parameter.

***DELETE**

Deletes the data in the library information file for the program specified on the PGM parameter. The deletion of data from this library information file results in a compressed file. Data that does not relate to the program specified on the PGM parameter remains in the library information file. No new information is added to the library information file.

***UPDATE**

Deletes the existing information for the specified programs, and replaces it with new information. The deletion of data from this library information file results in a compressed file. If the specified library information file is empty, this option is equivalent to *REPLACE.

Top

Create the file (CRTFILE)

Creates a library information file to store the extracted information.

***NO** Does not create a library information file.

***YES** Creates a library information file to store the extracted information. Select *YES when the specified library information file does not exist. If the file exists, a message is displayed.

Top

Library name to record (RECLIB)

Specifies the name of the library where the programs are stored. At run-time and when you enter the SETPGMINF command, the system searches for programs in the library you specify here.

***LIBL** The system searches the library list.

***FOUND**

During the processing of the EXTPGMINF command, the system records the name of the library where the specified programs are found. The system searches for the library that contained the programs at the time the EXTPGMINF command was processed.

library-name

Enter the name of the library.

Top

Consistency Check (CHECK)

Specifies that the data and entry points in your library information file are checked for consistency. If *NONE is specified, then no warning message will be given; otherwise a PSE warning message will be issued.

***ALL** Checks consistency of both data and entry points.

***ENTRY**

Checks consistency of entry points.

***DATA**

Checks consistency of data.

***NONE**

No consistency checking is performed.

Top

Examples

None

Top

Error messages

None

File Document (FILDOC)

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

Parameters
 Examples
 Error messages

The File Document (FILDOC) command allows a user to file a document in the document library.

Restrictions: (1) You can file a document on behalf of another user if you are authorized to work on behalf of the other user. You must be granted authority to work on behalf of another with the Grant User Permission (GRTUSRPMN) command. (2) The user ID and address must be enrolled in the system distribution directory. (3) Security for the new document is taken from the parameters in the FILDOC command and not inherited from the folder.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Information to be filed	*FILE, *IDP, *DSTID	Required, Positional 1
TODOC	To document	Character value, *NONE	Optional
TOFLR	To folder	Character value, *NONE	Optional
SENSITIV	Sensitivity	*NONE, *PERSONAL, *PRIVATE, *CONFIDENTIAL	Optional
USRAUT	User authority	Single values: *NONE Other values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: User profile	Name, *PUBLIC	
	Element 2: Authority level	*EXCLUDE, *USE, *CHANGE, *ALL, *AUTL	
AUTL	Authorization list	Name, *NONE	Optional
ACC	Access code	Single values: *NONE Other values (up to 50 repetitions): 0-2047	Optional
ALWRPL	Allow replacement	*NO, *YES	Optional
IDPFILE	Profile file	Single values: *NONE, *DOCFILE, *DSTIDIDP Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Profile file	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
IDPMBR	Profile member	Name, *FIRST	Optional
USRID	User identifier	Single values: *CURRENT Other values: <i>Element list</i>	Optional
	Element 1: User ID	Character value	
	Element 2: Address	Character value	
DOCFILE	Document file	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Document file	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
DOCMBR	Document member	Name, *FIRST	Optional
DSTID	Distribution identifier	Character value, *NONE	Optional
DSTIDEXN	Distribution ID extension	1-99, *NONE	Optional

Keyword	Description	Choices	Notes
KEEP	Keep in mail log	<u>*NO</u> , *YES, *REF	Optional
DOCTYPE	Document type	2-65535, <u>*DFT</u> , *FFT, *RFT	Optional
SYSCOD	Document system code	<i>Character value</i> , <u>*DFT</u>	Optional
DOCD	Document description	<i>Character value</i> , <u>*DFT</u>	Optional
AUTHOR	Author	Single values: <u>*NONE</u> , *USRID Other values (up to 50 repetitions): <i>Character value</i>	Optional
DOCCLS	Document class	<i>Character value</i> , <u>*NONE</u>	Optional
KWD	Keyword	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Character value</i>	Optional
SUBJECT	Subject	Single values: <u>*NONE</u> , *DOCD Other values (up to 50 repetitions): <i>Character value</i>	Optional
DOCDATE	Document date	<i>Date</i> , <u>*NONE</u> , *CURRENT	Optional
FILCAB	File cabinet location	<i>Character value</i> , <u>*NONE</u>	Optional
CPYLST	Copy list	Single values: <u>*NONE</u> Other values (up to 50 repetitions): <i>Character value</i>	Optional
EXPDATE	Expiration date	<i>Date</i> , <u>*NONE</u>	Optional
REFERENCE	Reference	<i>Character value</i> , <u>*NONE</u>	Optional
ACTDATE	Action due date	<i>Date</i> , <u>*NONE</u> , *CURRENT	Optional
STATUS	Document status	<i>Character value</i> , <u>*NONE</u>	Optional
CMPPATE	Completion date	<i>Date</i> , <u>*NONE</u> , *CURRENT	Optional
PROJECT	Project	<i>Character value</i> , <u>*NONE</u>	Optional
DOCCHRID	Document character identifier	Single values: <u>*SYSVAL</u> , *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	
DOCLANGID	Language ID	<i>Character value</i> , <u>*JOB</u>	Optional
DOCCNTRYID	Country or region ID	<i>Character value</i> , <u>*JOB</u>	Optional
PERSONAL	Personal	<u>*NO</u> , *YES	Optional
DSTEXPDATE	Distribution expiry indicator	<i>Element list</i>	Optional
	Element 1: Date	<i>Date</i> , <u>*NONE</u> , *CURRENT	
	Element 2: Time	<i>Time</i> , <u>*ENDOFDAY</u>	
CMDCHRID	Command character identifier	Single values: <u>*SYSVAL</u> , *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

Information to be filed (TYPE)

Specifies the type of information being filed and the parameters that are valid on this command.

This is a required parameter.

The possible values are:

***FILE** The database file specified on the **Document file** prompt (DOCFILE parameter) and the **Document member** prompt (DOCMBR parameter) parameter is filed.

Note: If this value is specified, you must specify the default values on the FILCAB, DSTID, DSTIDEXN, and KEEP parameters and you cannot specify DOCFILE(*NONE).

***IDP** The interchange document profile (IDP) specified on the **Profile file** prompt (IDPFILE parameter) and the **Profile member** prompt (IDPMBR parameter), or the document profile built by this command, is filed.

Note: If this value is specified, you must specify the default values on the DOCFILE, DOCMBR, DOCTYPE, SYSCODE, DOCCHRID, DSTID, DSTIDEXT, and KEEP parameters. If this value is specified, and IDPFILE and FILCAB cannot both specify *NONE.

***DSTID**

The distribution document identified by the distribution identifier specified in the **Distribution identifier** prompt (DSTID parameter) is filed from the mail log into the document library.

Note: If this value is specified, you cannot specify DSTID(*NONE).

Top

To document (TODOC)

Specifies the name of the newly filed document.

document-name

Specify the user-assigned name of the newly filed document. A maximum of 12 characters can be specified. This document name must not exist in the folder that the document is being filed into.

Top

To folder (TOFLR)

Specifies the name of the folder that contains the newly filed document. This parameter can be specified only when a value is also specified on the **To document** prompt (TODOC parameter).

The possible values are:

***NONE**

The newly filed document does not have a user-assigned name and is not filed in a folder.

folder-name

Specify the name of the folder to contain the newly filed document. A folder name can consist of a series of folder names (FLR1/FLR2/etc.) if the document is being filed in a folder that is contained in another folder. A maximum of 63 characters can be specified.

You must specify a folder name when a document name is specified on the **To document** prompt (TODOC parameter).

Top

Sensitivity (SENSITIV)

Specifies the level of sensitivity defined by the X.400 standard. The four levels include no sensitivity, personal, private and company confidential. Any document marked as private is still available to users who are normally authorized to it, but is unavailable to users who are working on your behalf (even though it may be available to them when they are not working on your behalf).

The possible values are:

***NONE**

The document has no sensitivity restrictions.

***PERSONAL**

The document is intended for the user as an individual.

***PRIVATE**

The document contains information that should be accessed only by the owner.

***CONFIDENTIAL**

The document contains information that should be handled according to company procedures.

Top

User authority (USRAUT)

Specifies name of an existing user and the user authority level. This parameter is used to change the authorized users for this document by giving more users authority to the document, removing a user's authority for the document, or changing the user's authority to the document.

The possible **user profile** values are:

***NONE**

No users have specific authority to access the document.

***PUBLIC**

Authority is given to the users who do not have specific authority to the document, who are not on the authorization list, and whose user's group does not have specific authority to the document.

user-profile-name

Specify the user profile names of one or more users being given authority to the document.

The possible **authority level** values are:

***ALL** The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

***CHANGE**

The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**

The user cannot access the object.

***AUTL**

The authority of the authorization list specified on the **Authorization list** prompt (AUTL parameter) is used for the document. The *AUTL value is valid only if *PUBLIC is also specified.

Top

Authorization list (AUTL)

Specifies the name of an authorization list used to secure the document specified on the **To document** prompt (TODOC parameter).

The possible values are:

***NONE**

An authorization list is not specified.

authorization-list-name

Specify the name of the authorization list whose authority is used for the document.

Top

Access code (ACC)

Specifies the access codes used with this document. The access codes must already exist on the system. If they do not already exist, they must be added on the system with the Add Access Code (ADDACC) command.

You can enter multiple values for this parameter.

The possible values are:

***NONE**

No access codes are assigned to this document. Authority for this document is controlled by the values specified on the **User authority** prompt (USRAUT parameter) and the **Authorization list** prompt (AUTL parameter).

access-code

Specify the access codes, ranging from 0 through 2047, that control who can use the document. A maximum of 50 access codes can be specified. Access code 0 gives *USE authority to all users.

Top

Allow replacement (ALWRPL)

Specifies the setting to allow replacement of the document content of the document being filed. If this parameter is specified when filing a document that cannot be replaced, it is ignored. A document that cannot be replaced cannot be changed back to a document that can be replaced.

The possible values are:

***NO**

The document content of the document being filed cannot be replaced.

***YES**

The document content of the document being filed can be replaced.

Top

Profile file (IDPFILE)

Specifies where the document profile information is located. If you specify this parameter, the remaining parameters after the **Profile member** prompt (IDPMBR parameter) are ignored, except the **Command character identifier** prompt (CMDCHRID parameter) and the **Document character identifier** prompt (DOCCHRID parameter).

The possible values are:

***NONE**

The interchange document profile (IDP) is supplied by other parameters on this command. There is no database file containing the IDP information. If *NONE is specified, the **Profile member** prompt (IDPMBR parameter) is ignored.

***DSTIDIDP**

The IDP information associated with the distribution document is used. The **Profile member** prompt (IDPMBR parameter) is ignored. This is valid only when TYPE (*DSTID) is specified.

***DOCFILE**

The database file specified for the document also contains the profile information. If *DOCFILE is specified, the **Document file** prompt (DOCFILE parameter) and **Document member** prompt (DOCMBR parameter) are used for the document profile information.

data-base-file-name

Specify the name of the database file that contains the IDP. The document profile database file can be a user-defined file or the output file specified on the Receive Distribution (RCVDST) or Retrieve Document (RTVDOC) commands. If you specify a user-defined file, it must have the same format as the output file produced by RCVDST or RTVDOC. If an output file is specified, only the data portion of the document profile record is read from the output file. The prefix is removed from the document profile record.

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 is specified as the library for the job, QGPL is used.

library-name

Specify the library where the database file is located.

Top

Profile member (IDPMBR)

Specifies the interchange document file member name being used. This parameter is used only when a database file name is also specified on the **Profile file** prompt (IDPFILE parameter).

The possible values are:

***FIRST**

The first member created in the database file is used.

member-name

Specify the name of the database file member being used.

Top

User identifier (USRID)

Specifies which user ID and user ID address should be associated with the request.

The possible values are:

***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 file (DOCFILE)

Specifies the names of the database file and the library that contains the document data. The database file is a user-defined file or the output file specified in either the Receive Distribution (RCVDST) command or the Retrieve Document (RTVDOC) command. If an output file is specified, only the data portion of the document data record is read from the output file. The prefix is removed from the document data record.

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 library is specified as the library for the job, QGPL is used.

library-name

Specify the library where the database file is located.

data-base-file-name

Specify the name of the database file that contains the document data.

Top

Document member (DOCMBR)

Specifies the document database file member that is used.

The possible values are:

***FIRST**

The first member created in the database file is used.

member-name

Specify the name of the database file member that is used.

Top

Distribution identifier (DSTID)

Specifies the unique distribution identifier of the distribution. The distribution identifier is assigned by the system when the distribution is originated. Distribution identifiers can be found by using the Query Distribution (QRYDST) command. Identifiers are also returned from the Send Distribution (SNDDST) command.

The possible values are:

***NONE**

No distribution identifier is used.

distribution-id

Specify the 3-part distribution identifier which 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 leading zeros. For example, 'NEWYORK SMITH 0204'. This parameter is required when *DSTID is specified on the **Information to be sent** prompt (TYPE parameter).

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.

The possible values are:

*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.

Top

Keep in mail log (KEEP)

Specifies whether to keep a copy of the distribution document filed in the mail log, delete the distribution from the mail log, or keep a reference in the mail log of the filed distribution document.

The possible values are:

*NO Delete the distribution document from the mail log after the file is complete.

*YES Keep a copy of the filed distribution document in the mail log.

*REF The distribution document is deleted and a reference to the filed distribution document is kept in the mail log.

Top

Document type (DOCTYPE)

Specifies the type of document being used. This identifier is used by the system to determine whether the data stream can be handled properly.

The possible values are:

*DFT The system creates the proper document type identifier based on the source of the data.

*FFT The document is in Final Form Text. This type of document is intended to be viewed and printed, but not edited, by the receiver.

*RFT The document is in Revisable Form Text. This type of document can be viewed, printed, and edited by the receiver.

document-type-number

Specify a document type identifier value ranging from 2 through 65,535. The numbers from 2

through 32,767 are controlled by registering them with the IBM Document Interchange Architecture and are used for IBM-defined document types. The numbers ranging from 32,768 through 65,535 are not registered with IBM and can be used for non-IBM-defined document types. The meaning of these document types must be determined by defining the value of the system code on the **System code** prompt (SYSCOD parameter).

Top

System code (SYSCOD)

Specifies the text used with the value specified on the **Document type** prompt (DOCTYPE parameter) to help uniquely identify the type of document being used. The receiver of the data stream determines the document data stream and processing requirements to edit, view, print, or change the document.

The possible values are:

***DFT** The system supplies a default system code. If the value specified on the **Document type** prompt (DOCTYPE parameter) is a number ranging from 2 through 32,767, the default is 'IBM AS/400 CL' and is retrieved from message CPX9026. If the value specified on the **Document type** prompt (DOCTYPE parameter) is in the range from 32,768 through 65,535, a system code must be specified.

system-code

Specify the text that uniquely identifies the type of document being sent. A maximum of 13 characters can be specified.

Top

Document description (DOCD)

Specifies a description for the document being filed. This description is in the Document Interchange Architecture document name field.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

The possible values are:

***DFT** The system creates a document description from the database files. The default is (library-name/file-name/member-name) for database files. If *IDP is specified on the **Information to be filed** prompt (TYPE parameter) to file only a reference to a printed document, the default document name is Hardcopy Document Reference and is retrieved from the message CPX9025. An installation may change this message, but only the first 44 characters are used in the document name. If *DSTID is specified on the **Information to be filed** prompt (TYPE parameter), the default document name will be the distribution document name specified when the distribution was sent.

document-description

Specify the description of the document. A maximum of 44 characters can be specified.

Top

Author (AUTHOR)

Specifies the author or authors of the document.

You can enter multiple values for this parameter.

The possible values are:

*NONE

No author is identified for the document.

*USRID

The user ID and address specified on the USRID parameter **User identifier** prompt(USRID parameter) is used as the author's name.

document-author-name

Specify the name of the author or authors. A maximum of 50 authors can be specified.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Document class (DOCCLS)

Specifies the class associated with this document, such as MEMO, FORM, or SHEET.

The possible values are:

*NONE

No class is assigned to the document.

document-class

Specify the document class. A maximum of 16 characters can be specified.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Keyword (KWD)

Specifies the keywords that describe the document.

You can enter multiple values for this parameter.

The possible values are:

*NONE

No keywords are defined for this document.

document-keyword

Specify the keywords to describe the document. A maximum of 50 keywords can be specified. Each keyword can have a maximum of 60 characters.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Subject (SUBJECT)

Specifies the subject or subjects of the document.

You can enter multiple values for this parameter.

The possible values are:

***NONE**

No subject is defined for the document.

***DOCD**

The document description is used as the subject for the document.

document-subject

Specify the subject or subjects of the document. A maximum of 50 subjects can be specified and each subject can have a maximum of 60 characters of text.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Document date (DOCDATE)

Specifies any date the user needs to assign to the document.

The possible values are:

***NONE**

No date is assigned to the document.

***CURRENT**

The system assigns the current system date to the document.

document-date

Specify the document date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

File cabinet location (FILCAB)

Specifies the location of the document. This parameter is intended to describe the location of printed documents. The interchange document profile (IDP) that refers to the printed document is distributed. This parameter is required if *IDP is also specified on the **Information to be sent** prompt (TYPE parameter) and *NONE is specified on the **Profile file** prompt (IDPFILE parameter).

The possible values are:

***NONE**

No filing cabinet reference is defined for this document.

filing-cabinet-reference

Specify the text that describes where the printed document is located. A maximum of 60 characters can be specified.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Copy list (CPYLST)

Specifies the names of the users that receive this document.

You can enter multiple values for this parameter.

The possible values are:

*NONE

No copy list is included for this document.

recipient-list

Specify the names of the users that receive the document. A maximum of 50 names can be specified. Each name can have a maximum of 60 characters.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Expiration date (EXPDATE)

Specifies the date on which the document is no longer needed.

The possible values are:

*NONE

No document expiration date is specified.

expiration-date

Specify the document expiration date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Reference (REFERENCE)

Specifies a reference associated with the document.

The possible values are:

*NONE

No reference field is included for this document distribution.

reference

Specify text that describes the reference associated with the document. A maximum of 60 characters can be used.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Action due date (ACTDATE)

Specifies the date when the action requested is due.

The possible values are:

***NONE**

No action due date request is specified.

***CURRENT**

The current system action due date is used.

action-due-date

Specify the action due date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Document status (STATUS)

Specifies the user-defined status of the document. Examples of status are: In Process, Pending Approval, or Retired.

The possible values are:

***NONE**

No status is included in this document.

status-of-document

Specify text that describes the status of the document. A maximum of 20 characters can be specified.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Completion date (CMPDATE)

Specifies the date when the requested action is completed.

The possible values are:

***NONE**

No completion date is included.

***CURRENT**

The current system date is used as the completion date.

date-complete

Specify the action completion date. The date must be specified in the job date format.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Project (PROJECT)

Specifies the project associated with the document.

The possible values are:

*NONE

No project field information is included in this document.

project

Specify text that describes the project of the document. A maximum of 10 characters can be specified.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Document character identifier (DOCCHRID)

Specifies the character identifier (graphic character set and code page) for the document data being used. The character identifier is related to the display device used to create the document data.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

*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.

graphic-character-set code-page

Specify the graphic character set and code page values used to create the data being distributed.

Note: Both parts can be up to 5 digits in length.

Top

Language ID (DOCLANGID)

Specifies the language identifier to be placed in this document's interchange document profile (IDP).

The possible values are:

*JOB The language identifier specified for the job in which this command is entered is used.

language-identifier

Specify a language identifier. Press the PF4 key from the **Language ID** prompt (DOCLANGID parameter) to see a list of valid identifiers.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Country or region ID (DOCCNTRYID)

Specifies the country or region identifier to be placed in this document's interchange document profile (IDP).

The possible values are:

***JOB** The country or region identifier specified for the job in which this command is entered is used.

country-or-region-ID

Specify a country or region identifier. Press the PF4 key from the **Country or region ID** prompt (DOCCNTRYID parameter) to see a list of valid identifiers.

Note: This parameter is ignored if the **Profile file** prompt (IDPFILE parameter) is specified.

Top

Personal (PERSONAL)

Specifies whether the document distribution is private or not. This parameter is replaced by SENSITIV but the PERSONAL parameter can still be used. However, because this parameter may be removed in a later release, whenever possible use the SENSITIV parameter.

If PERSONAL(*YES) is used, the SENSITIV parameter must be omitted or it must be with the value SENSITIV(*NONE). If the command is prompted without this parameter specified, this parameter is not displayed.

The possible values are:

***NO** Only the owner and users that have authorization to the document can get access to documents that are not sensitive. Users authorized to work on behalf of other users who have access to the document can access documents that are not sensitive. This value will map to SENSITIV(*NONE).

***YES** Only the owner can get access to private documents. Users authorized to work on behalf of other users who have access to the document cannot get access to the document. This value will map to SENSITIV(*PRIVATE).

Top

Distribution expiry indicator (DSTEXPDATE)

Specifies the date and time on which the distribution is no longer needed in the mail log.

The possible dist-expiration-date values are:

***NONE**

No expiration date,

***CURRENT**

The current date is used.

'expiration-date'

Specify the value to use as the expiration date. The date must be specified in the format specified by the system value QDATFMT.

The possible dist-expiration-time values are:

***ENDOFDAY**

An expiration time is requested by the end of the specified date. The time is set to 23:59:59.

'expiration-time'

Specify the value used as the expiration time.

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 character set and code page are stored with the fields to allow the display station that accesses the document to correctly print or display the fields. The fields are translated to a common character set and code page when the fields are written to the search database. The interchangeable character set and code page are '697 500'.

The following fields are translated:

- **User identifier** (USRID)
- **Distribution identifier** (DSTID)
- **Document system code** (SYSCOD)
- **Document description** (DOCD)
- **Author** (AUTHOR)
- **Document class** (DOCCLS)
- **Keyword** (KWD)
- **Subject** (SUBJECT)
- **File cabinet location** (FILCAB)
- **Copy list** (CPYLST)
- **Reference** (REFERENCE)
- **Document status** (STATUS)
- **Project** (PROJECT)

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

Example 1: Filing a Personal Document

```
FILDOC TYPE(*FILE) DOCFILE(MARYLIB/MARYFILE)
      SENSITIV(*PRIVATE) IDPFILE(*DOCFILE)
```

This command files a private document using a database file that has the document content and document profile information. The default for the distribution ID extension is 01 (DSTID(01)).

