



IBM Systems - iSeries
i5/OS Commands
Starting with ENDJOB (End Job)

Version 5 Release 4





IBM Systems - iSeries

i5/OS Commands

Starting with ENDJOB (End Job)

Version 5 Release 4

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 455.

Second Edition (February 2006)

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

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.

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

Contents

End Job (ENDJOB)	1	Examples	29
Parameters	1	Error messages	29
Job name (JOB)	2		
How to end (OPTION)	2	End Job Log Server (ENDLOGSVR)	31
Controlled end delay time (DELAY)	3	Parameters	31
Delete spooled files (SPLFILE)	3	How to end (OPTION)	31
Maximum log entries (LOGLMT)	3	Examples	32
Additional interactive jobs (ADLINTJOBS)	4	Error messages	32
Duplicate job option (DUPJOB OPT).	4		
Examples	4	End Mode (ENDMOD)	33
Error messages	5	Parameters	33
		Remote location (RMTLOCNAME)	33
End Job Abnormal (ENDJOBABN)	7	Device (DEV).	33
Parameters	8	Mode (MODE)	34
Job name (JOB)	8	Local location (LCLLOCNAME)	34
Duplicate job option (DUPJOB OPT).	9	Remote network identifier (RMTNETID).	34
Examples	9	Complete pended requests (CPLPNDRQS)	35
Error messages	9	Examples	35
		Error messages	35
End Journal (ENDJRN)	11		
Parameters	11	End Mail Server Framework (ENDMSF) 37	
Objects (OBJ)	12	Parameters	37
File identifier (OBJFID)	12	How to end (OPTION)	37
Directory subtree (SUBTREE)	13	Controlled end delay time (DELAY)	37
Name pattern (PATTERN)	13	Examples	38
Journal (JRN)	14	Error messages	38
Examples	14		
Error messages	15	End NFS Server (ENDNFSSVR)	39
		Parameters	39
End Journal Access Path (ENDJRNAP) 17		Server daemon (SERVER).	40
Parameters	17	Timeout for end of daemon (ENDJOB TIMO)	40
Journalled file (FILE)	18	Examples	40
Journal (JRN)	18	Error messages	40
Examples	18		
Error messages	19	End NetWare Connection	
		(ENDNTWCNN)	43
End Journal Object (ENDJRNOBJ).	21	Parameters	43
Parameters	21	Server (SERVER)	43
Object (OBJ)	22	Connection number (CNNNBR)	43
Object type (OBJTYPE)	22	Examples	43
Journal (JRN)	22	Error messages	43
Examples	23		
Error messages	23	End Network Interface Recovery	
		(ENDNWIRCY)	45
End Journal Physical File (ENDJRNP F) 25		Parameters	45
Parameters	25	Network interface description (NWI)	45
Journalled physical file (FILE)	26	Examples	45
Journal (JRN)	26	Error messages	45
Examples	27		
Error messages	27	End Pass-Through (ENDPASTHR)	47
		Parameters	47
End Line Recovery (ENDLINRCY)	29	Job log (LOG)	47
Parameters	29	Examples	48
Line (LINE)	29		

Error messages	48
End Performance Explorer (ENDPEX)	49
Parameters	49
Session ID (SSNID)	50
Option (OPTION)	50
Data option (DTAOPT)	51
Data library (DTALIB)	51
Data member (DTAMBR)	51
Management collection (MGTCOL)	52
Replace data (RPLDTA)	52
Number of threads (NBRTHD)	52
Text 'description' (TEXT)	53
Examples	53
Error messages	53
End Performance Collection (ENDPFCOL)	55
Parameters	55
Force end (FRCCOLEND)	55
Examples	55
Error messages	56
End Performance Trace (ENDPFTRC)	57
Parameters	57
Dump the trace (DMPTRC)	57
Member (MBR)	57
Library (LIB)	58
Text 'description' (TEXT)	58
Examples	58
Error messages	58
End Program (ENDPGM)	59
Parameters	59
Examples	59
Error messages	59
End Program Export List (ENDPGMEXP)	61
Parameters	61
Examples	61
Error messages	61
End Program Profiling (ENDPGMPRF)	63
Parameters	63
Examples	63
Error messages	63
End Prestart Jobs (ENDPJ)	65
Parameters	65
Subsystem (SBS)	65
Program (PGM)	66
How to end (OPTION)	66
Controlled end delay time (DELAY)	66
Delete spooled files (SPLFILE)	67
Maximum log entries (LOGLMT)	67
Examples	67
Error messages	68

End Printer Emulation (ENDPRTEML)	69
Parameters	69
Emulation device (EMLDEV)	69
Emulation location (EMLLOC)	70
Print device (PRTDEV)	70
Examples	70
Error messages	70
End Receive (ENDRCV)	71
Parameters	71
Display device (DEV)	71
Open file identifier (OPNID)	71
Examples	72
Error messages	72
(ENDRDBRQS)	73
Parameters	73
Examples	73
Error messages	73
End Reader (ENDRDR)	75
Parameters	75
Reader (RDR)	75
When to end reader (OPTION)	75
Examples	75
Error messages	76
End Remote Support (ENDRMTSPT)	77
Parameters	77
Delete library (DLTLIB)	77
How to end (OPTION)	77
Examples	78
Error messages	78
End RPC Binder Daemon (ENDRPCBIND)	79
Parameters	79
Examples	79
Error messages	79
End Request (ENDRQS)	81
Parameters	81
Request level (RQSLVL)	81
Examples	82
Error messages	82
End S/36 Session (ENDS36)	83
Parameters	83
Examples	83
Error messages	83
End Subsystem (ENDSBS)	85
Parameters	85
Subsystem (SBS)	86
How to end (OPTION)	86
Controlled end delay time (DELAY)	87
End subsystem option (ENDSBSOPT)	87
Batch time limit (BCHTMLMT)	88