Example 2: Filing a Distribution Document

```
FILDOC TYPE(*DSTID) DSTID('NEWYORK SMITH 0201') DSTID(02)
      DOCCLS('NEW CLASS') TODOC(DST0201) TOFLR(FLRDST)
```

This command files a distribution document in the document library QDOC in document DST0201 and folder FLRDST. The document class is changed in the distribution document, and the second distribution that was sent to the user is filed.

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.

CPF901B

Document filing request failed.

CPF902B

Authority of *AUTL is allowed only with USRAUT(*PUBLIC).

CPF905C

Error occurred trying to find a translation table.

CPF9096

Cannot use CMDCHRID(*DEVVD), DOCCHRID(*DEVVD) in batch job.

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

Format Data (FMTDTA)

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

Parameters
Examples
Error messages

The Format Data (FMTDTA) command processes a series of Sort specifications stored in a source file member.

Top

Parameters

Keyword	Description	Choices	Notes
INFILE	Input file	Values (up to 8 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, *CURLIB, *LIBL</i>	
	Element 2: Member	<i>Name, *FIRST</i>	
OUTFILE	Output file	<i>Element list</i>	Required, Positional 2
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *CURLIB, *LIBL</i>	
	Element 2: Member	<i>Name, *FIRST</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Source file	<i>Name, QFMTSRC</i>	
	Qualifier 2: Library	<i>Name, *CURLIB, *LIBL</i>	
SRCMBR	Source member	<i>Name, *FIRST</i>	Optional, Positional 4
PRTFILE	Print file	<i>Qualified object name</i>	Optional, Positional 5
	Qualifier 1: Print file	<i>Name, QSYSPRT</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OPTION	Options:	Values (up to 8 repetitions): *CHK, *NOCHK, *PRT, *NOPRT, *DUMP, *NODUMP, *NOSECLVL, *SECLVL	Optional, Positional 6
PGMDATE	Program date:	<i>Date, *CURRENT</i>	Optional

Top

Input file (INFILE)

Specifies up to eight names for files that are to be used as input. For database files, one member name can be specified for each file name. For diskette files, the diskette identifier can be specified for each device file name. This is a required parameter.

file-name

Enter the name of the file that is to be used as input.

*CURLIB

The current library will be used. If you have not specified a current library, QGPL will be used.

***LIBL** The system searches the library list to find the library where the file is located.

library-name

Enter the name of the library of the input file.

FIRST

The first member in the file is to be used as input.

data-file-identifier

For diskette files, enter one data file identifier per device file name specified. If more than one diskette data file is to be processed for a device file name, the device file name should be specified as many times as required.

member-name

For database files, enter one member name per database file name specified. If more than one member of the same database file is to be processed, the database file name should be specified as many times as required.

Top

Output file (OUTFILE)

Specifies the name of the file and the name of the member to be used for output. Both the file and member must exist before being named in this parameter. This is a required parameter.

file-name

Enter the name of the output file to be used.

CURLIB

The current library will be used. If you have not specified a current library, QGPL will be used.

***LIBL** The system searches the library list to find the library where the file is located.

library-name

Enter the name of the library of the output file.

FIRST

The first member in the file is to be used for output.

member-name

Enter the name of the member in the output file that is to be used for output.

Top

Source file (SRCFILE)

Specifies the name of the source file containing the sort specifications to be run. The source file may be a device or database file, and it must have the attributes of a source file.

QFMTSRC

The IBM-supplied source file QFMTSRC contains the sort specifications.

source-file-name

Enter the name of the source file that contains the sort specifications.

***LIBL** The system searches the library list to find the library where the source file is located.

***CURLIB**

The current library will be used. If you have not specified a current library, QGPL will be used.

library-name

Enter the name of the library that contains the source file.

Source member (SRCMBR)

Specifies the name of the source file member containing the sort specifications to be run. The source file may be a device or database file, and it must have the attributes of a source file.

*FIRST

The first member of the source file containing the sort specifications is to be run.

data-file-identifier

Enter the name of the diskette data file identifier that contains the sort specification statements, if the data file resides on diskette.

member-name

Enter the name of the member of the source file containing the sort specifications to be run.

Top

Print file (PRTFILE)

Specifies the name of the printer device file to which the print data is to be sent.

QSYSPRT

The data is to be printed by the system printer.

print-file-name

Enter the name of the printer device file that is to print the data.

*LIBL The system searches the library list to find the library where the file is located.

*CURLIB

The current library will be used. If you have not specified a current library, QGPL will be used.

library-name

Enter the name of the library that contains the file.

Top

Options: (OPTION)

Specifies the sequence checking and printing options to be used while the sort utility is running.

*CHK The sort specifications are to be sequence-checked.

*NOCHK

The sort specifications are not to be sequence-checked.

*PRT The sort specifications and any error or informational messages are to be printed.

*NOPRT

The sort specifications and any error or informational messages are not to be printed.

*NODUMP

The internal tables used for problem analysis are not to be printed.

*DUMP

The internal tables used for problem analysis are to be printed.

*NOSECLVL

Suppresses the printing of second level text for errors detected during compilation.

***SECLVL**

Prints second level text for errors detected during compilation.

[Top](#)

Program date: (PGMDATE)

Specifies the date that can be used with factor 2 as a keyword in record specifications.

***CURRENT**

Use the current system date when the command is processed.

***DATE**

Enter the date in the format specified by system value QDHTFMT, or if separators are used, by QDATSEP.

[Top](#)

Examples

None

[Top](#)

Error messages

Unknown

[Top](#)

Generate Message Catalog (GENCAT)

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

Parameters
Examples
Error messages

This command is an alias for the Merge Message Catalog (MRGMSGCLG) command and can also be issued using the following alternative command name:

- MRGMSGCLG

The Generate Message Catalog (GENCAT) 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.

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
TEXT	Text 'description'	<i>Character value</i> , *BLANK	Optional
CLGCCSID	Message catalog CCSID	1-65533, *SRCCCSID, *JOB	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.

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

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

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

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

Examples for MRGMSGCLG

```
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

Generate Command Documentation (GENCMDDOC)

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

Parameters
Examples
Error messages

The Generate Command Documentation (GENCMDDOC) command generates an output file which contains documentation for a Control Language (CL) command. The generated file will be one of the following:

- If *HTML is specified for the **Generation options (GENOPT)** parameter, the file will contain HyperText Markup Language (HTML) source. The generated file can be displayed using a standard internet browser, and conforms to HTML 4.0 specifications. The information used to generate the file is retrieved from the specified command (*CMD) object and any command help panel group (*PNLGRP) objects associated with the command.
- If *UIM is specified for the GENOPT parameter, the file will contain User Interface Manager (UIM) source. The generated source is an outline for the online command help for the command. The information used to generate the file is retrieved only from the specified command (*CMD) object. This option is intended to simplify the task of writing online help for CL commands.

See the CL Programming book, SC41-5721, for more information on writing command documentation using UIM.

Restrictions:

- You must have use (*USE) authority to the specified command and execute (*EXECUTE) authority for the library that the command is in. If a generic name or *ALL is specified for the command name, no output file is generated for any commands that you do not have *USE authority for.
- For each associated panel group that contains command help information for the specified command, you must have *USE authority to the panel group and *EXECUTE authority for the library that the panel group is in.
- You must have execute (*X) authority to the directories in the path for the generated file, and write and execute (*WX) authorities to the parent directory of the generated file.
- If the output file already exists, you must have write (*W) authority to the file and *YES must be specified for the **Replace file (REPLACE)** parameter.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Command	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
TODIR	To directory	<i>Path name, .</i>	Optional
TOSTMF	To stream file	<i>Character value, *CMD</i>	Optional
REPLACE	Replace file	<i>*YES, *NO</i>	Optional
GENOPT	Generation options	Values (up to 3 repetitions): *HTML, *UIM, *NOSHOWCHOICEPGMVAL, *SHOWCHOICEPGMVAL, *NOSERVICE, *SERVICE	Optional

Top

Command (CMD)

Specifies the command for which a documentation output file is to be generated.

Note: If a generic command name or *ALL is specified for the command name, *LIBL is not allowed as the library name qualifier and the value for the **To stream file (TOSTMF)** parameter must be *CMD.

This is a required parameter.

Qualifier 1: Command

***ALL** Documentation files for all of the commands in the specified library are to be generated.

generic-name

Specify the generic name of the commands for which documentation files are to be generated. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all commands that have names with the same prefix as the generic command name will have documentation files generated.

name Specify the name of the command for which you want to generate a documentation output 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 command. If no library is specified as the current library for the job, QGPL is used.

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

Top

To directory (TODIR)

Specifies the directory in which the generated command documentation file will be stored. The file name to be used within this directory is specified by the **To stream file (TOSTMF)** parameter.

./ The output file will be stored in the current working directory.

path-name

Specify the path name for the directory where you want the generated output file stored.

Top

To stream file (TOSTMF)

Specifies the target stream file to be used to store the generated command documentation file. The specified file will be located using the directory path specified for the **To directory (TODIR)** parameter.

Note: If a generic command name or *ALL is specified for the **Command (CMD)** parameter, the value specified or defaulted for this parameter must be *CMD.

***CMD** If the TODIR parameter specifies that the target is in the /QSYS.LIB file system, the generated file name will be same as the command name.

Otherwise, the generated file name depends on whether *HTML or *UIM is specified for the **Generation options (GENOPT)** parameter. If *HTML is specified, the generated file name will be

libname_cmdname.html, where **cmdname** is the command name and **libname** is the name of the library where the command is located. If *UIM is specified, the generated file name will be **libname_cmdname.uim**

character-value

Specify the name to be used for the generated command documentation file.

Top

Replace file (REPLACE)

Specifies whether or not to replace an existing file in the target directory (TODIR parameter) by the file name specified or generated (TOSTMF parameter).

***YES** If a file already exists by the name specified or implied, the file contents will be replaced with the generated command documentation file.

***NO** If a file already exists by the name specified or implied, an error message is sent and no command documentation file is generated. If no file by the same name exists in the target directory, the file will be created and no error message sent.

Top

Generation options (GENOPT)

Specifies options to control the command information to be generated. Multiple option values can be specified in any order on this parameter. If neither or both of the values in each group are specified, the underlined value will be used.

Note: The underlined values for this parameter are *similar* to, but not *actually* default values, and therefore, cannot be changed with the Change Command Default (CHGCMDDFT) command.

Generated Source Option

***HTML**

The generated file will contain HyperText Markup Language (HTML) source. The generated file can be displayed using a standard internet browser, and conforms to HTML 4.0 specifications. The information used to generate the file is retrieved from the specified command (*CMD) object and any command help panel group (*PNLGRP) objects associated with the command.

***UIM** The generated file will contain User Interface Manager (UIM) source. The generated source is an outline for the online command help for the command. The information used to generate the file is retrieved only from the specified command (*CMD) object. This option is intended to simplify the task of writing online help for CL commands. After editing the generated file to add descriptive text for the command and storing the source in a source file member, this UIM source can be used as input to the Create Panel Group (CRTPNLGRP) command to create a command help panel group for the command.

See the CL Programming book, SC41-5721, for more information on writing command documentation using UIM.

Choice Program Values Option

***NOSHOWCHOICEPGMVAL**

For command parameters that have an associated *choices program*, do not show the values which would be returned from the choices program in the generated parameter summary table. Choices program values can vary from system to system. Not showing the choices program values gives you just the parameter values defined in the command object.

***SHOWCHOICEPGMVAL**

For command parameters that have an associated *choices program*, show the values returned from calling the choices program in the generated parameter summary table. Showing the choice program values gives you the same parameter values that you would see if prompting the command on this system.

Service Option

***NOSERVICE**

No extra trace or dump information is generated.

***SERVICE**

This option is intended to be used if the command is not working and you are told by your software service provider to write an APAR for the problem. Specifying this option will cause additional trace and dump information to be generated. Send this additional generated information with the APAR.

Top

Examples

Example 1: Generating HTML Documentation for an OS/400 Command

```
GENCMDDOC  CMD(CRTUSRPRF)
```

This command generates a documentation file for the CRTUSRPRF command. The command will be located using the library list for the current thread. The generated stream file will be stored in the current working directory of the job. Assuming the command is found in library QSYS, the generated file name will be **QSYS_CRTUSRPRF.html**. If a file by that name already exists in the target directory, it will be replaced by the generated file.

Example 2: Generating UIM Documentation for User Command

```
GENCMDDOC  CMD(MYLIB/MYCMD)
            TODIR(' /QSYS.LIB/MYLIB.LIB/QPNLSRC.FILE')
            TOSTMF('MYCMD.MBR') REPLACE(*NO) GENOPT(*UIM)
```

This command generates a documentation file for command MYCMD which is located in library MYLIB. The generated file will be stored in file QPNLSRC in library MYLIB with a member name of MYCMD. If a member already exists in the target file with this name, an error message will be sent and no documentation file will be generated.

Top

Error messages

***ESCAPE Messages**

CPF6E74

&1 command documents failed; &2 command documents created successfully.

CPF6E75

Error detected on the CMD parameter.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9899

Error occurred during processing of command.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A0

Object already exists. Object is &1.

CPFA0A9

Object not found. Object is &1.

[Top](#)

Go to Menu (GO)

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

Parameters
Examples
Error messages

The Go to Menu (GO) command shows the menu requested. This command allows you to specify either a particular menu or a generic menu name. You can optionally specify whether or not to return to the menu from which the command is entered after showing the menu specified.

Using the Previous and Exit Keys

A menu is placed on an internal menu stack before it is run. If a stack is not available for the menu, one is created. When the Cancel key is pressed for a menu, the previous menu in the stack is shown. Each menu stack is ten elements (menus) deep. When the eleventh menu is placed on the menu stack, the first, or oldest, menu is removed from the stack. This menu cannot be returned to by using the Cancel key.

Pressing the Exit key returns the user to the last display or menu from which a GO command was entered with RTNPNT(*YES). The display that the user is returned to is found by removing menus from the menu stack until a return point is found. This process may also cause a program in the call stack to return to its calling program unless the program is a return point.

Pressing either the Exit or Cancel key while viewing help for a menu returns the user to the menu.

Restrictions:

- You must have use (*USE) authority for the menu and its display and message files or program (whichever applies).
- You must also have *USE authority for the library where the menu is located.

Top

Parameters

Keyword	Description	Choices	Notes
MENU	Menu	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Menu	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL, *ALL, *ALLUSR</i>	
RTNPNT	Return point	<i>*YES, *NO</i>	Optional, Positional 2

Top

Menu (MENU)

Specifies the menu to be shown.

This is a required parameter.

Qualifier 1: Menu

***ALL** A list of all menus in the specified library is shown from which you select the menu to be run.

generic-name

Specify the generic name of the menu to be run. A generic name is a character string that contains one or more characters followed by an asterisk (*). A list of all menus that have names that begin with the same characters as the generic menu name is shown from which you select the menu to be run.

name Specify the name of the menu to be shown.

Qualifier 2: Library

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

***CURLIB**

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

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched.

***ALLUSR**

All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

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

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered user libraries and are also searched:

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

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALL** All libraries in the system, including QSYS, are searched.

name Specify the name of the library to be searched.

Top

Return point (RTNPNT)

Specifies whether to return to the display where the command is entered when the Exit key is pressed.

***YES** The display where the command is entered is returned to when the Exit key is pressed.

***NO** The display where the command is entered is not returned to when the Exit key is pressed.

Examples

GO MENU(PERSMENU)

This command runs a menu called PERSMENU, located in a library found by searching the library list (*LIBL default value).

If the Exit key is pressed while PERSMENU is being shown, the display where the GO command was entered (*YES default value on the RTNPNT parameter) is shown.

Top

Error messages

*ESCAPE Messages

CPF6ACD

Menu &1 in &2 is wrong version for system.

CPF6AC7

Menu &1 in library &2 not displayed.

Top

Go To (GOTO)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The Go To (GOTO) command is used in CL procedures for branching from one part of the program to another. The branching is to the label on another command that is specified on the Go To (GOTO) command. Branching can be either forward or backward, but the specified label must be inside the program.

Restrictions: This command is valid only within a CL procedure.

Top

Parameters

Keyword	Description	Choices	Notes
CMDLBL	Command label	<i>Simple name</i>	Required, Positional 1

Top

Command label (CMDLBL)

Specifies the label name of the command to which control is transferred; when the Go To (GOTO) command is processed. The command with the label is then processed. If the specified command cannot be run (for example, if it is a DCL command), control is transferred to the next command following the command with the specified label. The label must be within the same program as the GOTO command. A CL variable name cannot be used to specify the label name.

This is a required parameter.

Top

Examples

```
LOOP: CHGVAR &A (&A + 1)
IF (&A *LT 30) THEN(GOTO LOOP)
```

The Change Variable (CHGVAR) command increases the value of &A by 1 until &A is equal to or greater than 30. The GOTO command is processed each time that the IF command tests the expression and the result is true; the GOTO command following the THEN parameter causes the procedure to branch back to the label LOOP on the CHGVAR command. Refer to the descriptions of the CHGVAR command and the IF command for additional explanations of their functions.

Top

Error messages

None

[Top](#)

Grant Access Code Authority (GRTACCAUT)

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

Parameters
Examples
Error messages

The Grant Access Code Authority (GRTACCAUT) command allows you to give the specified users authority to access documents and folders associated with the access codes. Access is restricted to read only (*USE authority).

Restrictions:

1. The access code must be defined to the system before you can grant access code authority.
2. The user being granted access code authority must be enrolled in the system distribution directory.
3. To use this command, you must have *ALLOBJ authority.

Top

Parameters

Keyword	Description	Choices	Notes
ACC	Document access code	Single values: *REFUSER Other values (up to 300 repetitions): 1-2047	Required, Positional 1
USER	User profile	Values (up to 300 repetitions): <i>Name</i>	Required, Positional 2
REFUSER	Reference user profile	<i>Name</i> , *NONE	Optional

Top

Document access code (ACC)

Specifies the access code being authorized for use by the user identified on the **User profile** prompt (USER parameter).

You can enter multiple values for this parameter.

The possible values are:

*REFUSER

The access code authority being granted is based on a second (referred to) user profile name; that name must be entered on the **Reference user profile** prompt (REFUSER parameter).

access-code

Specify a number, ranging from 1 through 2047 that identifies the access code to which you want authority to be granted. The access code must be defined to the system using the Add Access Code (ADDACC) command before being specified on this parameter.

Top

User profile (USER)

Specifies the user profile name of the users to whom you are granting access code authority. The users identified will have the access code added to their current list of authorized access codes; this access code is used to verify additional document and folder accesses from the document library. The user must be enrolled in the system distribution directory before being granted authority to use an access code.

Note: By granting an access code to a group user profile, that access code is granted (implicitly) to every user under that group.

You can enter multiple values for this parameter.

Top

Reference user profile (REFUSER)

Specifies the referred to user profile on which the access code authority is based. If this parameter is used, then *REFUSER must be entered on the **Document access code** prompt (ACC parameter).

The possible values are:

*NONE

No referred to user is used to grant access code authority.

based-on-user-profile

Specify the user-profile name that the access code authority is based on. This user must also be enrolled in the system distribution directory.

Top

Examples

Example 1: Granting Authority to Multiple Users

```
GRTACCAUT ACC(3 30 60) USER(SAM LARRY)
```

This command gives authority to access codes 3, 30, and 60 to SAM and LARRY.

Example 2: Granting Authority Based on Another User

```
GRTACCAUT ACC(*REFUSER) USER(JOE) REFUSER(TOM)
```

This command grants access code authority to JOE based on TOM's authority. For example, if JOE currently has authority to access codes 1, 12, and 50, and TOM currently has authority to access codes 8 and 9, the GRTACCAUT command authorizes JOE to access codes 1, 8, 9, 12, and 50.

Top

Error messages

*ESCAPE Messages

CPF9009

System requires file &1 in &2 be journaled.

CPF9013

Access code authority given to &1 users, not granted to &2 users.

CPF9024

System cannot get correct record to finish operation.

CPF9065

Not allowed to give access code authority.

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

Grant Object Authority (GRTOBJAUT)

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

Parameters
Examples
Error messages

The Grant Object Authority (GRTOBJAUT) command grants specific authority for the objects named in the command to another user or group of users.

Authority can be given to:

- Named users
- Users (*PUBLIC) who do not have specific authority to the object or the authorization list
- Users of the object referred to by the **Reference object (REFOBJ)** and **Reference object type (REFOBJTYPE)** and parameters
- Authorization lists

If AUT(*AUTL) is specified, the PUBLIC authority for the object comes from the PUBLIC authority of the authorization list securing the object.

The AUTL parameter is used to secure an object with an authorization list or remove an authorization list from an object. User profiles cannot be secured by an authorization list (*AUTL).

This command can be used by an object's owner, or by a user with object management authority for the specified object. A user with object management authority can grant to other users any authority that the user has, except object management authority. Only the owner of the object, or someone with all object special authority (*ALLOBJ), can grant object management authority to a user.

A user with *ALL authority can assign a new authorization list.

When granting authority to users, the REPLACE parameter indicates whether the authorities you specify replace the user's existing authorities. The default value of REPLACE(*NO) gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority. REPLACE(*YES) removes the user's current authorities, then grants the authority that you specify.

When granting authority with a reference object, this command gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority.

This command gives the authority that you specify, but it does not remove any authority that is greater than you specified, unless you are granting *EXCLUDE authority or specify REPLACE(*YES).

Restrictions:

1. This command must get an exclusive lock on a database file before read or object operational authority can be given to a user.
2. If a user requests authority for another specified user to a device currently in use by another authorized user, authority to the device is not given.
3. Object type *AUTL cannot be specified.
4. AUT(*AUTL) is valid only with USER(*PUBLIC).
5. A user must either be the owner of the object or have *ALL authority to use the AUTL parameter.
6. The user must have object management authority to the object.

7. If the object is a file, the user must have object operational and object management authorities.
8. For display stations or for work station message queues associated with the display station, if this command is not entered at the device for which authorities are to be granted, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
9. You must have *USE authority to the auxiliary storage pool device if one is specified.

Note: Caution should be used when changing the public authority on IBM-supplied objects. For example, changing the public authority on the QSYSOPR message queue to be more restrictive than *CHANGE will cause some system programs to fail. The system programs will not have enough authority to send messages to the QSYSOPR message queue. For more information, refer to the iSeries Security Reference, SC41-5302 book.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *ALL, *ALLUSR, *USRLIBL, *ALLAVL, *ALLUSRAVL</i>	
OBJTYPE	Object type	<i>*ALL, *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNNTBL, *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, *NWSO, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDL0D, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *S36, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST</i>	Required, Positional 2
ASPDEV	ASP device	<i>Name, *, *SYSBAS</i>	Optional
USER	Users	Single values: *PUBLIC Other values (up to 50 repetitions): <i>Name</i>	Optional, Positional 3
AUT	Authority	Single values: <i>*CHANGE, *ALL, *USE, *EXCLUDE, *AUTL</i> Other values (up to 10 repetitions): <i>*OBJALTER, *OBJEXIST, *OBJMGT, *OBJOPR, *OBJREF, *ADD, *DLT, *READ, *UPD, *EXECUTE</i>	Optional, Positional 4
AUTL	Authorization list	<i>Name, *NONE</i>	Optional
REFOBJ	Reference object	<i>Qualified object name</i>	Optional
	Qualifier 1: Reference object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
REFOBJTYPE	Reference object type	* OBJTYPE , *ALRTBL, *BNDDIR, *AUTL, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNNTBL, *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, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *S36, *SBS, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
REFASPDEV	Reference ASP device	Name, *, *SYSBAS	Optional
REPLACE	Replace authority	* NO , *YES	Optional

Top

Object (OBJ)

Specifies the objects for which specific authority is to be given to one or more users.

This is a required parameter.

generic-name

Specify the generic name of the objects for which specific authority is to be given to one or more users. A generic name is a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, all objects that have names with the same prefix as the generic name are shown.

name Specify the name of the object for which specific authority is to be given to one or more users.

***ALL** Specific authority is to be given to all objects of the specified object type (OBJTYPE parameter). A specific library name must be specified for the library qualifier when *ALL is specified.

Top

Object type (OBJTYPE)

Specifies the object type of the object for which specific authorities are to be given to the specified users or to an authorization list. Any of the object types can be specified except *AUTL. 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).

This is a required parameter.

***ALL** Specific authorities for all object types (except *AUTL) are given to the specified users or to the authorization list.

object-type

Specify the object type of the object for which specific authorities are to be given to the specified users.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where the library that contains the object (OBJ parameter) is located. If the object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is used as the target of this command's operation.

- * The ASPs that are currently part of the job's library name space will be searched to locate the object. This includes the system ASP (ASP number 1), all defined basic user ASPs (ASP numbers 2-32), and, if the job has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs will be searched to locate the object. No independent ASPs will be searched, even if the job has an ASP group.

name Specify the device name of the independent ASP to be searched to locate the object. The independent ASP must have been activated (by varying on the ASP device) and have a status of AVAILABLE. The system ASP and basic user ASPs will not be searched.

Top

Users (USER)

Specifies one or more users to whom authority for the named object is to be given.

This is a required parameter unless either the **Reference object (REFOBJ)** parameter or **Authorization list (AUTL)** parameter is specified.

*PUBLIC

Users are authorized to use the object as specified in the AUT parameter when they do not have authority specifically given to them for the object, are not on the authorization list and none of their groups have any authority or are on the authorization list. Users who do not have any authority, and whose groups do not have any authority, are authorized to use the object as specified in the AUT parameter.

name Specify the names of one or more users to be given specific authority for the object. Up to 50 user profile names can be specified.

Top

Authority (AUT)

Specifies the authority to be given to the users specified for the **Users (USER)** parameter.

If a value is specified for this parameter, you cannot specify a value for the AUTL, REFOBJ, or REFOBJTYPE parameters.

*CHANGE

The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

*ALL The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**

The user cannot access the workstation object.

***AUTL**

The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

Note: You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

A maximum of ten of the following values can be specified.

***OBJALTER**

Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority 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. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJMGT**

Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

***OBJEXIST**

Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

***OBJOPR**

Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

***OBJREF**

Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

The possible data authorities are:

***ADD** Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

***DLT** Delete authority provides the authority to remove entries from an object.

***EXECUTE**

Execute authority provides the authority needed to run a program or locate an object in a library.

***READ**

Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

***UPD** Update authority provides the authority to change the entries in an object.

Top

Authorization list (AUTL)

Specifies the authorization list whose entries are to be used to grant authority for the object specified. You must have authorization list management (*AUTLMGT) authority for the specified authorization list.

If a value is specified for this parameter, you cannot specify a value for the AUT, REFOBJ, or REFOBJTYPE parameters.

*NONE

The authorization list that secures the object is removed. If public authority in the object is *AUTL, it is changed to *EXCLUDE.

name Specify the name of the authorization list to be used.

Top

Reference object (REFOBJ)

Specifies the reference object to be queried to obtain authorization information. Those authorizations are given to the object specified by the OBJ and OBJTYPE parameters. Users authorized to the reference object are authorized in the same manner to the object for which authority is to be given. If the reference object is secured by an authorization list, that authorization list secures the object specified by the OBJ and OBJTYPE parameters.

If a value is specified for this parameter, you cannot specify a value for the AUT or AUTL parameters.

name Specify the name of the reference 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 used.

name Specify the name of the library to be searched.

Top

Reference object type (REFOBJTYPE)

Specifies the object type of the reference object specified for the **Reference object (REFOBJ)** parameter.

*OBJTYPE

The object type of the reference object is the same as the object type specified for the **Object type (OBJTYPE)** parameter.

object-type

Specify the object type of the reference object. 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).

Top

Reference ASP device (REFASPDEV)

Specifies the auxiliary storage pool (ASP) device name where the library that contains the reference object (REFOBJ parameter) is located. If the reference object's library resides in an ASP that is not part of the library name space associated with the job, this parameter must be specified to ensure the correct object is queried for authorities.

* The ASPs that are currently part of the job's library name space will be searched to locate the reference object. This includes the system ASP (ASP number 1), all defined basic user ASPs (ASP numbers 2-32), and, if the job has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs will be searched to locate the reference object. No independent ASPs will be searched, even if the job has an ASP group.

name Specify the device name of the independent ASP to be searched to locate the reference object. The independent ASP must have been activated (by varying on the ASP device) and have a status of AVAILABLE. The system ASP and basic user ASPs will not be searched.

Top

Replace authority (REPLACE)

Specifies whether the authorities replace the user's current authorities.

*NO The authorities are given to the user, but no authorities are removed, unless you are granting *EXCLUDE authority.

*YES The user's current authorities are removed, then the authorities are given to the user.

Top

Examples

Example 1: Granting Authority to All Users

```
GRTOBJAUT OBJ(USERLIB/PROGRAM1) OBJTYPE(*PGM) USER(*PUBLIC)
```

This command gives authority to use the object named PROGRAM1 to all users of the system who do not have authorities specifically given to them, who are not on an authorization list, whose user groups do not have authority to the object, or whose user groups are not on the authorization list. The object is a program (*PGM) located in the library named USERLIB. Because the AUT parameter is not specified, the authority given to all users is change authority. This allows all users to run the program and to debug it.

Example 2: Granting Object Management Authority

```
GRTOBJAUT OBJ(ARLIB/PROGRAM2) OBJTYPE(*PGM) USER(TMSMITH)
AUT(*OBJMGT)
```

This command gives object management authority to user named TMSMITH. This authority allows TMSMITH to grant to others personally possessed authorities for the object named PROGRAM2, which is a program located in the library named ARLIB.

Example 3: Granting Authority to Users on Authorization List

```
GRTOBJAUT OBJ(MYLIB/PRGM3) OBJTYPE(*PGM) AUTL(KLIST)
```

This command gives to users the authority specified for them on authorization list KLIST for the object named PRGM3. The object is a program located in library MYLIB.

Error messages

*ESCAPE Messages

CPF22A0

Authority of *AUTL is allowed only with USER(*PUBLIC).

CPF22A1

OBJTYPE(*AUTL) not valid on this command.

CPF22A2

Authority of *AUTL not allowed for object type *USRPRF.

CPF22A3

AUTL parameter not allowed for object type *USRPRF.

CPF22A9

Authority of *AUTL cannot be specified.

CPF22DA

Operation on file &1 in &2 not allowed.

CPF2207

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

CPF2208

Object &1 in library &3 type *&2 not found.

CPF2209

Library &1 not found.

CPF2210

Operation not allowed for object type *&1.

CPF2211

Not able to allocate object &1 in &3 type *&2.

CPF2216

Not authorized to use library &1.

CPF2223

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

CPF2227

One or more errors occurred during processing of command.

CPF2236

AUT input value not supported.

CPF2243

Library name &1 not allowed with OBJ(generic name) or OBJ(*ALL).

CPF2245

Process profile not owner of object &1 in &3 type *&2.

CPF2253

No objects found for &1 in library &2.

CPF2254

No libraries found for &1 request.

CPF2273

Authority may not have been changed for object &1 in &3 type *&2 for user &4.

CPF2283

Authorization list &1 does not exist.

CPF2290

*EXCLUDE cannot be specified with another authority.

CPF9804

Object &2 in library &3 damaged.

Top

Grant User Authority (GRTUSRAUT)

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

Parameters
Examples
Error messages

The Grant User Authority (GRTUSRAUT) command grants authority to a user by referring to another user profile.

Note: You should use group support or authorization lists instead of the Grant User Authority (GRTUSRAUT) command support whenever possible for better performance in granting authority and the subsequent SAVSYS or SAVSECDTA function.

If a security officer issues this command, the authorities in the user profile are granted to the receiving user, including object management authority.

If this command is run by the owner of the user profile, all authorities for each object owned are granted, including object management authority.

For objects that the user profile being referred to does not own but is authorized to use, the user of this command must have object management authority and the authorities being granted for the object, or must own the object. Otherwise, no authority is granted for the object.

Ownership of objects or authorities held by a user profile cannot be changed by this command. Authorities to objects granted to a user profile are added to any authorities that the user profile already had.

Restriction: The following user profiles cannot be specified for either of the parameters on this command:

QAUTPROF, QCLUMGT, QCLUSTER, QCOLSRV, QDBSHR, QDBSHRDO, QDFTOWN, QDIRSRV, QDLFM, QDOC, QDSNX, QEJB, QEJBSVR, QGATE, QIPP, QLPAUTO, QLPINSTALL, QMGTC, QMSF, QNETSPLE, QNFSANON, QNTP, QPEX, QPM400, QRJE, QSNADS, QSPL, QSRVAGT, QSYS, QTCM, QTCP, QTMHHTP1, QTMHHTTP, QTSTRQS, QYCMCIMOM, QYPSJSVR

Top

Parameters

Keyword	Description	Choices	Notes
USER	User	Name	Required, Positional 1
REFUSER	Referenced user	Name	Required, Positional 2

Top

User (USER)

This is a required parameter.

The name of the user profile to whom authority is being granted.

Referenced user (REFUSER)

This is a required parameter.

The name of the user profile being referred to for authority.

Examples

Example 1: Running GRTUSRAUT under QSECOFR User Profile

```
GRTUSRAUT  USER(USRB)  REFUSER(USRA)
```

This command grants the user profile USRB the same authorities that USRA has for all objects that USRA owns (including object management authority) or has authority to.

Example 2: Running GRTUSRAUT under User Profile USRA

```
GRTUSRAUT  USER(USRB)  REFUSER(USRC)
```

This command grants the user profile USRB the same authorities that USRC has for all objects that USRC has authorities to only if USRA, entering this command, has object management authority to the objects or is the owner of the objects being referred to.

Error messages

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2211

Not able to allocate object &1 in &3 type *&2.

CPF2213

Not able to allocate user profile &1.

CPF2217

Not authorized to user profile &1.

CPF2222

Storage limit is greater than specified for user profile &1.

CPF2223

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

CPF2252

Authority given to &2 objects. Authority not given to &3 objects.

Grant User Permission (GRTUSRPMN)

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

Parameters
Examples
Error messages

The Grant User Permission (GRTUSRPMN) command allows you to grant permission for a user to handle documents and folders or to perform OfficeVision/400-related tasks on behalf of another user. Access is restricted to documents, folders, and mail items that are not personal. The users specified must be enrolled in the system distribution directory before you run this command.

Restrictions: The user must have *ALLOBJ authority to grant permission for a user to work on behalf of another user.

Top

Parameters

Keyword	Description	Choices	Notes
TOUSER	To user profile	<i>Name</i>	Required, Positional 1
FORUSER	For user profile	Single values: *CURRENT Other values (up to 300 repetitions): <i>Name</i>	Optional, Positional 2

Top

To user profile (TOUSER)

Specifies the name of the user profile that is permitted to work on behalf of the user specified on the **For user profile** prompt (FORUSER parameter). Access is restricted to OfficeVision/400 documents, folders, and mail items that are not personal. The user profile must exist, and the user must be enrolled in the system distribution directory before you run this command.

Top

For user profile (FORUSER)

Specifies the names of the user profiles for which the user specified on the **To user profile** prompt (TOUSER parameter) works. The users must be enrolled in the system distribution directory before you run this command.

You can enter multiple values for this parameter.

The possible values are:

***CURRENT**

You are granting permission to someone working on your behalf.

user-profile-name

Specify the name of the user profile on whose behalf the user specified on the **To user profile** prompt (TOUSER parameter) works.

Examples

GRTUSRPMN TOUSER(JUDY) FORUSER(PEGGY)

JUDY is the administrative assistant for an executive. This command allows JUDY to work with documents or folders for PEGGY that are not personal.

Error messages

*ESCAPE Messages

CPF9007

User permission given for &1 users, not given &2 users.

CPF9009

System requires file &1 in &2 be journaled.

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.

Grant Workstation Object Aut (GRTWSOAUT)

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

Parameters
Examples
Error messages

The Grant Workstation Object Authority (GRTWSOAUT) command is used by one user to grant specific authority for the workstation object named in this command to another user or group of users. Workstation objects are used by the OS/400 Graphical Operations program.

Authority can be given to:

- Named users.
- Users (*PUBLIC) who do not have authority specifically given to them either for the object or for the authorization list.
- Groups of users who do not have any authority to the object or are not on the authorization list that secures the object.
- Users of the referenced workstation object (specified on the REFWSO parameter).
- Users on an established authorization list.

When AUT(*AUTL) is specified, the user can The authority for:

- All users who do not have authority specifically given to them for an object.
- Users who are not on the authorization list that secures the object.
- Users whose user group does not have authority specifically given to it.
- Users whose user group is not on the authorization list that secures the object.

This command can be used by an object owner, by the security officer, or by a user with object management authority for the specified object.

Restrictions:

1. A user must be either the owner of the object or have *ALL authority to use the AUTL parameter.
2. The user must have object management authority to the object to grant authority to the object.
3. AUT(*AUTL) can be specified only with USER(*PUBLIC). User profile names cannot be secured by an authorization list (*AUTL).
4. Only the owner of the object, or someone with all object authority (*ALLOBJ), can grant object management authority to a user.

Top

Parameters

Keyword	Description	Choices	Notes
WSOTYPE	Workstation object type	<i>Element list</i>	Required, Positional 1
	Element 1:	*TPLWRKARA, *WRKARA, *TPLPRTOL, *PRTOL, *TPLPRTL, *PRTL, *TPLOUTQ, *TPLOUTQL, *OUTQL, *TPLJOBQ, *JOBQ, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSG, *MSG, *TPLMSGQ, *TPLMSGSD, *MSGSD, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLIBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSET	

Keyword	Description	Choices	Notes
USER	Users	Single values: *PUBLIC Other values (up to 50 repetitions): <i>Qualifier list</i>	Optional, Positional 2
	Qualifier 1: Users	<i>Name</i>	
AUT	Authority	Single values: <u>*CHANGE</u> , *ALL, *USE, *EXCLUDE, *AUTL Other values (up to 7 repetitions): *OBJEXIST, *OBJMGT, *OBJOPR, *ADD, *DLT, *READ, *UPD	Optional, Positional 3
AUTL	Authorization list	<i>Name</i>	Optional
REFWSO	Reference workstation object	<i>Element list</i>	Optional
	Element 1:	*TPLWRKARA, *WRKARA, *TPLPRTOL, *PRTOL, *TPLPRTL, *PRTL, *TPLOUTQ, *TPLOUTQL, *OUTQL, *TPLJOBL, *JOBL, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSGSL, *MSGSL, *TPLMSGQ, *TPLMSGSL, *MSGSL, *TPLSGNUSL, *SGNUSL, *TPLOBJL, *OBJL, *TPLLBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSET	

Top

Workstation object type (WSOTYPE)

Specifies the workstation objects whose authority is to be edited.

This is a required parameter.

*TPLWRKARA

The work area template is the workstation object.

*WRKARA

The work area objects are the workstation objects.

*TPLPRTOL

The printer output list template is the workstation object.

*PRTOL

The printer output list objects are the workstation objects.

*TPLPRTL

The printer list template is the workstation object.

*PRTL The printer list objects are the workstation objects.

*TPLOUTQ

The output queue template is the workstation object.

*TPLOUTQL

The output queue list template is the workstation object.

*OUTQL

The output queue list objects are the workstation objects.

*TPLJOBL

The job list template is the workstation object.

*JOBL The job list objects are the workstation objects.

*TPLJOBQ

The job queue template is the workstation object.

*TPLJOBLOG

The job log template is the workstation object.

- *JOBLOG**
The job log objects are the workstation objects.
- *TPLJOBQL**
The job queue list template is the workstation object.
- *JOBQL**
The job queue list objects are the workstation objects.
- *TPLMSGL**
The message list template is the workstation object.
- *MSGL**
The message list objects are the workstation objects.
- *TPLMSGQ**
The message queue template is the workstation object.
- *TPLMSGSEND**
The message sender template is the workstation object.
- *MSGSEND**
The message sender objects are the workstation objects.
- *TPLSGNUSL**
The signed-on user list template is the workstation object.
- *SGNUSL**
The signed-on user list objects are the workstation objects.
- *TPLOBJL**
The object list template is the workstation object.
- *OBJL** The object list objects are the workstation objects.
- *TPLLIBSL**
The library list template is the workstation object.
- *LIBSL**
The library list objects are the workstation objects.
- *TPLLIB**
The library template is the workstation object.
- *TPLLAUNCH**
The job submitter template is the workstation object.
- *LAUNCH**
The job submitter objects are the workstation objects.
- *PRSSET**
The personal settings objects are the workstation objects.

Top

Users (USER)

Specifies one or more users to whom authorities for the named object are to be given. If user names are specified, the authorities are given specifically to those users. Authority given by this command can be revoked specifically by the Revoke Workstation Object Authority (RVKWSOAUT) command.

This is a required parameter unless either the **Reference workstation object (REFWSO)** parameter or **Authorization list (AUTL)** parameter is specified.

***PUBLIC**

All users of the system, who do not have authority specifically given to them for the object, who are not on the authorization list, whose user group does not have any authority, or whose user group is not on the authorization list, are authorized to use the object as specified on the AUT parameter.

name Specify the name of one or more user profiles. A maximum of 50 user profile names can be specified.

Top

Authority (AUT)

Specifies the authority to be given to the users specified on the **Users (USER)** parameter.

***CHANGE**

The user can perform all operations on the object except those limited to the owner or controlled by object existence (*OBJEXIST) and object management (*OBJMGT) authorities. The user can change and perform basic functions on the object. *CHANGE authority provides object operational (*OBJOPR) authority and all data authority. If the object is an authorization list, the user cannot add, change, or remove users.

***ALL** The user can perform all operations except those limited to the owner or controlled by authorization list management (*AUTLMGT) authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user also can change ownership of the object.

***USE** The user can perform basic operations on the object, such as running a program or reading a file. The user cannot change the object. Use (*USE) authority provides object operational (*OBJOPR), read (*READ), and execute (*EXECUTE) authorities.

***EXCLUDE**

The user cannot access the workstation object.

***AUTL**

The public authority of the authorization list specified on the AUTL parameter is used for the public authority for the object.

Note: You can specify AUT(*AUTL) only when USER(*PUBLIC) is also specified.

A maximum of ten of the following values can be specified.

***OBJALTER**

Object alter authority provides the authority needed to alter the attributes of an object. If the user has this authority 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. If the user has this authority on an SQL package, the user can change the attributes of the SQL package. This authority is currently only used for database files and SQL packages.

***OBJMGT**

Object management authority provides the authority to The security for the object, move or rename the object, and add members to database files.

***OBJEXIST**

Object existence authority provides the authority to control the object's existence and ownership. If a user has special save system authority (*SAVSYS), object existence authority is not needed to perform save restore operations on the object.

***OBJOPR**

Object operational authority provides authority to look at the description of an object and use the object as determined by the data authority that the user has to the object.

*OBJREF

Object reference authority provides the authority needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user has this authority on a physical file, the user can add referential constraints in which the physical file is the parent. This authority is currently only used for database files.

The possible data authorities are:

***ADD** Add authority provides the authority to add entries to an object (for example, job entries to an queue or records to a file).

***DLT** Delete authority provides the authority to remove entries from an object.

*EXECUTE

Execute authority provides the authority needed to run a program or locate an object in a library.

*READ

Read authority provides the authority needed to get the contents of an entry in an object or to run a program.

***UPD** Update authority provides the authority to change the entries in an object.

Top

Authorization list (AUTL)

Specifies the authorization list whose members are to be given authority for the object specified for the **Workstation object type (WSOTYPE)** parameter. You must have authorization list management (*AUTLMGT) authority for the specified authorization list.

This is a required parameter unless either the **Users (USER)** parameter or the **Reference workstation object (REFWSO)** parameter is specified.

Top

Reference workstation object (REFWSO)

Specifies the workstation object referred to for authorizations. These authorizations are given to the object specified for the **Workstation object type (WSOTYPE)** parameter. Users authorized to the reference object are authorized in the same manner to the object for which authority is to be given. If the reference object is secured by an authorization list, that authorization list secures the object specified on the WSOTYPE parameter.

This is a required parameter unless either the **Users (USER)** parameter or the **Authorization list (AUTL)** parameter is specified.

Top

Examples

```
GRTWSOAUT  WSOTYPE(*TPLWRKARA)  AUTL(KLIST)
```

This command gives authority to the work are template to the users with authority specified for them on the authorization list KLIST.

Top

Error messages

Unknown

[Top](#)

Hold Communications Device (HLDCMNDEV)

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

Parameters
Examples
Error messages

The Hold Communications Device (HLDCMNDEV) command allows the operator to hold communication through the specified device. Communications are restarted with the Release Communications Device (RLSCMNDEV) command or by varying the device off and then on with the Vary Configuration (VRYCFG) command.

Restriction: This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, and QSRVBAS user profiles have private authorities to use the command.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i>	Required, Positional 1
OPTION	Option	*CNTRL, *IMMED	Optional, Positional 2

Top

Device (DEV)

Specifies the name of the device whose communications are to be held. Devices whose communications are held are:

DEV Value

Device

- 3180 Display station
- 3277 Display station
- 3278 Display station
- 3279 Display station
- 3287 Printer (work station)
- 5219 Printer (work station)
- 5224 Printer (work station)
- 5225 Printer (work station)
- 5251 Display station
- 5252 Display station
- 5256 Printer (work station)
- 5291 Display station

5292 Display station

PLU1 Primary logical unit, type 1 (for SNA)

BSC Binary synchronous device (Base and RJE)

BSCT This &sys. system as a BSC multipoint tributary station

APPC Logical unit in advanced program-to-program communications (APPC) network

This is a required parameter.

Top

Option (OPTION)

Specifies the manner in which communication is held with this device.

The possible values are:

*CNTRLD

The specified device is not capable of communications at the next OPEN or ACQUIRE operation. The controlled option allows any program using the communications device to continue to do input/output operations, but no new uses of the device are started.

*IMMED

The specified device is not capable of communications at the next READ, WRITE, OPEN, or ACQUIRE operation. The immediate option stops a communications device immediately, and a permanent input/output error is sent to the program.

Top

Examples

```
HLDCMNDEV DEV(WSPR05)
```

This command holds the communications capability of the device WSPR05 at the time of the next OPEN or ACQUIRE operation.

Top

Error messages

*ESCAPE Messages

CPF5920

Device &1 varied off or in diagnostic mode.

CPF5921

Device &1 not a communications device.

CPF5922

Device &1 already held with option *IMMED.

CPF5935

Error occurred during command processing.

CPF5984

Not authorized to perform function.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

[Top](#)

Hold Distribution Queue (HLDDSTQ)

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

Parameters
Examples
Error messages

The Hold Distribution Queue (HLDDSTQ) command prevents a distribution queue from being sent.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	<i>Character value</i>	Required, Positional 1
PTY	Priority	*NORMAL, *HIGH	Required, Positional 2

Top

Distribution queue (DSTQ)

Specifies the name of the distribution queue that is held. The queue must have been previously configured using the Configure Distribution Services (CFGDSTSRV) command or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Top

Priority (PTY)

Specifies whether the normal priority or high priority portion of the specified queue is held.

The possible values are:

***NORMAL**

The normal priority queue is for those distributions with a service level of data low.

***HIGH**

The high priority queue is for those distributions with a service level of fast, status, or data high.

Note: This value is not valid for a SystemView distribution services (SVDS) type of distribution queue.

This is a required parameter.

Top

Examples

Example 1: Holding the Normal Priority Portion of a Queue

```
HLDDSTQ DSTQ(CHICAGO) PTY(*NORMAL)
```

This command holds the normal priority portion of the CHICAGO distribution queue.

Example 2: Holding the High Priority Portion of a Queue

```
HLDDSTQ DSTQ(ATLANTA) PTY(*HIGH)
```

This command holds the high priority portion of the ATLANTA distribution queue.

Top

Error messages

***ESCAPE Messages**

CPF8802

Distribution queue &1 was not found.

CPF8805

Special value for System name/Group not permitted or not used correctly.

CPF8806

Value &1 not valid for system name or system group.

CPF881C

High priority queue not allowed for *SVDS distribution queue &1

CPF8812

Error occurred while processing distribution queues.

CPF8816

QSNADS communications subsystem is not active.

CPF8817

Distribution queue is held.

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.

Hold Job (HLDJOB)

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

Parameters
Examples
Error messages

The Hold Job (HLDJOB) command makes a job ineligible for processing by the system. The job is held until it is:

- Released by the Release Job (RLSJOB) command
- Cleared by the Clear Job Queue (CLRJOBQ) command
- Ended by the End Job (ENDJOB) command
- Ended (while the job is active) by the End Subsystem (ENDSBS) command, the End System (ENDSYS) command, or the Power Down System (PWRDWN SYS) command

Holding a job causes all threads within the job to be held.

Note: If you use this command to hold a job that has exclusive access to any resources on the system, these resources are not available to other jobs. Other jobs which require access to those resources will either fail or wait indefinitely.

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 held, 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. It is described in more detail in the Work Management book.

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	
SPLFILE	Hold spooled files	*NO, *YES	Optional, Positional 2
DUPJOB OPT	Duplicate job option	*SELECT, *MSG	Optional

Top

Job name (JOB)

Specifies the name of the job being held.

This is a required parameter.

Qualifier 1: Job name

name Specify the name of the job.

Qualifier 2: User

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

Qualifier 3: Number

000000-999999

Specify the system-assigned job number.

Note: If no user name or job number is specified, all jobs currently in the system are searched for the job name. If more than one occurrence of the specified name is found, a qualified job name must be specified.

Top

Hold spooled files (SPLFILE)

Specifies whether spooled output files created by the job being held are also held.

***NO** The spooled output files produced by the job are not held.

***YES** The spooled output files produced by the job are also held. Only those spooled output files which are on output queues in the library name space of the thread issuing this command will be held. If the **Spooled file action (SPLFACN)** job attribute is *DETACH and the job is ended while the spooled files are held, the spooled files cannot be released using the Release Job (RLSJOB) command. To release spooled files after the job has been removed from the system, use the Release Spooled File (RLSSPLF) command.

Top

Duplicate job option (DUPJOB OPT)

Specifies the action taken when duplicate jobs are found by this command.

***SELECT**

The selection display is shown when duplicate jobs are found during an interactive session. Otherwise, a message is issued.

***MSG** A message is issued when duplicate jobs are found.

Top

Examples

Example 1: Making a Job Ineligible for Processing

```
HLDJOB JOB(PAYROLL) SPLFILE(*YES)
```

This command makes the job named PAYROLL ineligible for processing. All spooled files for this job are also held.

Example 2: Holding a Job that has a Duplicate Name

```
HLDJOB JOB(DEPTXYZ/PAYROLL)
```


This command holds the job named PAYROLL submitted by a user operating under the user profile DEPTXYZ. The qualified form of the job name is used when jobs with duplicate names exist in the system. Spooled files are not held.

Top

Error messages

*ESCAPE Messages

CPF1E52

Not authorized to hold job &1.

CPF1E53

Job &1 has ended and cannot be held.

CPF1E54

Job &1 cannot be held.

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1321

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

CPF1332

End of duplicate job names.

CPF1340

Job control function not performed.

CPF1341

Reader or writer &3/&2/&1 not allowed as job name.

CPF1342

Current job not allowed as job name on this command.

CPF1343

Job &3/&2/&1 not valid job type for function.

CPF1344

Not authorized to control job &3/&2/&1.

CPF1345

Cannot hold job &3/&2/&1.

CPF1346

Job &3/&2/&1 already held.

CPF1347

Cannot hold job &3/&2/&1.

CPF1348

Job &3/&2/&1 held but unable to hold its files.

CPF1350

SPLFILE(*NO) specified but job &3/&2/&1 on OUTQ.

CPF1351

Function check occurred in subsystem for job &3/&2/&1.

CPF1352

Function not done. &3/&2/&1 in transition condition.

CPF1378

Job &3/&2/&1 not held at current time.

[Top](#)

Hold Job Queue (HLDJOBQ)

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

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

The Hold Job Queue (HLDJOBQ) command prevents the processing of all jobs currently waiting on the job queue and of all jobs that are added to the queue after the command is issued. This command has no effect on jobs that are running. Additional jobs can be placed on the job queue while it is held, but they are not processed. The jobs are held until a Release Job Queue (RLSJOBQ) command is issued. When a job queue is held, the jobs can be cleared with the Clear Job Queue (CLRJOBQ) command or a specific job can be canceled by the End Job (ENDJOB) command.

Restriction: The QLPINSTALL job queue cannot be held.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
JOBQ	Job queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Job queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

[Top](#)

Job queue (JOBQ)

This is a required parameter.

Specifies the name of the job queue that will have its current and future entries withheld from further processing.

The possible library values are:

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

***CURLIB**

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

library-name

Specify the library where the job queue is located.

job-queue-name

Specify the name of the job queue to be held.

[Top](#)

Examples

HLDJOBQ JOBQ(QBATCH)

This command prevents the processing of the jobs currently on the QBATCH job queue and any jobs added to the queue. They are held until the queue is released or cleared. Individual jobs can also be ended with the ENDJOB command, which removes the job from the job queue.

[Top](#)

Error messages

*ESCAPE Messages

CPF2207

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

CPF2240

User &7 not authorized to use *&5 &6/&4.

CPF3307

Job queue &1 in &2 not found.

CPF3330

Necessary resource not available.

CPF3425

Job queue &1 in &2 already held.

[Top](#)

Hold Job Schedule Entry (HLDJOBSCDE)

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

Parameters
Examples
Error messages

The Hold Job Schedule Entry (HLDJOBSCDE) command allows you to hold an entry, all entries, or a set of entries in the job schedule. Each job schedule entry contains the information needed to automatically submit a job to be run once or at regularly scheduled intervals.

If you hold a job schedule entry:

- The entry is held until it is released using the Release Job Schedule Entry (RLSJOBSCDE) or Work with Job Schedule Entries (WRKJOBSCDE) command.
- The job is not submitted when it is released, even if a date and time at which it was scheduled to be submitted passed while the entry was held.

Restriction: To hold entries, you must have job control (*JOBCTL) special authority; otherwise you can hold only those entries that you added.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	Generic name, name, *ALL	Required, Positional 1
ENTRYNBR	Entry number	000001-999999, <u>*ONLY</u> , *ALL	Optional

Top

Job name (JOB)

Specifies the name of the job schedule entry.

This is a required parameter.

***ALL** All of the job schedule entries for which you have authority are held. If JOB(*ALL) is specified, ENTRYNBR(*ALL) must also be specified.

generic-name

Specify the generic name used to find job schedule entries. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all entries with names that begin with the generic name, and for which the user has authority, are held. If a generic name is specified, ENTRYNBR(*ALL) must also be specified. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete job name.

name Specify the name of the job schedule entry that you want held.

Top

Entry number (ENTRYNBR)

Specifies the number of the job schedule entry you want held. The message sent when an entry is successfully added contains the entry number. You can also determine the entry number by using the Work with Job Schedule Entries (WRKJOBSCDE) command. Press F11 from the WRKJOBSCDE display to show the entry numbers of the selected entries.

*ONLY

Only one entry in the job schedule has the job name specified for the JOB parameter. If *ONLY is specified and more than one entry has the specified job name, no entries are held and an error message is sent.

***ALL** All entries with the specified job name are held.

000001-999999

Specify the number of the job schedule entry you want held.

Top

Examples

Example 1: Holding a Job Schedule Entry

```
HLDJOBSCDE JOB(CLEANUP)
```

This command holds the job schedule entry with the job name CLEANUP.

Example 2: Holding All Job Schedule Entries

```
HLDJOBSCDE JOB(*ALL) ENRYNBR(*ALL)
```

This command holds all entries in the job schedule.

Example 3: Holding an Individual Job Schedule Entry

```
HLDJOBSCDE JOB(PAYROLL) ENRYNBR(*ONLY)
```

This command holds the entry PAYROLL in the job schedule.

Example 4: Holding a Generic Job Schedule Entry

```
HLDJOBSCDE JOB(PAY*) ENRYNBR(*ALL)
```

This command holds all entries in the job schedule that have the prefix PAY in their names.

Top

Error messages

*ESCAPE Messages

CPF1628

Job schedule entry &3 number &4 not found.

CPF1629

Not authorized to job schedule &1.

CPF1630

Not authorized to job schedule entry &3 number &4.

CPF1632

Job schedule entry &3 number &4 damaged.

CPF1636

More than one entry with specified entry job name found.

CPF1637

Job schedule &1 in library &2 in use.

CPF1638

Job schedule entry &3 number &4 in use.

CPF1640

Job schedule &1 in library &2 does not exist.

CPF1641

Job schedule &1 in library &2 damaged.

CPF1645

No job schedule entries found for specified name.

CPF1646

Entry number must be *ALL when generic name specified.

CPF1647

&3 entries successfully held, &4 entries not held.

CPF1649

Entry number must be *ALL.

Top

Hold Output Queue (HLDOUTQ)

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

Parameters
Examples
Error messages

The Hold Output Queue (HLDOUTQ) command prevents all currently waiting spooled files, and all spooled files that are added to the output queue after the command is issued, from being processed by a spooling writer. This command has no effect on jobs currently running and adding spooled files to the output queue. It also has no effect on the spooled output that is being produced by a spooling writer at the time the command is issued. When the spooling writer completes all copies of the current output file, it cannot begin the output for any other files until the queue is released.

Top

Parameters

Keyword	Description	Choices	Notes
OUTQ	Output queue	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Output queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Top

Output queue (OUTQ)

This is a required parameter.

Specifies the name of the output queue that will have its current and future spooled files withheld from further processing.

The possible library values are:

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

***CURLIB**

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

library-name

Specify the library where the output queue is located.

output-queue-name

Specify the name of the output queue to hold.

Top

Examples

HLDOUTQ OUTQ(QPRINT)

This command prevents the processing of the spooled files on the QPRINT queue and any spooled files added to the queue. They are held until the queue is released or cleared. A specific job (with its spooled files) can also be ended with the ENDJOB command, which removes the spooled files from the output queue.

Top

Error messages

*ESCAPE Messages

CPF2207

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

CPF3330

Necessary resource not available.

CPF3357

Output queue &1 in library &2 not found.

CPF3426

Output queue &1 in library &2 already held.

Top

Hold Reader (HLDRDR)

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

Parameters
Examples
Error messages

The Hold Reader (HLDRDR) command immediately stops the activity of the specified spooling reader. The reader itself is not ended, nor is its associated input device made available to the system. The reader remains inactive until a Release Reader (RLSRDR) or End Reader (ENDRDR) command is issued. Data is not lost when the reader is held.

Top

Parameters

Keyword	Description	Choices	Notes
RDR	Reader	<i>Name</i>	Required, Positional 1

Top

Reader (RDR)

This is a required parameter.

Specifies the name of the spooling reader to be held.

reader-name

Specify the name of the reader to be held.

Top

Examples

HLDRDR RDR(QDKT)

This command causes the diskette reader QDKT to immediately stop reading data. To release the reader, so that it can continue to read data, a Release Reader (RLSRDR) command must be entered. If the End Reader (ENDRDR) command is used, the reader is stopped and the job that was being read in is lost because no job entry was added to the job queue.

Top

Error messages

*ESCAPE Messages

CPF1E52

Not authorized to hold job &1.

CPF1E53

Job &1 has ended and cannot be held.

CPF1E54

Job &1 cannot be held.

CPF1317

No response from subsystem for job &3/&2/&1.

CPF1340

Job control function not performed.

CPF1345

Cannot hold job &3/&2/&1.

CPF1347

Cannot hold job &3/&2/&1.

CPF1350

SPLFILE(*NO) specified but job &3/&2/&1 on OUTQ.

CPF1351

Function check occurred in subsystem for job &3/&2/&1.

CPF1352

Function not done. &3/&2/&1 in transition condition.

CPF1378

Job &3/&2/&1 not held at current time.

CPF3312

Reader &1 neither active nor on job queue.

CPF3330

Necessary resource not available.

CPF3333

Reader &3/&2/&1 already held.

CPF3490

Not authorized to specified reader.

Top

Hold Spooled File (HLDSPLF)

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

Parameters
 Examples
 Error messages

The Hold Spooled File (HLDSPLF) command stops the specified spooled file from additional processing by a spooled writer. If the file is being produced on an output device, the writer stops processing that file and gets the next file to be processed. When the file is released and selected for output, it is again processed starting at the beginning of the file. If multiple copies are being produced for the file when it is held, the incomplete copy is produced from the beginning again and any remaining copies follow it.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Spooled file	Name, *SELECT	Required, Positional 1
JOB	Job name	Single values: * Other values: <i>Qualified job name</i>	Optional, Positional 2
	Qualifier 1: Job name	Name	
	Qualifier 2: User	Name	
	Qualifier 3: Number	000000-999999	
SPLNBR	Spooled file number	1-999999, *ONLY, *LAST, *ANY	Optional, Positional 3
JOBSYSNAME	Job system name	Name, *ONLY, *CURRENT, *ANY	Optional
CRTDATE	Spooled file created	Single values: *ONLY, *LAST Other values: <i>Element list</i>	Optional
	Element 1: Creation date	Date	
	Element 2: Creation time	Time, *ONLY, *LAST	
SELECT	Select files for	<i>Element list</i>	Optional
	Element 1: User	Name, *CURRENT, *ALL	
	Element 2: Print device	Name, *ALL, *OUTQ	
	Element 3: Form type	Character value, *ALL, *STD	
	Element 4: User data	Character value, *ALL	
	Element 5: ASP	1-32, *ALL, *ASPDEV	
ASPDEV	ASP device	Name, *, *SYSBAS, *CURASPGRP	Optional
OPTION	When to hold file	*IMMED, *PAGEEND	Optional

Top

Spooled file (FILE)

This is a required parameter.

Specifies the name of the spooled file to hold.

The possible values are:

***SELECT**

All spooled files that meet the selection values specified on the **Select files for** prompt (SELECT parameter) are held. This value is mutually exclusive with values specified on the **Job name** prompt (JOB parameter), **Spooled file number** prompt (SPLNBR parameter), **Job system name** prompt (JOBSYSNAME parameter), and **Spooled file created** prompt (CRTDATE parameter).

spooled-file-name

Specify the name of the spooled file to hold.

Top

Job name (JOB)

Specifies the name of the job that created the file being held.

The possible values are:

***** The job that issued this Hold Spooled File (HLDSPLF) command is the job that produced this file.

job-name

Specify the name of the job that created the file being held.

user-name

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

job-number

Specify the system-assigned job number.

Top

Spooled file number (SPLNBR)

Specifies the number of the spooled file that was created by the specified job.

The possible values are:

***ONLY**

Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

***LAST**

The spooled file with the highest number and the specified file name is used.

***ANY** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

spooled-file-number

Specify the number of the spooled file having the specified file name that is being held.

Top

Job system name (JOBSYSNAME)

Specifies the name of the system where the job that created the spooled file (JOB parameter) ran. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

*ONLY

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and spooled file create date and time.

*CURRENT

The spooled file created on the current system with the specified job name, user name, job number, spooled file name, spooled file number, and create date and time is used.

*ANY The job system name is not used to determine which spooled file is used. Use this value when the spooled file create date and time parameter is to take precedence over the job system name when selecting a spooled file.

job-system-name

Specify the name of the system where the job that created the spooled file ran.

Top

Spooled file created (CRTDATE)

Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met.

The possible single values are:

*ONLY

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

*LAST

The spooled file with the latest create date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

The possible create date value is:

spooled-file-create-date

Specify the date the spooled file was created.

The possible create time values are:

*ONLY

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.

*LAST

The spooled file with the latest create time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.

spooled-file-create-time

Specify the time the spooled file was created.

Top

Select files for (SELECT)

Specifies which group of files are to be held. Positional values can be specified to select the files: the user that created the file, the device that the file is queued for, the form type specified, the user data tag associated with the file, or the auxiliary storage pool the file is in. Only files that meet each of the values are selected.

The possible values for the user are:

*CURRENT

Only files created by the user running this command are selected.

***ALL** Files created by all users are selected.

user-name

Specify a user name. Only files created by that user name are selected.

The possible values for the device are:

***ALL** Files on any device-created or user-created output queue are selected.

***OUTQ**

All files on any user-created output queue are selected. A user-created output queue is any output queue that is not automatically created by a device. A user-created output queue does not generally have the same name as a device, but if it does, it does not reside in library QUSRSYS.

device-name

Specify a device name. Only files on the device created output queue for that device are selected. A device created output queue is one that has the same name as a device and resides in the QUSRSYS library. Unless it already exists, it will automatically be created by the system when the device is created. A device created output queue cannot be deleted.

The possible values for the form type are:

***ALL** Files for all form types are selected.

***STD** Only files that specify the standard form type are selected.

form-type

Specify the form type to select the file.

The possible values for the user data are:

***ALL** Files with any user data tag specified are selected.

user-data

Specify the user data tag to select the file.

The possible values for Auxiliary Storage Pool number (ASP) are:

***ALL** All files as specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

***ASPDEV**

Files specified in the Auxiliary Storage Pool Device (ASPDEV) parameter are selected.

ASP-number

Specify the auxiliary storage pool (ASP) of the files being selected. Valid values are 1 to 32.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool device name from which spooled files are to be selected. This parameter is only valid if the ASP number (ASP) element of the Select parameter is *ALL or *ASPDEV.

The possible values are:

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

*SYSBAS

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

*CURASGRP

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

auxiliary-storage-pool-device-name

Files which are found in the specified primary or secondary ASP are selected. Only primary or secondary ASPs which are in the thread's ASP group may be specified. If no ASP group is associated with the thread, an error will be issued.

Top

When to hold file (OPTION)

Specifies which option to use when holding the spooled file.

Note: Specifying an option when the file is not being written has no effect.

The possible values are:

*IMMED

The file is to be held as soon as possible.

*PAGEEND

The file is to be held on a page boundary.

Top

Examples

Example 1: Holding a File Created by Another Job

```
HLDSPLF FILE(SHIPITEMS) JOB(00009/JONES/ORDER)
```

This command withholds the spooled file SHIPITEMS, created by the job ORDER, from additional processing.

Example 2: Holding a File at a Page Boundary

```
HLDSPLF FILE(QPJOBLOG) OPTION(*PAGEEND)
```

This command holds the spooled file QPJOBLOG at a page boundary.

Example 3: Holding a File Immediately

```
HLDSPLF FILE(QPJOBLOG) OPTION(*IMMED)
```

This command holds the spooled file QPJOBLOG immediately. Holding a spooled file by specifying this option causes the CHGSPLFA command RESTART(*NEXT) to be inaccurate if the spooled file is currently being processed by a spool writer.

Top

Error messages

*ESCAPE Messages

CPF337E

ASP device &1 not in current ASP group for thread.

CPF337F

ASP device &1 not allowed with ASP number &2.

CPF33D0

Printer &1 does not exist.

CPF33D1

User &1 does not exist.

CPF3303

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

CPF3309

No files named &1 are active.

CPF3330

Necessary resource not available.

CPF3337

File &1 number &8 already held or saved.

CPF3340

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

CPF3342

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

CPF3343

Duplicate job names found.

CPF3344

File &1 number &8 no longer in the system.

CPF3357

Output queue &1 in library &2 not found.

CPF34A4

File &1 number &8 not held or deleted.

CPF3492

Not authorized to spooled file.

CPF9825

Not authorized to device &1.

CPF9833

*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF8ED

Device description &1 not correct for operation.

Hold Writer (HLDWTR)

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

Parameters
Examples
Error messages

The Hold Writer (HLDWTR) command stops the specified writer at the end of a record, at the end of a spooled file, or at the end of a printed page. If multiple copies of a file are produced, the writer can be held at the end of the copy currently being produced. The writer is not stopped and the device is not made available to the system. The writer remains inactive until a Release Writer (RLSWTR) or End Writer (ENDWTR) command is issued. Data is not lost when the writer is held.

Top

Parameters

Keyword	Description	Choices	Notes
WTR	Writer	<i>Name</i>	Required, Positional 1
OPTION	When to hold writer	<u>*IMMED</u> , *CNTRL D, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

This is a required parameter.

Specifies the name of the spooling writer being held.

writer-name

Specify the name of the writer to be held.

Top

When to hold writer (OPTION)

Specifies when the spooling writer should stop producing output.

The possible values are:

*IMMED

The writer stops immediately after it has written the last record, in the current block of records, to the output device. Each time the writer finishes producing a block of records on a device, it makes another I/O request to get the next block from the file being spooled to the device. If *IMMED is specified, the writer stops only after it has written the last record in the block being processed, which (for diskette output) is a complete diskette record being written on diskette.

When *IMMED is specified for printed output, the writer stops anywhere within or at the end of a print line or at the end of a complete block, which may not be at the end of a line. This is because some data records (which are blocked to improve performance) may be split in two, with the first part of a record at the end of one block and the last part of the record at the beginning of

the next block. If only one copy of the file is being produced or if the last copy is being produced, the entry for the file is removed from the output queue when the output is completed.

***CNTRLD**

The writer stops at the end of the current copy of the file. If only one copy of the file is to be produced or if the last copy is being produced, the entry for the file is removed from the output queue when the output is completed.

***PAGEEND**

The writer is held at the end of a page. This value is valid only when the spooling writer is a printer writer.

[Top](#)

Examples

```
HLDWTR WTR(PRINTER) OPTION(*CNTRLD)
```

This command stops the writer named PRINTER at the end of the current file. The writer is held until an RLSWTR (Release Writer) or ENDWTR (End Writer) command is issued.

[Top](#)

Error messages

*ESCAPE Messages

CPF1340

Job control function not performed.

CPF3313

Writer &1 not active nor on job queue.

CPF3330

Necessary resource not available.

CPF3331

Not authorized to control writer &3/&2/&1.

CPF3332

Writer &3/&2/&1 already held.

CPF3334

Previous hold to writer &3/&2/&1 pending.

CPF3438

*PAGEEND not valid for writer &3/&2/&1.

[Top](#)

If (IF)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The If (IF) command evaluates a logical expression and conditionally processes CL procedure commands according to the evaluation of the expression. If the logical expression is true (a logical 1), the command (or the group of commands in a Do group) specified in the THEN parameter is processed, and the ELSE command with its associated command or Do group is not processed. If the result of the logical expression is false (a logical 0), the command specified in the THEN parameter is not processed and control passes to the next command. If that command is an ELSE command, the command or Do group specified in that command is processed. If the ELSE command is not specified, control passes to the next command.

When a DO command is specified, either in the THEN parameter of the IF command or in the CMD parameter of the ELSE command, the Do group is bypassed if the result of the expression is not the one needed for the group being processed. That is, control passes to the command that follows the ENDDO command associated with that DO.

When the command or Do group specified by the THEN parameter or the ELSE command is completed (and no GOTO command has been processed), control passes to the next command following the command or Do group specified by the ELSE command. If a GOTO command is processed, control is passed to the command with the label specified by the GOTO command, and processing continues from that command.

The following command sequence shows this flow. In this example, &TESTSW is a logical variable.

```
IF &TESTSW DO
  Group A (group of CL commands)
  .
  .
  ENDDO
ELSE DO
  Group B (group of CL commands)
  .
  .
  ENDDO
Group C (continued CL commands)
.
.
```

The IF command tests the logical variable &TESTSW. If a true condition exists (&TESTSW contains a value of '1'), the commands in Group A are processed, then control passes to the commands in Group C. If a false condition exists (&TESTW contains a value of 0), the commands in group A are bypassed, the commands in Group B are processed, then control passes to the commands in Group C.

Restrictions:

- The IF command is valid only in CL procedures.
- Up to ten levels of nested IF and ELSE commands are allowed.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
COND	Condition	<i>Logical value</i>	Required, Positional 1
THEN	Command	<i>Command string</i>	Optional, Positional 2

Top

Condition (COND)

Specifies the logical expression that is evaluated to determine a condition in the program and what is done next. Refer to "Logical Expressions" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> for a description of logical expressions. Note that variables, constants, and the %SUBSTRING, %SWITCH, and %BINARY built-in functions can be used within the expression.

This is a required parameter.

logical-value

Specify the name of a CL logical variable or a logical expression.

Top

Command (THEN)

Specifies the command or group of commands (in a Do group) that are processed if the result of evaluating the logical expression is true. After the command or Do group is processed, control is passed to the next command *after* the ELSE command associated with this IF command. If the result is true, the ELSE command associated with the IF command is not processed. If the command specified in this parameter is a DO command, all commands within the Do group are considered to be the command specified by the parameter.

If the command specified by the THEN keyword is not coded on the same line when the keyword is coded, the THEN keyword must be immediately followed (on the same line) either by the left parenthesis or by a plus sign (+) or a minus sign (-) to show continuation. (A blank cannot immediately follow any keyword.) The command and the right parenthesis can then be coded on the next line. For example:

```
IF COND(&A *EQ &B) THEN(      +
    GOTO C)
```

If any part of the command specified by the THEN parameter continues on the next line, a continuation character (+ or -) must be specified.

If a DO command is specified, only the DO command (not the commands specified within the Do group) is within the parentheses. For example:

```
IF COND(&A *EQ &B) THEN(DO)
  CMD1
  CMD2
  .
  .
ENDDO
```

If no command is specified for the THEN parameter (a null THEN) and the ELSE command immediately follows it, the ELSE is processed if the IF expression is false and it is skipped if the expression is true.

Any CL command can be specified for the THEN parameter, except the following commands:

- ELSE
- PGM, ENDPGM
- ENDDO
- MONMSG
- DCL, DCLF
- WHEN, OTHERWISE, ENDSELECT

The command can be another IF, unless there are already ten levels of nested IF and ELSE commands.

Top

Examples

```
IF COND(&A *EQ &B) THEN(GOTO X)
IF (&A *EQ &B) THEN(GOTO X)
IF (&A *EQ &B) (GOTO X)
IF COND(&A *EQ &B) THEN(GOTO X)
```

The examples above show a number of different ways the IF command can be specified to test a condition and branch to a label. In each of these examples, if &A equals &B, control passes to a CL command that has a label named X.

```
IF COND(&TESTSW) THEN(CHGVAR VAR(&A) VALUE(23))
```

If &TESTSW has a logical value of 1 (true), the CHGVAR command is processed to set &A to decimal 23; if &TESTSW has a logical value of 0 (not true), the Change Variable (CHGVAR) command is not processed.

```
IF COND((&ALPHA *EQ &BETA) *AND *NOT &GAMMA)
THEN(RETURN)
```

If the value of &ALPHA equals the value of &BETA and if &GAMMA is a logical 0, then return to the program or procedure that called this CL procedure.

```
IF &LOG1 THEN(IF (&A *GT 10) THEN(GOTO X))
ELSE(GOTO Y)
ELSE DO
: (group of CL commands)
ENDDO
```

This is an example of nested IF commands. If &LOG1 has a logical value of 1 (true) and if &A is greater than 10, a branch is made to label X. If &LOG1 has a logical value of 1 and &A is *not* greater than 10, a branch is made to label Y. If &LOG1 has a logical value of 0 (false), &A is not compared to 10. Instead, the DO group of the second ELSE command is processed.

```
IF &TEST THEN(DO)
CHGVAR &A (&A + 1)
GOTO X
ENDDO
ELSE DO
CHGVAR &B (&B + 1)
CALL EXTPGM (&B)
ENDDO
```

This example shows how the THEN parameter can be continued on the next line. If &TEST has a logical value of 1, the Do group specified in the THEN parameter is processed. Otherwise, the Do group specified by the ELSE command is processed.

```
IF (&A *EQ YES) DO
  CHGVAR &B 1
  CHGVAR &C 'Z'
ENDDO
```

This example shows a Do group as the THEN parameter. The two Change Variable (CHGVAR) commands are processed if, in the relational expression, &A is equal to YES.

```
IF %SWITCH(10XXXX10) THEN(GOTO X)
```

This example shows how the built-in function %SWITCH is used to test the eight job switches in a job. Refer to the end of for a complete description of %SWITCH. In this example, job switches 1, 2, 7, and 8 are tested for the values indicated in the 8-character mask. If switches 1 and 7 contain 1s and switches 2 and 8 contain 0s, then control branches to the command having the label X. If any of the four switches do not contain the value indicated, the branch does not occur.

[Top](#)

Error messages

*ESCAPE Messages

CPF0816

%SWITCH mask &1 not valid.

[Top](#)

Install Program Temporary Fix (INSPTF)

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

Parameters
Examples
Error messages

The Install Program Temporary Fix (INSPTF) command allows the user to load and apply PTFs for multiple products with a single command. PTF groups will be copied to the system when they do not already exist on the system or when the level of the PTF group on the media is higher than the level of the PTF group that exists on the system. If LICPGM(*ALL) is specified, any special handling PTFs listed in PTF groups on the media will be used during the install.

The OMIT and HIPER parameters are supplied to allow the user of the INSPTF command to be more selective. These parameters apply only to the PTF loading activity. Any PTF already loaded on the system will be applied.

The INSTYP parameter controls the apply of the PTFs. Using the different special values allows immediate and delayed apply combinations as well as starting an IPL.

INSPTF does not support loading PTFs from tape for products that have multiple releases of the base option installed on the system. If PTFs for such a product exist on the tape, the INSPTF will not load those PTFs and will return an error.

Top

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product description	Values (up to 300 repetitions): <i>Element list</i>	Required, Positional 1
	Element 1: Product	<i>Character value</i> , *ALL	
	Element 2: Release	<i>Character value</i> , *ONLY	
DEV	Device	<i>Name</i> , *SERVICE, *NONE	Optional
INSTYP	PTF apply type	*SRVATT, *DLYIPL, *DLYALL, *IMMONLY, *IMMDLY	Optional
OMIT	PTF omit list	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Product	<i>Character value</i>	
	Element 2: PTF identifier	<i>Character value</i>	
	Element 3: Release	<i>Character value</i> , *ONLY	
HIPER	HIPER PTFs only	*YES, *NO	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
RESTART	Restart type	*IPLA, *SYS, *FULL	Optional
PMTMED	Prompt for media	*SNGVOLSET, *MLTVOLSET, *MLTSRV	Optional

Top

Product description (LICPGM)

Specifies the product ID, version, release, and modification level of the products for which PTFs should be installed.

The possible values are:

***ALL** The available PTFs for all installed products are installed. This must be the first and only value if specified. All values specified after it are ignored.

The possible Licensed Program value is:

licensed-program

Specify the product ID of the PTFs to be installed. A maximum of 300 product IDs can be specified.

The possible Release Level of the Licensed Program values are:

release-level

Specify the release level of the base product option in the format VxRyMz, where Vx is the version number, Ry is the release number, and Mz is the modification level.

***ONLY**

This value is valid only when one release of the product's base option is installed on the system. PTFs for all installed options of the product are loaded and applied regardless of the release-level of the option.

Top

Device (DEV)

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

The possible values are:

***SERVICE**

The PTFs sent from the service support system are installed.

***NONE**

No PTFs are loaded. Any PTF already loaded on the system will be applied. This special value is used after the IPL following an incomplete PTF install. Special handling PTFs in a PTF group being installed must be applied and active before the remaining PTFs in the PTF group can be applied.

tape-device-name

Specify the name of the tape device from which the PTFs are installed.

optical-device-name

Specify the name of the optical device from which the PTFs are installed.

Top

PTF apply type (INSTYP)

Specifies the type of install to perform.

The possible values are:

***SRVATT**

The type of install depends on the service attribute setting.

Attention:

The service attribute is shipped with *DLYIPL as the default. Use the Change Service Attributes (CHGSRVA) command to change the default.

***DLYIPL**

All PTFs are marked for delayed apply and an initial program load (IPL) is done on the system.

***DLYALL**

All PTFs are marked for delayed apply and an initial program load (IPL) is not done on the system.

***IMMDLY**

The immediate PTFs are applied and the delayed PTFs are marked for apply at the next initial program load (IPL).

***IMMONLY**

All PTFs are loaded, but only the immediate PTFs are applied and an initial program load (IPL) is not done on the system.

Top

PTF omit list (OMIT)

Specifies the product ID, version, release, modification level, and PTF ID for PTFs that should not be loaded. The current state of the PTF is not checked before being passed to LODPTF. If the PTF is already loaded it is applied. A maximum of 50 PTFs can be omitted.

The possible Licensed-Program value is:

licensed-program

Specify the product ID for the PTFs that should not be loaded.

The possible PTF-Number values is:

PTF-number

Specify the PTF ID for the PTFs that should not be installed.

The possible Release Level values are:

release-level

Specify the release level of the base product option or the release level of the PTF for the PTFs that should not be loaded. The release level must be specified in VxRyMz format, where Vx is the version number, Ry is the release number, and Mz is the modification level.

***ONLY**

The only release of the product selected on the LICPGM parameter.

Top

HIPER PTFs only (HIPER)

Specifies whether only high-impact pervasive (HIPER) PTFs should be loaded when installing from a media.

Note: This parameter is ignored if DEV(*SERVICE) is specified. This is only valid when installing IBM cumulative/preventive PTF packages.

The possible values are:

***NO** All PTFs, other than those listed in the omit list, should be installed.

***YES** Only HIPER PTFs that are not on the omit list should be installed.

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 only valid 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.

The possible values are:

***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

Restart type (RESTART)

Specifies the point from which the initial program load (IPL) restarts when the PTF apply type (INSTYP) parameter indicates an IPL will be performed.

Note: This is valid only when INSTYP(*DLYIPL) is specified or when INSTYP(*SRVATT) is specified and the PTF install type (PTFINSTYP) service attribute is set to *DLYIPL.

The possible values are:

***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** Specifies that the system determines how much of the system to restart.

The operating system is always restarted. The hardware is restarted only if a PTF that requires a restart is applied. Other hardware functions, such as some configuration changes, that occur during a *FULL IPL are not processed.

*SYS can result in a shorter IPL time than if you specify *FULL.

***FULL** All portions of the system, including the hardware, are restarted.

Top

Prompt for media (PMTMED)

Specifies whether the user will be prompted for additional PTF volume sets and load PTFs from *SERVICE after loading PTFs from a device.

Note: This parameter is ignored if DEV(*SERVICE) or DEV(*NONE) is specified.

The possible values are:

*SNGVOLSET

The user will be prompted to mount each volume in a single volume set when loading the PTFs. If a virtual optical device is specified on the DEV parameter, all mounted PTF volumes will be processed.

*MLTVOLSET

The user will be prompted for volumes in multiple volume sets when loading the PTFs.

*MLTSRV

The user will be prompted for volumes in multiple volume sets when loading the PTFs. After PTFs are loaded from the last volume set, PTFs will be loaded from the service support system (*SERVICE).

Top

Examples

Example 1: Omitting PTFs

```
INSPTF LICPGM((*ALL)) DEV(*SERVICE) INSTYP(*IMMDLY)
        OMIT((5722999 MF12345 V5R3M0) (5722SS1 SI12345 V5R3M0))
```

This command will load all PTFs that are in *SERVICE for all products installed on the system except MF12345 and SI12345. It will then apply all PTFs in loaded status that can be applied immediately and mark the rest for delayed apply.

Example 2: Installing HIPER only

```
INSPTF LICPGM((5722PT1 V5R3M0)) DEV(TAP01) INSTYP(*IMMONLY)
        HIPER(*YES)
```

This command will search the media for PTFs for the V5R3M0 Performance Tools product in the HIPER section. Each PTF that can be applied immediately will be. Delayed PTFs will be loaded, but not marked for apply.

Example 3: Installing Only Immediate PTFs

```
INSPTF LICPGM((*ALL)) DEV(TAP01) INSTYP(*IMMONLY)
        ENDOPT(*LEAVE)
```

This command will load all PTFs for the products that are installed on the system, from the device TAP01. All PTFs in loaded status on the system that can be applied immediately will be. No delayed PTFs will be set for apply.

Top

Error messages

*ESCAPE Messages

CPF358A

Release not valid.

CPF358F

LICPGM parameter contains duplicate entries.

CPF35EB

Multiple releases of product &1 installed.

CPF3586

List of PTFs not correct.

CPF36B7

PTF install processing incomplete; IPL required.

CPF3606

Product &1 &2 not installed.

CPF361A

PTFs installed successfully, but actions pending.

CPF361B

PTF install processing failed, and there are actions pending.

CPF361C

No PTFs installed.

CPF3615

PTF install processing failed.

CPF3618

The mode is not set at Normal.

Top

Initialize Diskette (INZDKT)

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

Parameters
Examples
Error messages

The Initialize Diskette (INZDKT) command initializes a diskette for use. This command writes identification information on a diskette and sets the format that will be used on it. Initializing a diskette includes the following:

- Checking for files that are still active and should not be cleared.
- Testing each track for physical defects on the recording surface. A diskette is unusable if more than two defective cylinders are found, if cylinder 0 is defective, or if the track identifier of a defective track cannot be read.
- Formatting each track to a specified sector size (128, 256, 512, or 1024 bytes) and recording density (single density or double density). A diskette's format determines what the diskette may be used for in later processing. This is explained more fully in the FMT and SCTSIZ parameters.
- Defining a single (expired) file covering the entire diskette. The file is identified as DATA.

IBM-supplied diskettes are initialized before they are shipped to a customer. A diskette should be reinitialized when:

- Its format is changed.
- The sequence of the records on the diskette is changed (they can only be sequential).
- A defect has occurred in one or two tracks.
- The diskette has been exposed to a strong magnetic field.

One INZDKT command can initialize only one diskette at a time.

Note: When initializing diskettes with labels that are not IBM standard labels, specify CHECK(*NO).

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Diskette device	<i>Name</i>	Required, Positional 1
NEWVOL	New volume identifier	<i>Character value</i> , *NONE	Optional, Positional 2
NEWOWNID	New owner identifier	<i>Character value</i> , *BLANK	Optional, Positional 3
FMT	Diskette format	*DATA, 1, 2, 2D, *DATA2, *SAVRST	Optional, Positional 4
SCTSIZ	Sector size	*STD, 128, 256, 512, 1024	Optional, Positional 5
CHECK	Check for active files	*YES, *NO	Optional
CODE	Code	*EBCDIC, *ASCII	Optional

Top

Diskette device (DEV)

Specifies the name of the device where the diskette being initialized is placed.

This is a required parameter.

Top

New volume identifier (NEWVOL)

Specifies the volume identifier for the diskette being initialized.

The possible values are:

*NONE

No volume identifier is specified; only the system date is written in the volume identifier field of the volume label.

volume-identifier

Specify a maximum of 6 characters for the volume identifier to identify the diskette being initialized; any combination of letters and numbers can be used.

Top

New owner identifier (NEWOWNID)

Specifies the identification of the diskette owner to write in the volume label. Any combination of alphabetic characters or numeric characters can be used.

The possible values are:

*BLANK

Blanks are written in the owner identification field.

owner-identifier

Specify a maximum of 14 uppercase letters and numbers that identify the owner of the diskette. Even if the value is enclosed in apostrophes, no lowercase letters, embedded blanks, or special characters can be used. If fewer than 14 characters are specified, the field is left-justified and padded on the right with blanks.

Top

Diskette format (FMT)

Specifies the format to use to initialize the diskette. Either *DATA, *DATA2, 1, 2, or 2D can be specified if the diskette will contain data files that are in the basic, H, I, or S/36-E exchange formats. The basic exchange format has a set of requirements that ensures that a diskette can be exchanged between systems that are capable of using both the IBM diskettes types 1 or 2. *SAVRST (type E general data exchange) must be specified if the diskette is used in the AS/400 system or System/38 save and restore operations; that is, if their data files will contain saved objects. *SAVRST is also a valid format for the System/36 save and restore operations, and it is the preferred format to use.

The possible values are:

*DATA

A one-sided or two-sided diskette is formatted with single-density recording.

- 1 A one-sided diskette is formatted with single-density recording.
- 2 A two-sided diskette is formatted with single-density recording.
- 2D A two-sided diskette is formatted with double-density recording.

***DATA2**

A two-sided diskette is formatted with double-density recording.

***SAVRST**

A two-sided diskette is formatted with double-density recording. This format is required if the diskette is used in a save or restore operation.

Note: Because *DATA2, *SAVRST, and 2D specify that the diskette is used for double-density recording, it is recommended that a type 2D diskette be used, rather than a type 2. A type 2D diskette is made for double-density recording, whereas a type 2 is made for single-density recording and is more prone to media errors if used for double-density recording.

Note: If (FMT(*SAVRST) is specified, CODE(*ASCII) cannot be specified.

Top

Sector size (SCTSIZ)

Specifies the number of bytes per sector with which each track is initialized.

The possible values are:

***STD** A standard sector size, based on the value of the FMT parameter, is used.

	SCTSIZ
FMT (*STD)	
*DATA	128
1	128
2	128
2D	256
*DATA2	256
*SAVRST	1024

- 128 Each track is initialized with 128 bytes per sector.
- 256 Each track is initialized with 256 bytes per sector.
- 512 Each track is initialized with 512 bytes per sector.
- 1024 Each track is initialized with 1024 bytes per sector.

Top

Check for active files (CHECK)

Specifies whether a check for active files (files having an expiration date later than the system date) is made.

The possible values are:

- *YES** A check is performed on files that have labels or that are in cylinder 0 only. File labels in an extended file label area are not checked. If any active files are found, an operator message is sent. The operator can continue initialization (in which case active files are destroyed) or end the initialization of that diskette.
- *NO** Diskette initialization proceeds without a check for active files being made.

Top

Code (CODE)

Specifies the code in which the volume label is written. All data subsequently written must be in the same code; codes cannot be intermixed on a diskette.

The possible values are:

- *EBCDIC** The volume label is written in EBCDIC and is an IBM standard label; all subsequent data must also be written in EBCDIC.
- *ASCII** The volume label is written in ASCII and is an IBM standard label in the same format as the EBCDIC label; all subsequent data must also be written in ASCII. If FMT(*SAVRST) is specified, *ASCII cannot be specified.

Top

Examples

Example 1: Initializing a Diskette

```
INZDKT  DEV(DKT1)  NEWVOL(ORIGIN)  NEWOWNID(DEPT504)
```

This command initializes the diskette device DKT1. The diskette is checked for active files (CHECK(*YES) is assumed). The diskette is formatted for basic data exchange files (FMT(*DATA) is assumed). The volume label has ORIGIN written in the volume identifier field and DEPT504 written in the owner identifier field.

Example 2: Initializing a Diskette to the Save and Restore Format

```
INZDKT  DEV(DKT2)  NEWVOL(SAVE)  NEWOWNID(DON)
        FMT(*SAVRST)  CHECK(*NO)
```

This command initializes the diskette in device DKT2 to the save and restore format. The diskette is initialized with the NEWVOL parameter of SAVE. The owner identifier field has DON written into it.

Top

Error messages

*ESCAPE Messages

- CPF6156**
Cancel reply received for message &6.
- CPF6716**
Device &1 not a diskette device.
- CPF6717**
Initialize diskette ended; previous errors occurred.
- CPF6718**
Cannot allocate device &1.
- CPF6757**
Owner identifier &1 contains wrong characters.
- CPF6758**
Volume identifier &1 contains wrong characters.
- CPF6779**
Format (FMT) specified for diskette in device &1 not valid.
- CPF9814**
Device &1 not found.
- CPF9825**
Not authorized to device &1.

Top

Initialize DLFM (INZDLFM)

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

Parameters
Examples
Error messages

The Initialize DLFM (INZDLFM) command prepares the DataLink File Manager (DLFM) to be started, and clears information from the database files used by the DLFM.

Restrictions:

- To use this command, you must have input/output system configuration (*IOSYSCFG) special authority.

Top

Parameters

Keyword	Description	Choices	Notes
CLEARDB	Clear existing databases	<u>*LNKSTS</u> , *ALL	Optional, Positional 1

Top

Clear existing databases (CLEARDB)

Specifies which databases should be cleared.

*LNKSTS

The database files containing link status of DataLinks will be cleared. Database files containing registered prefixes and host database names will not be cleared.

***ALL** All database files used by the DataLink File Manager (DLFM) will be cleared.

Top

Examples

Initializing and Clearing a DataLink File Manager

```
INZDLFM CLEARDB(*ALL)
```

This command initializes the DataLink File Manager, and clears all database files of existing data.

Top

Error messages

*ESCAPE Messages

CPF3168

DataLink File Manager (DLFM) command failed.

Initialize Distribution Queue (INZDSTQ)

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

Parameters
Examples
Error messages

The Initialize Distribution Queue (INZDSTQ) command resets the status of a distribution queue and the entries on the queue. It also optionally clears all distributions on the queue. This command applies to both the normal and high priority sections of the specified queue.

Attention: Initializing a distribution queue can result in the loss or duplication of distributions in the network, depending on the status of the distributions in transit at the time this command is run.

Initializing a distribution queue includes the following:

- If a SNADS (Systems Network Architecture (SNA) distribution services) sender job is active for the queue, the job is ended. This job cancellation takes effect immediately. Distribution queues being sent are interrupted.
- If the queue type is a SystemView distribution services (SVDS) queue type and a receiver job is active for this connection, the job is ended. This job cancellation takes effect immediately. All partially received distributions are discarded.
- If the distribution queue is to be cleared, all distributions on the queue are deleted as specified on the **Clear queue entries** prompt (CLEAR parameter).
- If the queue is not cleared, the distributions on the queue that do not have "Held" status are set to "Ready." Distributions with a status of "Held" remain held.
- The queue status is set to "Ready" unless the queue is in the "Held" status.
- If the QSNADS system is active, a SNADS sender job is submitted for the queue following the same rules used to start the QSNADS subsystem.

Distribution queue names are translated to the graphic character set and code page 930 500, using the job's coded character set identifier (CCSID).

Restrictions:

- This command is shipped with public *EXCLUDE authority, and the QPGMR and QSYSOPR user profiles have private authorities to use the command.
- Messages that report errors about distribution queues may display or print different characters than you entered for the distribution queue name because of internal system transformations. Similarly (depending on the language used for the work station), the internal value for a distribution queue name may differ from the characters shown for the Work with Distribution Queue (WRKDSTQ) command. An error may be reported if the character-string value specified for the **Distribution queue** prompt (DSTQ parameter) does not match the rules for an internal distribution queue value or if it does not match the internal value for any defined distribution queue (ignoring case differences).

Top

Parameters

Keyword	Description	Choices	Notes
DSTQ	Distribution queue	<i>Character value</i>	Required, Positional 1
CLEAR	Clear queue entries	*NO, *YES, *PURGE	Optional

Distribution queue (DSTQ)

Specifies the name of the distribution queue to initialize. The queue must be previously configured using the Configure Distribution Services (CFGDSTSRV) or the Add Distribution Queue (ADDDSTQ) command.

This is a required parameter.

Clear queue entries (CLEAR)

Specifies whether distributions on the queue are deleted.

Attention: Using the *PURGE value results in the loss of distributions with no trace.

The possible values are:

***NO** Distributions on the queue are not deleted.

***YES** Distributions on the queue are deleted. Each deleted distribution is logged and, if the distribution originator requested notification, a notification is sent to the originator or to the report destination specified in the distribution.

Note: System Network Architecture distribution services (SNADS) status distributions and distribution reports are used to report information about a distribution back to the originator. Status report distributions never result in another status report distribution. If a status report distribution is deleted, no notification is sent.

***PURGE**

Distributions on the queue are deleted. Deleted distributions are not logged and no notification is sent to the originator or to the report destination specified in the distribution.

Examples

Example 1: Initializing a Distribution Queue

```
INZDSTQ DSTQ('SYSTEMA APPN')
```

Connection information is about to be changed for system 'SYSTEMA APPN' by a central site administrator. This command initializes the queue to avoid error conditions that can be encountered by the Change Distribution Queue (CHGDSTQ) command. Distributions on the queue are not deleted.

Example 2: Initializing and Clearing a Distribution Queue

```
INZDSTQ DSTQ('ERRORQ') CLEAR(*YES)
```

This command clears the distribution queue ERRORQ that is being used as a repository for distributions that would have resulted in routing errors. Distributions that are deleted are logged, and the originators of the distributions are notified.

Example 3: Initializing and Purging a Distribution Queue

```
INZDSTQ DSTQ('TESTQ') CLEAR(*PURGE)
```

This command clears the distribution queue TESTQ that is being used for testing a new batch application. Distributions are deleted but not logged, and the originators are not notified.

[Top](#)

Error messages

*ESCAPE Messages

CPF8802

Distribution queue &1 was not found.

CPF8807

Error occurred while using QSNADS journal.

CPF8809

Errors detected on SNADS internal queues.

CPF8812

Error occurred while processing distribution queues.

CPF8849

Queue &1 in use by another distribution services function.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

[Top](#)

Initialize Optical (INZOPT)

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

Parameters
Examples
Error messages

The Initialize Optical (INZOPT) command initializes an optical volume. Depending on the type of optical volume being initialized this operation may take up to 30 minutes to complete. When an existing optical volume is initialized a second time, all existing information is lost.

Restriction: To use this command you must have *ALL authority to the authorization list securing the volume if it is in an optical media library device. You need *CHANGE authority to the authorization list securing the volume if it is in an optical device.

Top

Parameters

Keyword	Description	Choices	Notes
VOL	Volume identifier	Character value, <u>*MOUNTED</u>	Optional, Key, Positional 1
NEWVOL	New volume identifier	Character value, <u>*VOL</u>	Optional, Positional 2
DEV	Device	Name	Optional
THRESHOLD	Volume full threshold	1-100, <u>*CALC</u>	Optional
CHECK	Check for an active volume	*NO, <u>*YES</u>	Optional
ENDOPT	End of media option	<u>*LEAVE</u> , *UNLOAD	Optional
CLEAR	Clear	<u>*NO</u> , *YES	Optional
TEXT	Text 'description'	Character value, <u>*BLANK</u>	Optional
TYPE	Volume type	<u>*PRIMARY</u> , *BACKUP	Optional
CCSID	Coded character set ID	<u>*CALC</u> , 500, 850	Optional
MEDFMT	Media format	<u>*MEDTYPE</u> , *HPOFS, *UDF	Optional

Top

Volume identifier (VOL)

Specifies the volume identifier of the optical volume being initialized.

*MOUNTED

The volume mounted in the specified device (DEV parameter) will be initialized.

volume-identifier

Specify the identifier of the optical volume to initialize.

Top

New volume identifier (NEWVOL)

Specifies the identifier of the optical volume after it is initialized. The identifier must contain only alphabetic characters (A through Z), numeric characters (0 through 9), hyphen (-), underscore (_), or a period (.). The first character must be alphabetic or numeric and the identifier cannot contain blanks.

***VOL** The new volume identifier is the same as the old volume identifier.

new-volume-identifier

Specify the new volume identifier.

Top

Device (DEV)

Specifies the optical device which contains the volume to be initialized. This parameter is only required when VOL(*MOUNTED) is specified. The device cannot be an optical media library device.

optical-device

The name of the optical device containing the volume which will be initialized.

Top

Volume full threshold (THRESHOLD)

Specifies the percentage of space on the volume to use until the volume is considered full. This field is only used if the media format is *HPOFS. For other media formats, this field is ignored and the threshold will default to 100 percent.

***CALC**

The system will calculate the percentage of the volume to use based on media format and volume type.

- For a media format of *HPOFS and a volume type of *PRIMARY the threshold will be 95 percent.
- For a media format of *HPOFS and a volume type of *BACKUP the threshold will be 99 percent.
- For a media format of *UDF the threshold will be 100 percent.

volume-full-threshold

Specify the volume threshold percentage. Valid values range from 1 through 100.

Note: If the volume type is *BACKUP, this parameter is ignored and the volume-full-threshold is set to 99 percent.

If the media format is *UDF, this parameter is ignored and the volume-full-threshold is set to 100 percent.

Top

Check for an active volume (CHECK)

Specifies whether the system checks to see if the optical volume is initialized.

***YES** The system checks to see if the optical volume is initialized. If the volume is initialized, the operation is ended and an error message is sent.

***NO** The system does not check to see if the optical volume is initialized. The volume will be initialized and all existing data will be lost.

End of media option (ENDOPT)

Specifies whether the media is unloaded from the device after the initialize command completes.

Note: This parameter is ignored if the media is an optical library device.

*LEAVE

When the initialize completes the media is left in the device.

*UNLOAD

When the initialize completes the media is unloaded from the device.

Top

Clear (CLEAR)

Specifies whether or not existing data on the volume will be cleared during the initialize process. This parameter only applies when the volume media type is *DVD-RAM.

Note: If the volume media type is *WORM the volume is never cleared regardless of the parameter setting.

If the volume media type is *ERASE the volume is always cleared regardless of the parameter setting.

*NO The volume is not cleared.

*YES The volume is cleared of existing data prior to initialization.

Note: If this option is selected the INZOPT command may take several hours or more to complete, depending on the media capacity.

Top

Volume type (TEXT)

Specifies the text that briefly describes the optical volume.

*BLANK

Text is not specified.

'description'

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

Top

Volume type (TYPE)

Specifies the type of optical volume being initialized. Optical volumes for user applications are initialized as primary volumes. Backup optical volumes can be written to only by using the following set of optical backup commands: CVTOPTBKU, CPYOPT, and DUPOPT.

*PRIMARY

The optical volume is used as a primary volume.

*BACKUP

The optical volume is used as a backup volume.

Coded character set ID (CCSID)

Specifies the character set in which the optical volume, directory, file names, and volume description are written. This parameter does not affect how user data is written. The user application must determine the character set in which the file data is written.

*CALC

The system will select the default character set based on the media format.

500 The EBCDIC character set and code page 500 are used.

850 The ASCII character set and code page 850 are used.

Top

Media format (MEDFMT)

Indicates the media format to use when writing to the optical media. There are two media formats, either *HPOFS (High Performance Optical File System) or *UDF (Universal Disk Format). For a complete comparison of the two media formats refer to the Optical Support, SC41-4310 book.

*MEDTYPE

Specifies that the operating system will determine which media format is used to initialize the volume. The media format will be based upon the media type.

- If the media type is *WORM or *UNKNOWN, the media will be initialized using the *HPOFS format.
- If the media type is *ERASE and has not been previously initialized the media will be initialized using the *HPOFS format.
- If the media type is *ERASE and has been previously initialized it will be initialized using the previous media format.
- If the media type is *DVD-RAM, the media will be initialized using the *UDF format.

***HPOFS**

The High Performance Optical File System is used to initialize the volume. One of the characteristics of HPOFS is space occupied by a deleted file is not reused. The only way deleted file space can be recovered is to reinitialize the media thereby losing all previously recorded data on the media.

***UDF** The Universal Disk Format, a subset of the ISO 13346 standard, is used to initialize the volume. One of the characteristics of UDF is space occupied by a deleted file will be reused when needed for either the creation of a new file or the extension of an existing file. The UDF media format also provides file and directory level security through the use of permissions.

Top

Examples

```
INZOPT VOL(VOL01) THRESHOLD(99) CHECK(*N0)
```

This command initializes the optical volume VOL01 with a volume-full-threshold of 99 percent. The system does not check to see if the volume is initialized.

Top

Error messages

*ESCAPE Messages

OPT1305

Optical volume &1 is read only.

OPT1315

Optical volume &1 is write protected.

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.

OPT1335

Volume &1 already initialized.

OPT1342

Invalid volume identifier specified.

OPT1345

No free space available on media.

OPT1346

Operation not allowed to volume located in a remote optical device.

OPT1350

Write operation failed to optical volume &1.

OPT1360

Media directory corrupted on optical volume &1.

OPT1375

Optical volume &1 already exists.

OPT1460

Optical volume &1 is not in an optical device.

OPT1485

Initialize or rename of optical volume failed.

OPT1489

Volume parameter is not permitted for device &1.

OPT1530

&1 does not represent a valid optical device.

OPT1540

Invalid parameters specified.

OPT1555

Optical device &1 in use.

OPT1605

Media or device error occurred.

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.
- OPT2301**
Internal system object in use.
- OPT2420**
Not authorized to optical volume &2.
- OPT2422**
Not authorized to file or directory.
- OPT7740**
User not authorized to object &2 in library &3 type &4.

Top

Initialize Client Access/400 (INZPCS)

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

Parameters
Examples
Error messages

The Initialize Client Access/400 (INZPCS) command allows you to establish an operating environment for Client Access applications by creating various control documents on the Client 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 Client Access/400 (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, JEI, 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, 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. For additional information about what code page should be used, you can refer to the appendix in the Client Access/400 Installation and Administration Guide called "National Language Support for Client Access/400".

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. For additional information about what code page should be used, you can refer to the appendix in the Client Access/400 Installation and Administration Guide called "National Language Support for Client Access/400".

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 Client 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
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

Error messages

*ESCAPE Messages

IWS16D0

Initialize Client Access/400 (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.

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 Install, upgrade, or delete OS/400 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.

The possible values are:

*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.

volume-identifier

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.

The possible values are:

*BLANK

Text is not specified.

owner-identifier

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, then the volume specified should be the cartridge identifier to be mounted and used.

The possible values are:

*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.

volume-identifier

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.

The possible values are:

***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.

The possible values are:

***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

3590 *FMT3590

3590-Exx
*FMT3590E

3590-Hxx
*FMT3590H

4685-001
*VXA2

6335 *QIC3040

6343 *QIC1000

6344 *QIC2GB

6348 *QIC1000

6349 *QIC2GB

6368 *QIC1000

6369 *QIC2GB

6379 *QIC1000

6380 *QIC2GB

6381 *QIC2DC

6382 *QIC4DC

6383 *QIC5010

6384 *SLR60

6385 *QIC5010

6386 *MLR3

6387 *SLR100

6390 *FMT7GB

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.

tape-density

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.
- *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.
- *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.

***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.

Note: Self-configured tape devices may define additional valid values for the density parameter. Use iSeries 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.

The possible values are:

*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.

The possible values are:

*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.

The possible values are:

*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.

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.

Top

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 CL procedures.
- 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](#)

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 CL procedures.
- 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 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

Iterates on 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.

Top

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.

[Top](#)

Error messages

None

[Top](#)

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 optical 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 files that are in mounted or loaded status will be loaded in the specified virtual optical 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 images are removed from the specified virtual optical device. The allow save attribute will be set to allow save for all image catalog files.

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

Restrictions:

- You must have security administrator (*SECADM) and all object (*ALLOBJ) special authorities to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
DEV	Virtual optical device	<i>Name</i>	Required, Positional 2
OPTION	Option	*LOAD , *UNLOAD	Optional, Positional 3

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

Virtual optical 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 CD or DVD device.

Top

Option (OPTION)

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

*LOAD

The image catalog will be loaded into the virtual optical device.

*UNLOAD

The image catalog will be unloaded from the virtual optical 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.

Example 2: Unloading an Image Catalog

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

This command unloads image catalog MYCLG from device OPTVRT01.

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.

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.

*MOUNTED

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(*MOUNTED) is specified, an existing entry in mounted status will be changed to a loaded status.

Restrictions:

- You must have security administrator (*SECADM) and all object (*ALLOBJ) special authorities to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Required, Positional 1
IMGCLGIDX	Image catalog index	1-256, <u>*FIRST</u> , *LAST	Optional, Positional 2
OPTION	Option	<u>*MOUNT</u> , *LOAD, *UNLOAD	Optional, Positional 3

Top

Image catalog (IMGCLG)

Specifies the image catalog to be used.

This is a required parameter.

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.

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

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.

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.

Top

Load Program Temporary Fix (LODPTF)

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

Parameters
Examples
Error messages

The Load Program Temporary Fix (LODPTF) command loads program temporary fixes (PTFs) for a specified product from a tape, diskette, 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. Not all of the PTFs for the specified program must be loaded, because specific PTFs can be selected or omitted and loaded.

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

Top

Parameters

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

device-name

Specify the name of the tape, diskette, 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** prompt (OMIT parameter) cannot be specified if single PTF numbers are specified on the SELECT parameter.

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

You can enter multiple values for this parameter.

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

PTF-number

Specify the PTF identification numbers of the single PTFs that are loaded. A maximum of 50 PTF numbers can be specified.

Top

PTF numbers to omit (OMIT)

Specifies that all PTFs except for those specified in this parameter are loaded. Specify the PTF numbers that you want omitted (not loaded) when the rest are loaded. A maximum of 50 PTF numbers can be specified. The OMIT parameter cannot be specified if single PTF numbers are specified on the **PTF numbers to select** prompt (SELECT parameter).

You can enter multiple values for this parameter.

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.

The possible values are:

*ONLY

This value is only valid 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.

release-level

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** prompt (DEV parameter).

*SEARCH

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

sequence-number

Specify the sequence number of the PTF file being loaded. This sequence number must exist on the tape. Valid values range from 1 through 16777215.

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 only valid 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.

The possible values are:

*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.

The possible values are:

***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.

path-identifier

Specify the path identifier of the existing PTF file from which to load the PTF data.

Top

Save file (SAVF)

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

***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.

library-name

Specify the library where the save file is located.

The possible save-file values are:

save-file-name

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

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 device name or an optical device name is specified on the **Device** prompt (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.

Top

Examples

Example 1: Omitting PTFs

```
LODPTF LICPGM(5722SS1) OMIT(SI00003 SI00008 SI00014)
```

This command loads all of the PTFs from the service support system (*SERVICE) for the product 5722SS1 except SI00003, SI00008, and SI00014. The Apply Program Temporary Fix (APYPTF) command can then be used to apply these PTFs to the 5722SS1 product.

Example 2: Selecting PTFs

```
LODPTF LICPGM(5722SS1) 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 5722SS1 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.

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 Operation information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 AS/400 or System/36 naming conventions for devices:

- If QDEVNAMING is *NORMAL (AS/400 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 LODRUN command requires *SAVSYS authority. The QINSTAPP program that is loaded from the media and called may require additional authority in order to run correctly. The user supplying the QINSTAPP program should inform you if any additional authorities are required.

Top

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>l</u>	Optional

Top

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.

Top

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.

Top

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>.

Restrictions:

1. 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.
2. 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.
3. The user must have execute (*X) authority to each directory in the path.
4. When creating a directory in the "root" (/), QOpenSys or user_defined file system, the user must have write, execute (*WX) authority to the directory that contains the new directory.
5. 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.
6. 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.

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

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.

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, execute (*RWX) authority provides object operational (*OBJOPR) and all data authorities.

***RW** The user can view and change the contents of an object. Read, 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, 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, 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.

authorization-list-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 system auditing value (QCRTOBJAUD).

***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 turn on 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 System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Restrictions:

1. 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.
2. 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.
3. The user must have execute (*X) authority to each directory in the path.
4. When creating a directory in the "root" (/), QOpenSys or user_defined file system, the user must have write, execute (*WX) authority to the directory that contains the new directory.
5. 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.
6. 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.

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

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.

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, execute (*RWX) authority provides object operational (*OBJOPR) and all data authorities.

***RW** The user can view and change the contents of an object. Read, 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, 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, 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.

authorization-list-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 system auditing value (QCRTOBJAUD).

***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 turn on 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 System API Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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)

Parameters
Examples
Error messages

Threadsafe: Yes

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

Restrictions:

- This command is valid only in CL procedures.

- 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 CL procedures, except for the following: DO, DOWHILE, DOUNTIL, DOFOR, ELSE, ENDDO, SELECT, WHEN, OTHERWISE, ENDSELECT, ENDPGM, 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.

Top

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.

Top

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.

Top

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, a remote file system accessed through local Network File System client (*NFS), or a local or remote NetWare file system (*NETWARE). 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 OS/400 Network File System book, SC41-5714

Restrictions:

1. The user must have input/output (I/O) system configuration (*IOSYSCFG) special authority to use this command.
2. If the user is mounting a NetWare file system, the user must have execute (*EXECUTE) authority to the file system to be mounted.
3. The user must have write (*W) authority to the directory to be mounted over.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS, *NETWARE	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.

*NETWARE

The file system specified for the MFS parameter is a NetWare file system. The MFS parameter must be one of the following forms:

- *server/volume:pathname*, where *pathname* is optional.
- NetWare Directory Services (NDS) context to a volume, a directory map object to mount, or an alias to a volume or directory map object. The NDS context can be a distinguished or relative context. If a relative context is specified the current context for the job is searched, and if it is not found the default system context is searched. If a context to a volume or an alias to a volume is specified an optional directory path may also be specified.

Note: On the MFS parameter, if a relative context is specified that contains no dots and no path name after the colon, the user must be sure to quote the parameter value when prompting on the command. The command analyzer may interpret the MFS value as a label and remove the trailing colon.

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), a remote NFS path name, or the path of a NetWare file system. See the **Type of file system (TYPE)** parameter to determine the correct format for the MFS parameter.

This is a required parameter.

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.

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,rsize=8096,wsiz=8096,timeo=20,retrans=5,acregmin=30,acregmax=60,acdirmin=30,acdirmax=60,hard'
```

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

```
'rw,suid'
```

The default value for the options string for the mount of a NetWare file system (*NETWARE) is:

```
'rw,acregmax=60,acdirmax=60'
```

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. For the mount of a NetWare file system, only the *ro*, *rw*, *acregmax*, *acdirmax*, *noac*, and *nocto* 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.

rsiz=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 8096. If *rsiz* is not specified, the default value of 8096 is assumed. For better performance, the read buffer should be a multiple of the the application buffer size.

wsiz=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 8096. If *wsiz* is not specified, the default value of 8096 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 or a NetWare 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 or a NetWare 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 or a NetWare file system, specifies whether to force the refresh of remote attributes when opening a file. If this option is present, 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 present, the default value of no suppression is assumed.

noac For the mount of a Network File System or a NetWare file system, specifies whether to suppress local storage of attributes and names. If this option is present, local storage of attributes and names is suppressed. If *noac* is not present, 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.

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.

Example 4: Mounting a NetWare File System with OPTIONS

```
ADDMFS TYPE(*NETWARE) MFS('RCHNWSVR1/LOTUS:LOTSUITE/SMARTCTR')  
MNTOVRDIR('/temp1') OPTIONS('ro,agregmax=120')
```

This command mounts the NetWare directory *LOTSUITE/SMARTCTR* contained in the volume *LOTUS* that resides on server *RCHNWSVR1* over the directory */temp1*. In addition it specifies to mount as read-only, sets the maximum time to store file attributes locally to 120 seconds.

Example 5: Mounting using a NetWare Directory Services Context

Following are several examples of mounting a NetWare file system using NetWare Directory Services (NDS) contexts.

```
ADDMFS TYPE(*NETWARE) MFS(' .LOTUS_VOL.ROCHESTER.IBM')
      MNTOVRDIR('/temp1')
```

This command mounts NDS volume *LOTUS_VOL* using a distinguished context, over the directory */temp1*.

```
ADDMFS TYPE(*NETWARE)
      MFS('CN=LOTUS_VOL.OU=ROCHESTER:LOTSUITE/SMARTCTR')
      MNTOVRDIR('/temp1')
```

This command mounts path *LOTSUITE/SMARTCTR* on NDS volume *LOTUS* using a relative path and fully qualified names, over the directory */temp1*.

```
ADDMFS TYPE(*NETWARE) MFS(' .CN=LOTUSMAP.OU=ROCHESTER.O=IBM')
      MNTOVRDIR('/temp1')
```

This command mounts a directory map object using a distinguished context and fully qualified names, over the directory */temp1*.

Top

Error messages

*ESCAPE Messages

CPFA0A9

Object not found. Object is &1.

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Restrictions:

1. 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).
2. Only objects that are a byte stream file type move between file systems.
3. A directory cannot be moved to a subordinate directory.
4. Database file members cannot be moved.
5. Objects in QDLS can not be moved between auxiliary storage pools (ASPs).
6. 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.
7. 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 iSeries 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.

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

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

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.

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

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

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.

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

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

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.

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.

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

&1 objects moved. &2 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:

1. You must be enrolled in the System Directory and have *ALL authority to the document being moved and have *CHANGE authority to both the FROM and TO folders (if applicable).
2. To move a document to or from a folder, the user must have *CHANGE authority to the folder.
3. 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 FROMFLR parameter. If *SYSOBJNAM is specified on the FROMDOC parameter, then a system object name must be specified on the SYSOBJNAM 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. When *SYSOBJNAM is specified on the FROMDOC parameter, a user-defined variable must be specified on the SYSOBJNAM parameter and *document-name* must be specified.

document-name

Specify the name of the document to be moved.

Note: If FROMDOC(document-name) is specified, then FROMFLR(folder-name) is required. If FROMDOC(*SYSOBJNAM) is specified, then SYSOBJNAM(system-object-name) and RENAME(document-name) are required.

This is a required parameter.

Top

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 **Name, *SYSOBJNAM (FROMDOC)** parameter. FROMFLR(*NONE) cannot be specified if FROMDOC(document-name) is specified.

*NONE

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

folder-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.

folder-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.

If the user specifies *SYSOBJNAM on the FROMDOC parameter, then a document name must be specified on the RENAME parameter.

*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.

document-name

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

Note: If you specify FROMDOC(*SYSOBJNAM) then RENAME(document-name) must be specified.

Top

System object name (SYSOBJNAM)

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

Top

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.

Top

Error messages

*ESCAPE Messages

CPF8A13

Document &2 in folder &1 not moved.

Top

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Restrictions:

1. 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).
2. Only objects that are a byte stream file type move between file systems.
3. A directory cannot be moved to a subordinate directory.
4. Database file members cannot be moved.
5. Objects in QDLS can not be moved between auxiliary storage pools (ASPs).
6. 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.
7. 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 iSeries 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.

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

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

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.

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

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

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.

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

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

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.

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'.

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.

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

&1 objects moved. &2 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.

Restrictions:

1. You must have object management (*OBJMGT) authority for the object to be moved, delete (*DELETE) and execute (*EXECUTE) authorities for the library the object is currently in, and add (*ADD) and execute (*EXECUTE) authorities for the library to which the object is to be moved.
2. The following object types cannot be moved: Libraries, user profiles, edit descriptions, line descriptions, control unit descriptions, device descriptions, journals, and journal receivers.
3. The following objects cannot be moved: the system operator message queue QSYSOPR, all work station user message queues, and the system log QHST.
4. 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.
5. The library to which the object is to be moved cannot be QTEMP.
6. 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.
7. 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 (QHSMMOVL) 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.
 - 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.
- The object will not be journaled even if the original object was journaled.
- The private authorities for the objects will be preserved.

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 concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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.

***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.

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.

```
MOVOBJ 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)
 Threadsafte: 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 (MRGTCPTH)

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

Parameters
Examples
Error messages

The Merge TCP/IP Host Table (MRGTCPTH) 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 either *iSeries, *AIX, or *NIC file format to be merged with the local host table. The local host table is located in member QUSRSYS/QATOCHOST.HOSTS and is created as a physical file.

A maximum of 4 host names per IP address is allowed when host tables are merged. For example: If the local host table already has 3 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 original copy of the local host table is not saved by the Merge TCP/IP Host Table (MRGTCPTH) command. To save the original host table, create a copy of the file QUSRSYS/QATOCHOST.HOSTS using the Copy File (CPYF) command. Do this before issuing the MRGTCPTH 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, *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.

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 iSeries format.

Note: *AS400 can only be used if the physical file member specified is a host table from an iSeries running Version 3, Release 1, Modification 0 (V3R1M0) or later of OS/400. If you import a host table from an iSeries running any version of OS/400 prior to V3R1M0, specify *AIX.

***AIX** The physical file member to be merged with the local host table is *AIX format.

***NIC** The physical file member to be merged with the local host table is *NIC format.

Top

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.

Top

Examples

Example 1: Replacing Local Host Table

```
MRGTCPTH FROMFILE(AS400FILE) REPLACE(*YES)
```

This command replaces the contents of QUSRSYS/QATOCHOST.HOSTS with the contents of the first member of physical file AS400FILE. The first member of physical file AS400FILE is in AS/400 host table format.

Example 2: Merging Local Host Table

```
MRGTCPHT 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.

Top

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.

Top

Work with TCP/IP Network Sts (NETSTAT)

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

Parameters
Examples
Error messages

Use the Work with TCP/IP Network Status (WRKTCPSTS) command, also known as NETSTAT, 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.

To use this command, either the TCP/IP protocol stack or IP over SNA must be active. If neither is active, Netstat issues an escape message.

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	Option	<u>*SELECT</u> , *IFC, *RTE, *CNN	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 TCP/IP Interface Status list.

***RTE** Display the Display TCP/IP Route Information list.

***CNN** Display the Work with TCP/IP Connection Status list.

Top

Examples

Example 1: Displaying the Work with TCP/IP Network Status Menu

```
WRKTCPSTS
-or-
WRKTCPSTS  OPTION(*SELECT)
```

Either of these commands will display the Work with TCP/IP Network Status menu.

Example 2: Using the OPTION Parameter

WRKTCPSTS OPTION(*CNN)

This command displays the Work with TCP/IP Connection Status panel.

Example 3: Using a Positional Parameter

WRKTCPSTS *RTE

The OPTION parameter is a positional parameter. The OPTION keyword is not required. This command starts NETSTAT, and the Display TCP/IP Route Information 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 DNS Query (NSLOOKUP)

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

Parameters
Examples
Error messages

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

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

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

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

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

There are two parameters for this command:

1. HOSTNAME
2. DMNNAMSVR

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

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

NAME

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

NAME1 NAME2

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

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

help (or ?)

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

server NAME

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

lserver NAME

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

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

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

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

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

set debug
Show debugging information.

set nodebug
Do not show debugging information.

set d2 Show exhaustive (verbose) debugging information.

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

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

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

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

set nosearch
Do not use the srchlist option.

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

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

set vc Use TCP for queries instead of UDP.

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

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

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

set domain=NAME

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

set srchlist=N1/N2/N3...

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

set root=NAME

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

set retry=X

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

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

set timeout=X

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

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

set type=X

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

A IP Address record. This is the default value.

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

CNAME

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

HINFO

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

MX Mail Exchange record.

NS Name server (DNS server) information for the zone

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

SOA Start of Authority record.

TXT Text record.

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

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

set port=X

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

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

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

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

ls DOMAIN > FILE

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

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

ls -a DOMAIN

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

ls -h DOMAIN

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

ls -s DOMAIN

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

ls -d DOMAIN

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

ls -t TYPE DOMAIN

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

view FILE

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

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

Top

Parameters

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

Top

Top

Domain Name Server (DMNNSVR)

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

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

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

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

Valid values for the DMNNSVR parameter are:

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

domain-name-server-name

Specify the name of a DNS server.

domain-name-server-IP-address

Specify the IP address of a DNS server.

[Top](#)

Examples

[Top](#)

Error messages

Unknown

[Top](#)

Open Data Base File (OPNDBF)

Where allowed to run: All environments (*ALL)
 Threadsafes: 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.

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.

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.

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.

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 (OPNQRYP)

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

Parameters
Examples
Error messages

The Open Query File (OPNQRYP) 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.

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 OPNQRYP command.
2. The OPNQRYP 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 OPNQRYP 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 OPNQRYP command cannot be used with those functions.
5. The file opened with the OPNQRYP 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): <i>*INP, *OUT, *UPD, *DLT</i>	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	<i>*ASCEND, *DESCEND</i>	
	Element 3: Order by absolute value	<i>*ABSVAL</i>	
UNIQUEKEY	Unique key fields	1-120, <i>*NONE, *ALL</i>	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. 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 SQL Reference information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 10000 bytes. In addition, if the sum of the lengths of the key fields is greater than 2000 bytes, ***INP** must be specified for the **OPTION** parameter and fields cannot be ordered by their absolute value.

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 9994 bytes, since 10000 bytes - (3 variable-length fields * 2 bytes per field) = 9994 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.

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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.

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 CPYFRMQRYP 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(username qsysstrntbl) *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. QSYSSTRNTBL 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('username *ct ''GEORGE'')
        MAPFLD((username
              '%xlate(telefile/username qsysstrntbl)'))
```

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 orddtls) 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(*FIRSTIO)
        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 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  COND(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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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	*ACTGRPDFN, *CALLLVL, *JOB	Optional
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *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 name of 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** prompt (TOFILE parameter). Otherwise, any device file or database file name can be specified.

Top

Overriding to data base file (TOFILE)

Specifies the name of the database file that is used instead of the file specified on the **File being overridden** prompt (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.

***FILE** The database file named in the **File being overridden** prompt (FILE parameter) has some of its parameters overridden by values specified in this command.

database-file-name

Specify the name and library of the database file that is used instead of the file specified in the **File being overridden** prompt (FILE parameter).

The possible library values are:

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

***CURLIB**

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

library-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.

member-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 value other than *FILE has been specified in the **Overriding to data base file** prompt (TOFILE parameter), the first member in the file is used.

***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.

member-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** prompt (MBR parameter).

*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.

*RRN *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 precede the relative record number. For example, *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.

key-operation number-of-fields record-format-name key-value

Specify the first record that is read which is the record identified by the specified key operation, number of fields, record format name, and key value. If a record that matches these values does not exist, an exception is signaled.

Specify one of the following types of key-search types:

*KEYB (key-before)

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 (key-before or equal)

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 (Key-equal)

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 (key-after or equal)

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 (key-after)

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.

Specify the remaining search values as follows:

number-of-fields

Specify the number of key fields to use in the search. 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 *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.

record-format-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.

key-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 Data Management information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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 Programming book, SC41-5721.

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 enter multiple values for this parameter.

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 concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>, Appendix A. More information on journal management is in the Backup and Recovery 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.

number-of-write-operations-before-force

Specify the number of records. 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 name of a 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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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** prompt (MBR parameter).

program-name

Specify the selection program name, optionally qualified by the name of the library where the program is stored.

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

library-name

Specify the library where the program is located.

Top

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 concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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.

number-of-seconds

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/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.

The possible values are:

***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.

number-of-seconds

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. Valid values range from 1 through 32767. 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.

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** prompt (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.

The possible values are:

*NONE

Normal end-of-file processing is done.

number-of-seconds

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. Valid values range from 1 through 99999 seconds.

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** prompt (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** prompt (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.

*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.

More information on shared database files is in the Database information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

The possible values are:

- *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** prompt (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.

The possible values are:

- *ACTGRPDEFN**
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.

The possible values are:

- *NO The database file is not restricted to sequential only processing.
- *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

number-of-records

Specify *YES followed by a valid value ranging from 1 through 32767 for the number of records transferred each time. 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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/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 DB/2 Multisystem for iSeries information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

The possible values are:

***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 Diskette File (OVRDKTF)

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

Parameters
Examples
Error messages

The Override with Diskette File (OVRDKTF) 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 processed) is specified in the TOFILE parameter. The OVRDKTF 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 OVRDKTF 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 overriding files is in the Files and file systems topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>, the Application Display Programming book, SC41-5715, and the Printer Device Programming book, SC41-5713.

Note: Issuing 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 diskette file	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Overriding to diskette file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
DEV	Device	<i>Element list</i>	Optional, Positional 3
	Element 1: Diskette device	<i>Name</i>	
VOL	Volume identifier	Single values: *NONE Other values (up to 50 repetitions): <i>Character value</i>	Optional
LABEL	File label	<i>Character value</i>	Optional
EXCHTYPE	Diskette file exchange type	*STD, *BASIC, *H, *I	Optional
CODE	Code	*EBCDIC, *ASCII	Optional
CRTDATE	Creation date	<i>Date</i> , *NONE	Optional
EXPDATE	File expiration date	<i>Date</i> , *NONE, *PERM	Optional
SPOOL	Spool the data	*YES, *NO	Optional

Keyword	Description	Choices	Notes
OUTQ	Output queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Output queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MAXRCDS	Max spooled output records	1-500000, *NOMAX	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	<i>Character value, *BLANK</i>	Optional
IGCDTA	User specified DBCS data	*NO, *YES	Optional
WAITFILE	Maximum file wait time	<i>Integer, *IMMED, *CLS</i>	Optional
SECURE	Secure from other overrides	<u>*NO</u> , *YES	Optional
OVRSCOPE	Override scope	<u>*ACTGRPDEFN</u> , *CALLLVL, *JOB	Optional
SHARE	Share open data path	*YES, *NO	Optional
OPNSCOPE	Open scope	*ACTGRPDEFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the name of the file in the using program to which this override command is applied. The specified file must be a diskette unit file when *FILE is specified on the **Overriding to diskette file** prompt (TOFILE parameter). Otherwise, any device file or database file name can be specified.

Top

Overriding to diskette file (TOFILE)

Specifies the name of the diskette file that is used instead of the file specified on the **File being overridden** prompt (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 diskette device file, in the program, or in other called (OVRDKTF) commands.

***FILE** The diskette unit file named in the **File being overridden** prompt (FILE parameter) has some of its parameters overridden by values specified in this command.

diskette-unit-file-name

Specify the qualified name of the diskette unit file that is used instead of the overridden file.

The possible library values are:

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

***CURLIB**

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

library-name

Specify the library where the device file is located.

Top

Device (DEV)

Specifies the name of the diskette unit that is used with the diskette unit file to perform I/O operations. Specify the device name that overrides the device name specified in the diskette unit file, in the program, or in other called Override with Diskette File (OVRDKTF) commands. The device name of the IBM-supplied diskette unit description is QDKT. This parameter is ignored if a *YES value on the **Spool the data** prompt (SPOOL parameter) is in effect for the file when it is opened.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

device-name

Specify the name of the device that is used with this diskette device file. The device name must already exist on the system as a device description before this device file is created.

Top

Volume identifier (VOL)

Specifies one or more volume identifiers of the diskettes used by the diskette unit file. The volumes must be written on the device in the same order as their identifiers are specified in this parameter. If the file is opened for read backward, then 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). If a list of volume identifiers is provided for the file, operator messages indicate the name of the required volume.

Note: This parameter overrides the volume identifiers specified in the diskette device file, in the program, or in other called OVRDKTF commands.

***NONE**

The diskette volume identifiers are not specified for this file in this command. They can be specified later before the device file is opened, either in an Override with Diskette File (OVRDKTF) command, or a Change Diskette File (CHGDKTF) command, or in the high-level language program. Otherwise, no volume identifier checking is done.

volume-identifier

Specify the identifiers (up to 6 alphanumeric characters) in the order the volumes are written. A blank is used as a separator character when listing multiple identifiers.

You can enter multiple values for this parameter.

Top

File label (LABEL)

Specifies the data file label of the data file on diskette that is processed or created. For input files (diskette input to the system), this label specifies the identifier of the file that exists on the diskette. For output files (system output to diskette), it specifies the identifier of the file that is created on the diskette.

Note: This parameter overrides the label specified in the diskette device file, in the program, or in other called OVRDKTF commands.

data-file-label

Specify up to 8 characters for the identifier of the data file used with this diskette device file.

Top

Diskette file exchange type (EXCHTYPE)

Specifies, for diskette output files only, the exchange type used by the device file when the system is writing diskette data.

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

***STD** The BASIC exchange format is used for a type 1 or a type 2 diskette. The H exchange type is used for a type 2D diskette.

***BASIC**
The BASIC exchange type is used.

***H** The H exchange type is used.

***I** The I exchange type is used.

Top

Code (CODE)

Specifies the type of character code that is used by the device file when the system is reading or writing diskette data. The code can be either extended binary-coded decimal interchange code (*EBCDIC) or the American National Standard Code for Information Interchange (*ASCII).

This parameter overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

***EBCDIC**
The EBCDIC character code is used with this diskette unit file.

***ASCII**
The ASCII character code is used.

Top

Creation date (CRTDATE)

Specifies, for diskette input data files only, the date when the data file was created on the diskette. If the creation date specified in this parameter (if any) does not match the date written on the diskette, an error message is sent to the program.

This parameter overrides the values specified on the device file, in the program, or on other called OVRDKTF commands.

***NONE**
The creation date of the data file is not checked unless it is supplied before the device file is opened, either in an OVRDKTF command or CHGDKTF command, or in the high-level language program.

creation-date

Specify the creation date of the data file used by the device file. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP. However, the specified date is put in the diskette label in the format *yymmdd*.

Top

File expiration date (EXPDATE)

Specifies, for diskette output files only, the expiration date of the data file used by this device file. The data file is protected and cannot be written over until the day after the specified expiration date.

Note: The expiration date overrides the value specified in the device file, in the program, or in other called OVRDKTF commands.

*NONE

The data file is protected only on the day it is created.

*PERM

The data file is permanently protected. An expiration date of 999999 is written on the diskette.

expiration-date

Specify the date when the data file expires. The date must be specified in the format defined by the job attributes DATFMT and, if separators are used, DATSEP. However, the specified date is put in the diskette label in the format *yymmdd*.

Top

Spool the data (SPOOL)

Specifies whether the input or output data for the diskette unit file is spooled.

This parameter overrides the spool value specified in the device file, or in other called OVRDKTF commands.

***NO** The data is not spooled. If this file is opened for input, the data is read directly from the diskette. If this is an output file, the data is written directly to the diskette as it is processed by the program.

Note: If SPOOL(*NO) is specified, the following parameters in this command are ignored: OUTQ, MAXRCDS, SCHEDULE, HOLD, SAVE, OUTPTY, and USRDTA.

***YES** The data is spooled. If this file is opened for input, the named inline data file or the next unnamed inline data file is processed. If this is an output file, the data is spooled for processing by a spooling writer. More information on named and unnamed inline files is in the File systems and management information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

Output queue (OUTQ)

Specifies, for spooled output only, the name of the output queue for the spooled output file.

This parameter overrides the output queue name specified in the device file, or in other called OVRDKTF commands.

output-queue-name

Specify the name and library of the output queue to which the output data is spooled.

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

library-name

Specify the library where the output queue is located.

Top

Max spooled output records (MAXRCDS)

Specifies, for spooled output only, the maximum number of records that can be in the spooled output file for spooled jobs using the diskette device file.

This parameter overrides the value specified in the diskette device file, or in other called OVRDKTF commands.

***NOMAX**

There is no maximum on the number of records that can be in the spooled output file.

maximum-records

Specify a value ranging from 1 through 500000 that indicates the maximum number of records that can be in the spooled output file.

Top

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 device file, or in other called OVRDKTF commands.

***JOBEND**

The spooled output file is made available to the spooling writer only after the entire job is completed.

***FILEEND**

The spooled output file is made available to the spooling writer when the file is closed in the program.

***IMMED**

The spooled output file is made available to the spooling writer when the first output records are produced by the program.

Top

Hold spooled file (HOLD)

Specifies, for spooled output files only, whether the spooled file is held. The file is released by the Release Spooled File (RLSSPLF) command.

Note: This parameter overrides the hold value specified in the diskette device file, or in other called OVRDKTF commands.

- *NO The spooled output file is not held on the output queue. The spooled output is made available to a spooling writer based on the value on the **Spooled output schedule** prompt (SCHEDULE parameter).
- *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 files only, whether the spooled file is saved (left on the output queue) after the output is produced.

Note: This parameter overrides the save value specified in the diskette device file, or in other called OVRDKTF commands.

- *NO The spooled file data is not saved on the output queue after it is produced.
- *YES The spooled file data is saved on the output queue until the file is deleted. After the file is produced, its status 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) for the output priority.

Top

User data (USRDTA)

Specifies, for spooled output, user-specified data that identifies the file.

*BLANK

Ten blanks are used as the user data.

user-data

Up to 10 characters of user specified text is allowed.

Top

User specified DBCS data (IGCDTA)

Specifies whether the file processes double-byte character set (DBCS) data.

- *NO The file does not process double-byte character set (DBCS) data.
- *YES The file processes double-byte character set (DBCS) data.

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.

number-of-seconds

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

Secure from other overrides (SECURE)

Specifies whether this file is protected from the effects of any previously called file override commands.

***NO** This file is not protected from other file overrides. Its values are overridden by the effects of previously called file override commands.

***YES** This file is protected from the effects of 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.

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 OVRDKTF 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: Changing Spooling Specifications

```
OVRDKTF FILE(OUT) VOL(DPT706) LABEL(STATUSR) SPOOL(*YES)
```

This command changes the spooling specification for the output file named OUT. When a program produces output data for the OUT file, the data is spooled for processing by a spooling writer. The writer processes the data by writing it in a data file called STATUSR that is on a diskette whose volume identifier is DPT706.

Example 2: Specifying DBCS Processing

```
OVRDKTF FILE(IGCLIB/IGCDCT) IGCDTA(*YES)
```

This command overrides the diskette file IGCDCT, which is stored in the library IGCLIB, so that the file contains double-byte character set data.

Top

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

Top

Override with Display File (OVRDSPF)

Where allowed to run: All environments (*ALL)
Threadsafe: 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 name of 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** prompt (TOFILE parameter). Otherwise, any device file or database file name can be specified.

Top

Overriding to display file (TOFILE)

Specifies the name of the display file that is used instead of the file specified in the **File being overridden** prompt (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.

***FILE** The display device file named in the **File being overridden** prompt (FILE parameter) has some of its parameters overridden by values specified in this command.

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

library-name

Specify the library where the device file is located.

display-device-file-name

Specify the name and library of the display device file that is used instead of the overridden file.

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.

You can enter multiple values for this parameter.

***REQUESTER**

The display device from which the program is called is the device assigned to the file when the file is opened.

device-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** prompt (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.

***DEVVD**

The CHRID value specified in the device description of the work station on which the application is running is used. The *DEVVD 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 (CRTDEVVDSP) 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 *DEVVD on the CHRID command parameter for this file.

graphic-character-set code-page

Specify the graphic character set and code page values that match 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.

The possible values are:

- *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.

The possible values are:

- *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, the possible values are:

- *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, the possible values are:

- *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.

Top

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.

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 concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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.

The possible values are:

***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.

number-of-seconds

Specify the number of seconds that the program waits for the allocation of the file resources. Valid values range from 1 through 32767 seconds.

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.

The possible values are:

***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.

number-of-seconds

Specify the number of seconds that the program waits for the completion of the read-from-invited-program-device operations. Valid value range from 1 through 32767.

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.

The possible value is:

***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 name of 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 Programming book, SC41-5721.

The possible values are:

***NONE**

No data queue is specified.

data-queue-name

Specify the name of the data queue on which entries are placed.

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 is used to locate the data queue. If no library is specified as the current library, QGPL is used.

library-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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/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.

The possible values are:

***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)
 Threadsafes: 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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>, the Application Display Programming book, SC41-5715, and the Printer Device Programming book, SC41-5713.

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 D, *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 D	Optional
BLKLEN	Block length	1-32767, *DEV D	Optional
TRNSPY	Transmit in transparent mode	*DEV D, *NO, *YES	Optional
DTACPR	Data compression	*DEV D, *NO, *YES	Optional
TRUNC	Truncate trailing blanks	*DEV D, *NO, *YES	Optional
OVFLWDTA	Overflow data	*DISCARD, *RETAIN	Optional
GRPSEP	Group separator type	*DEV D, *EOT, *DEV3740, *OFCSYS	Optional
RMTBSCSEL	Remote BSCSEL	*DEV D, *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.

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.

remote-location-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** prompt (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:

- Device prompt (DEV parameter).
- Local location prompt (LCLLOCNAME parameter).
- Mode prompt (MODE parameter).
- Remote network identifier prompt (RMTNETID parameter).

Top

Communication type (CMNTYPE)

Specifies the communications type for which parameters appear in the prompt. 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.

device-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.

local-location-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.

mode-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.

remote-network-id

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(*DEV D) is used.

***DEV D**

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.

application-id

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** prompt (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(*DEV D) is used.

***DEV D**

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.

Top

Special host application (SPCHOSTAPP)

Specifies whether SNUF customizes support for special host applications outside the CICS or IMS application layer.

***DEV D**

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.

Top

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.

Top

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.

Top

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** prompt (BATCH parameter).

*NO Message protection is not used.

Top

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).

*NONE

This program device entry is not used to send and receive 3270 data streams.

The **Device type** values:

- 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.

The **Data format** values:

If an emulation device type is specified, one of the following emulation device data formats can be specified:

***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 BSCCL 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 BSCCL 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.

***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, BSCCL framing characters, transparency characters, and so forth, necessary to send records.

***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.

X'1E' The record separator character X'1E' is used.

record-separator-character

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 @). If a record separator character is not specified, the record separator character of X'1E' is used.

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(*DEV D) is used.

*DEV D

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.

record-length

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 *DEV D.

*DEV D

The block length specified in the device description is used.

block-length

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(*DEV D) is used.

*DEV D

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** prompt (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** prompt (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(*DEVVD) is used.

*DEVVD

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** prompt (BLOCK parameter). If *YES is specified and *YES is also specified in the **Data compression** prompt (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(*DEVVD) is used.

*DEVVD

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 BSCCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, GRPSEP(*DEVVD) is used.

*DEVVD

The group separator option specified in the device description is used.

*EOT The BSCCEL 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 BSCCEL control character ETX (end-of-information) is used as a data group separator.

Top

Remote BSCCEL (RMTBSCCEL)

Specifies whether the type of BSCCEL session is with a BSCCEL system. This parameter applies to the BSCCEL communications type only. If the ADDICFDEVE command is not run for the specified program device and this parameter is not overridden, RMTBSCCEL(*DEVVD) is used.

***DEVVD**

The option for BSCCEL specified in the device description is used.

***NO** An attribute of *NO indicates the remote system cannot recognize BSCCEL 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 BSCCEL 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 BSCCEL 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 BSCCEL 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.

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.

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.

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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>, the Application Display Programming book, SC41-5715, and the Printer Device Programming book, SC41-5713.

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	
MAXRCDLEN	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
OVRSCOPE	Override scope	*ACTGRPDFN, *CALLLVL, *JOB	Optional

Keyword	Description	Choices	Notes
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, *LIBL, *CURLIB</i>	
SHARE	Share open data path	*NO, *YES	Optional
OPNSCOPE	Open scope	*ACTGRPDFN, *JOB	Optional

Top

File being overridden (FILE)

Specifies the name of the file (up to 10 characters) 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** prompt (TOFILE parameter). Otherwise, any device file or database file name can be specified.

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** prompt (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.

***FILE** Some parameters of the ICF file named in the **File being overridden** prompt (FILE parameter) are overridden by values specified in this command.

ICF-communications-file

Specify the name and library of the ICF file that is used instead of the overridden file.

The possible library values are:

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

***CURLIB**

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

library-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.

program-device-name

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.

record-length

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.

number-of-seconds

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.

number-of-seconds

Specify the number of seconds that the program waits for the completion of the read-from-invited-program-devices operation. Valid values range from 1 through 32767.

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 concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseriess/infocenter>.

***NONE**

No data queue is specified.

data-queue-name

Specify the name of the data queue on which entries are placed.

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 is used to locate the data queue. If no library is specified as the current library, QGPL is used.

library-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.

***ACTGRPDFN**

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.

Top

Examples

```
OVRICFF FILE(ICFHIST) TOFILE(PRSNNL/ICFCURT)
```

This command overrides the file named ICFHIST to the ICF file named ICFCURT in library PRSNNL.

Error messages

*ESCAPE Messages

CPF180C

Function &1 not allowed.

CPF1892

Function &1 not allowed.

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:

1. In a multithreaded job, this command may only be issued from the initial thread.
2. 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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>, the Application Display Programming book, SC41-5715, and the Printer Device Programming book, SC41-5713.

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 name of the message file being used by the program to which this override command is applied.

Top

Overriding to message file (TOMSGF)

Specifies the name and library of the message file that is used instead of the message file specified in the **Message file being overridden** prompt (MSGF parameter); or, if the names are the same, specifies that the value specified in the **Secure from other overrides** prompt (SECURE parameter) is used for the message file.

message-file-name

Specify the name of the message file that is used instead of of the overridden message file.

The possible library values are:

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

***CURLIB**

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

library-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.

Top

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:

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.

More information on overriding files is in the Printer Device Programming book, SC41-5713.

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	Name, *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	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
DEV	Device	Element list	Optional, Positional 3
	Element 1: Printer	Name, *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, 7.5, 8.0, 9.0, 12.0	Optional
CPI	Characters per inch	5.0, 10.0, 12.0, 13.3, 13.3, 15.0, 16.7, 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' , X'40'	
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	Spoiled output schedule	*JOBEND, *FILEEND, *IMMED	Optional
HOLD	Hold spoiled file	*NO, *YES	Optional
SAVE	Save spoiled 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, *USRSPC	
SPLFNAME	Spool file name	Name, *FILE	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	
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	*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 name of 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** prompt (TOFILE parameter). Otherwise, any device file name or database file name is specified.

***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.

file-override-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 name of the printer file that is used instead of the file specified in the **File being overridden** prompt (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.

***FILE** The printer device file named in the **File being overridden** prompt (FILE parameter) has some of its parameters overridden by values specified in this command.

printer-device-file-name

Specify the name and library of the printer device file that is used instead of the overridden file.

The possible library values are:

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

***CURLIB**

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

library-name

Specify the library where the file is located.

Top

Device (DEV)

Specifies the name of 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.

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 Printer Device Programming book, SC41-5713.

***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.

***AFPDSL**

Mixed data (line data and AFPDS data) is created. This value can be specified when using any printer supported by PSF/400. 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/400. 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.

The possible values are:

page-length

Specify the page length used by this device file. Although a value ranging from 1 through 255 is allowed, the value specified should not exceed the actual length of the forms used.

page-width

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.

The possible **method of measure** values are:

***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.

The possible values are:

- 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

The possible values are:

- 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.

*DEV D

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).

The possible **offset-down** values are:

- 0 No offset of the origin occurs.

offset-down

Specify the offset of the origin from the top of the page.

The possible **offset-across** values are:

- 0 No offset of the origin occurs.

offset-across

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.

The possible values are:

*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).

The possible **offset-down** values are:

0 No offset of the origin occurs.

offset-down

Specify the offset of the origin from the top of the page.

The possible **offset-across** values are:

0 No offset of the origin occurs.

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** prompt (PAGESIZE parameter) for the file. Margins specified for the printer file are ignored when determining overflow. More information is in the Printer Device Programming book, SC41-5713. This parameter overrides the overflow value specified in the printer file, in the program, or in other called OVRPRTF commands.

The possible values are:

overflow-line-number

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 for AS/400 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** prompt (DEVTYPE parameter), a hyphen (-) is printed for the unprintable characters. The substitution character is ignored for the 3287 printer.

The possible values are:

***YES** Unprintable characters are replaced. The program is not notified when unprintable characters are detected.

***NO** Unprintable characters are not replaced. When an unprintable character is detected, a message is sent to the program.

' '
— A blank is used as the substitution character when an unprintable character is detected and *YES is specified.

'replacement-character'

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** prompt (IGCEXNCHR parameter), the system replaces unprintable characters with double-byte character set (DBCS) underscores.
2. If *NO is specified in the **DBCS extension characters** prompt (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** prompt (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 the data** prompt (SPOOL parameter), and *FILE is specified on the **Align page** prompt (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.

The possible values are:

- *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** prompt (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.

source-drawer

Specify the drawer from which the paper is fed. Valid values range from 1 through 255.

Top

Output bin (OUTBIN)

Specifies the destination of the output on printers capable of multiple output bins.

The possible values are:

***DEV D**

The destination of the output is the device default output bin.

output-bin

Specify the output bin for the destination of the output. Valid values range from 1 through 65535.

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.

The possible **font identifier** values are:

*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.

identifier

Specify the numeric font identifier being used with this printer device file.

The possible **point size** values are:

***NONE**

No point size is specified; the system sets one based on the type of printer being used.

point-size

Specify a point size. Valid values range from .1 through 999.9 points.

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 *DEV. 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 (*DEV D) 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(*DEV D) and PRTQLTY(*DRAFT) on this command to enable automatic rotation if the data does not fit on the paper.

The possible values are:

***STD** The output is printed with standard quality.

***DRAFT**
The output is printed with draft quality.

***DEV D**
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** prompt (CTLCHAR parameter).

Note: If one or more channel-number/line-number combinations are changed, all other combinations must be re-entered.

*NORMAL

The default values for skipping to channel identifiers are used.

channel-number

Specify an American National Standard channel number to be associated with a corresponding 'skip to' line number. Valid values for this parameter range from 1 through 12, corresponding 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.

line-number

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).

The possible values are:

*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.

The possible values are:

*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.

graphic-character-set code-page

Specify the graphic character set and code page values that match 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.

The possible values are:

*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.

The possible values are:

*FONT

The value specified on the FONT parameter is used.

The **font character set** value can be qualified by one of the following library values:

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

***CURLIB**

The current library for the 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.

library-name

Specify the name of the library where the font character set is located.

The **code page name** value can be qualified by one of the following library values:

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

***CURLIB**

The current library for the 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.

library-name

Specify the name of the library where the code page name is located.

The possible **Point size** values are:

***NONE**

The point size is supplied by the system and is determined by the specified font character set.

point-size

Specify a point size ranging from 0.1 through 999.9.

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.

The possible values are:

***FNTCHRSET**

The font specified on the FNTCHRSET parameter is used.

coded-font-name

Specify the coded font name to use.

The **coded font name** value can be qualified by one of the following library values:

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

***CURLIB**

The current library for the 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.

library-name

Specify the name of the library where the coded font name is located.

The possible **Point size** values are:

***NONE**

The point size is supplied by the system and is determined by the specified font character set.

point-size

Specify a point size ranging from 0.1 through 999.9.

Top

Page definition (PAGDFN)

Specifies the page definition to be used to format line data.

You can specify a page definition with *LINE, *AFPDSLNE, 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

The possible values are:

***NONE**

No page definition is specified.

Because PSF/400 requires a page definition when *LINE or *AFPDSLNE is specified, an inline page definition is built from the print file parameters and passed to PSF/400 when *NONE is specified.

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 *AFPDSLNE, *LINE, or *USERASCII must be specified when using a page definition.

The name of the page definition can be qualified by one of the following library values:

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

***CURLIB**

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

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 (*DEV D 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 *FORMD F specified)
- DRAWER (If *FORMD F specified)
- PAGRTT
- PRTQLTY
- FORMFEED
- FRONTMGN
- BACKMGN
- MULTIUP
- REDUCE
- CORNERSTPL
- EDGESTITCH
- SADLSTITCH

The possible values are:

***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.

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.

The name of the form definition can be qualified by one of the following library values:

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

***CURLIB**

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

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.

The possible values are:

***NONE**

No AFP characters (coded fonts) specified.

coded-font-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.

Table Reference Characters (TBLREFCHR)

Specifies whether table reference characters are present in the line data.

The possible values are:

***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 (PAGRRT)

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**

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.

The possible values are:

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 Printer Device Programming book, SC41-5713.

The possible values are:

***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.

'print-text'

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.

The possible values are:

*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.

The possible values are:

*NONE

No overlay is used.

The possible **Overlay** values are:

overlay-name

Specify the name of the overlay.

The name of the overlay can be qualified by one of the following library values:

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

*CURLIB

The current library for the job is used to locate the overlay. 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 overlay is located.

The possible **Offset down** values are:

0 No offset down from the point of origin is used.

offset-down

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.

The possible **Offset across** values are:

0 No offset across from the point of origin is used.

offset-across

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.

The possible values are:

*FRONTOVL

The values that are specified on the FRONTOVL parameter are used.

*NONE

No overlay is used.

The possible **Overlay** values are:

overlay-name

Specify the name of the overlay.

*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.

library-name

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

The possible **Offset down** values are:

0 No offset down from the point of origin is used.

offset-down

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.

The possible **Offset across** values are:

0 No offset across from the point of origin is used.

offset-across

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.

The possible **Constant Back** values are:

*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.

The possible values are:

*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.

The possible values are:

*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 USRDFNDDTA parameter in a printer file. You may continue using this support with existing printer files and PSF configuration objects by specifying IPDSPASTHR(*DEVD) in the printer file. If you specify a value of anything other than *DEVD for the IPDSPASTHR parameter, any IPDS pass-through value in the USRDFNDDTA 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 QFNTPCL, QFNTP01-QFNTP19, and QFNTP61-69 are searched.

The possible values are:

***DEVD**

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 *JOBLIBL 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.

user-resource-library-name

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(*DEVD) in the printer file. If you specify a value of anything other than *DEVD 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.

The possible values are:

***NONE**

A corner staple is not specified.

***DEVD**

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 *DEVD for that element. Page rotation does not affect the placement of an edge stitch.

The possible values are:

***NONE**

An edge stitch is not specified.

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. Possible values are:

***DEVD**

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.

Reference Edge Offset specifies the offset of the edge stitch from the reference edge toward the center of the media.

***DEVD**

The reference edge offset is the default reference edge offset used by the device.

reference-edge-offset

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.

Number of Staples specifies the number of staples that are to be applied along the finishing operation axis.

***DEVD**

The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEVD 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.

number-of-staples

Specify the number of staples to be used for the edge stitch. Valid values range from 1 to 122 staples. If the number of staples is specified, then *DEVD must be specified for staple offsets. The device default for the spacing of each staple will be used.

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.

staple-offset

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 *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

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.

The possible values are:

***NONE**

A saddle stitch is not specified.

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. Possible values are:

***DEVD**

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.

Number of Staples specifies the number of staples that are to be applied along the finishing operation axis.

***DEVD**

The number of staples depends on the value of the Staple Offsets element of this parameter. If *DEVD 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.

number-of-staples

Specify the number of staples to be used for the saddle stitch. Valid values range from 1 to 122 staples. If you specify the number of staples, then *DEVD must be specified for staple offsets. The device default for the spacing of each staple will be used.

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.

staple-offset

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.

The possible values are:

***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.

The possible values are:

***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** prompt (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 the data** prompt (SPOOL parameter) and if ***IMMED** is specified on the **Spooled output schedule** prompt (SCHEDULE parameter), output is held in the buffer until a page of output is available or until the system buffer is full.

Top

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.

***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.

output-queue-name

Specify the name and library of the output queue to which the output data is spooled.

The possible library values are:

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

***CURLIB**

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

library-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.

form-type

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.

Top

Page range to print (PAGERANGE)

Specifies, for spooled output files only, the starting and ending pages to print.

The possible **starting page** values are:

***ENDPAGE**

Use the end page value as the starting page.

starting-page-number

Specify the starting page to print.

The possible **ending page** values are:

***END** Printing continues until the end of file.

ending-page-number

Specify the ending page to print.

Top

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.

The possible values are:

***NOMAX**

There is no maximum on the number of records that can be in the spooled file.

maximum-records

Specify a value ranging from 1 through 999999 (999,999) to indicate the maximum number of records that can be in the spooled output file.

Top

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.

The possible values are:

number-of-file-separators

Specify the number of separator pages used at the start of each printed output file produced by this device file. Valid values range from 0 through 9. 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.

Top

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.

The possible values are:

***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** prompt (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.

The possible values are:

- *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.

user-data

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.

The possible values are:

***CURUSRPRF**

The spooled file is owned by the current effective user of the current job or thread. See the Printer Device Programming book, SC41-5713 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 Printer Device Programming book, SC41-5713 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.

The possible values are:

***NONE**

No user-defined options specified.

user-defined-option

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 (USRDFNDTA)

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.

The possible values are:

***NONE**

No user-defined data specified.

user-defined-data

Specify a user-defined data to be used by user applications or user-specified programs that process spooled files. All characters are acceptable.

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.

The possible values are:

***NONE**

No user-defined object specified.

The possible **User-Defined Object Name** values are:

object-name

Specify the user-defined object to be used by user applications or user-specified programs that process spooled files.

The user-defined object can be qualified by one of the following library values:

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

***CURLIB**

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

name Specify the name of the library to be searched.

The possible **User-Defined Object Type** values are:

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

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.

spool-file-name

Specify the name of the spooled output file. A maximum of 10 characters can be used.

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, the possible values are:

***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, the possible values are:

***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** prompt (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 double-byte characters.

***YES** The system prints double-byte characters as blanks.

***NO** The system does not print double-byte 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).

The possible values are:

***SYSVAL**

The DBCS coded font specified in the system value is used.

The possible **coded font name** values are:

coded-font-name

Specify the DBCS coded font name to use.

The name of the coded font can be qualified by one of the following library values:

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

***CURLIB**

The current library for the job is used to locate the coded font name. 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 coded font name is located.

The possible **Point size** values are:

***NONE**

The point size is supplied by the system and is determined by the specified font character set.

point-size

Specify a point size ranging from 0.1 through 999.9.

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.

number-of-seconds

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.

*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.

The possible values are:

*ACTGRPDEFN

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(IGCLIB/IGCPRT) IGCDTA(*YES) IGCCHRTT(*YES)
```

This command overrides the IGCPRT printer file, which is stored in the IGCLIB library. 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.

[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 topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 name of the 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** prompt (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.

Top

Save file (TOFILE)

Specifies the name of the save file that is used instead of the file specified on the **File being overridden** prompt (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.

***FILE** The save file named in the **File being overridden** prompt (FILE parameter) has certain parameters overridden by the values specified in this command.

save-file-name

Specify the name and library of the save file that is used instead of the overridden file name.

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 save file. If no library is specified as the current library for the job, QGPL is used.

library-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.

***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.

***RRN** *relative-record-number*

Specify the record number (its position from the beginning of the file) of the record that you get first. The value *RRN must go before the relative record number. For example, *RRN 480 specifies that the 480th record in the file is gotten 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.

number-of-seconds

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.

***ACTGRPDFN**

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

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY8809
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
IBM
Infoprint
InfoWindow
iSeries
LPDA
OfficeVision

OS/400
Print Services Facility
RPG/400
SystemView
System/36
TCS
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.



Printed in USA