Examples	88
Error messages	88
End Select (ENDSELECT).	91
Parameters	91
Examples	91
Error messages	91
End Service Agent (ENDSRVAGT)	93
Parameters	93
Type (TYPE)	93
Examples	93
Error messages	93
End Service Job (ENDSRVJOB)	95
Parameters	95
Examples	95
Error messages	95
End Subroutine (ENDSUBR)	97
Parameters	97
Return value (RTNVAL)	97
Examples	98
Error messages	98
End System (ENDSYS)	99
Parameters	99
How to end (OPTION)	99
Controlled end delay time (DELAY)	100
End subsystem option (ENDSBSOPT)	100
Confirm (CONFIRM)	101
Examples	101
Error messages	101
End TCP/IP (ENDTCP).	103
Parameters	103
How to end (OPTION)	104
Controlled end delay time (DELAY)	104
End application servers (ENDSVR)	104
Examples	105
Error messages	105
End TCP/IP Abnormal (ENDTCPABN)	107
Parameters	107
Examples	107
Error messages	107
End TCP/IP Connection (ENDTCPENN)	109
Parameters	109
Protocol (PROTOCOL)	109
Local internet address (LCLINTNETA)	109
Local port (LCLPORT)	110
Remote internet address (RMTINTNETA)	110
Remote port (RMTPORT)	110
Examples	110
Error messages	111

End TCP/IP Interface (ENDTCPIFC)	113
Warning: Temporary Level 2 Header	113
Parameters	113
Internet address (INTNETADR)	113
Alias name (ALIASNAME)	114
Examples	114
Error messages	114
End Point-to-Point TCP/IP (ENDTCPPTP)	117
Parameters	117
Configuration profile (CFGPRF)	117
Operating mode (OPRMODE)	118
Examples	118
Error messages	118
End TCP/IP Server (ENDTCPSVR).	121
Parameters	121
Server application (SERVER)	121
HTTP server (HTTPSVR)	124
DNS server (DNSSVR)	124
TCM server (TCMSVR)	124
ASFTOMCAT server (TOMCATSVR)	124
Examples	125
Error messages	125
End TIE Session (ENDTIESSN)	127
Parameters	127
Examples	127
Error messages	127
End Trace (ENDTRC)	129
Parameters	129
Session ID (SSNID)	129
Data option (DTAOPT)	130
Data library (DTALIB)	130
Replace data (RPLDTA)	130
Print trace data (PRTRC)	130
Examples	131
Error messages	131
End Trap Manager (ENDTRPMGR)	133
Parameters	133
Examples	133
Error messages	133
End Watch (ENDWCH)	135
Parameters	135
Session ID (SSNID)	135
Examples	136
Error messages	136
End Writer (ENDWTR).	137
Parameters	137
Writer (WTR)	137
When to end writer (OPTION)	137
Examples	138
Error messages	138

Remove Link (ERASE) 139

Parameters	140
Object link (OBJLNK)	140
Examples	140
Error messages	140

Export a Program Symbol (EXPORT) 143

Parameters	143
Exported symbol name (SYMBOL)	143
Examples	143
Error messages	143

Change NFS Export (EXPORTFS). . . 145

Parameters	145
NFS export options (OPTIONS)	146
Directory (DIR)	149
Host name (HOSTOPT)	149
Examples	150
Error messages	151

Extract Program Information (EXTPGMINF) 153

Error messages for EXTPGMINF	153
Parameters	153
Program (PGM)	153
File to receive information (FILE)	154
Extract record options (OPTION)	154
Create the file (CRTFILE)	155
Library name to record (RECLIB)	155
Consistency Check (CHECK)	155
Examples	155
Error messages	155

File Document (FILDOC). 157

Parameters	157
Information to be filed (TYPE)	158
To document (TODOC)	159
To folder (TOFLR)	159
Sensitivity (SENSITIV)	159
User authority (USRAUT)	160
Authorization list (AUTL)	160
Access code (ACC)	161
Allow replacement (ALWRPL)	161
Profile file (IDPFILE)	161
Profile member (IDPMBR)	162
User identifier (USRID)	162
Document file (DOCFILE)	162
Document member (DOCMBR)	163
Distribution identifier (DSTID)	163
Distribution ID extension (DSTIDEXN)	163
Keep in mail log (KEEP)	163
Document type (DOCTYPE)	164
System code (SYSCOD)	164
Document description (DOCD)	164
Author (AUTHOR)	165
Document class (DOCCLS)	165
Keyword (KWD)	165
Subject (SUBJECT)	165
Document date (DOCDATE)	166
File cabinet location (FILCAB)	166

Copy list (CPYLST)	166
Expiration date (EXPDATE)	167
Reference (REFERENCE)	167
Action due date (ACTDATE)	167
Document status (STATUS)	167
Completion date (CMPDATE)	168
Project (PROJECT)	168
Document character identifier (DOCCHRID)	168
Language ID (DOCLANGID)	169
Country or region ID (DOCCNTRYID)	169
Personal (PERSONAL)	169
Distribution expiry indicator (DSTEXPDATE)	169
Command character identifier (CMDCHRID)	170
Examples	171
Error messages	171

Format Data (FMTDTA) 173

Parameters	173
Input file (INFILE)	173
Output file (OUTFILE)	174
Source file (SRCFILE)	174
Source member (SRCMBR)	175
Print file (PRTFILE)	175
Options: (OPTION)	175
Program date: (PGMDATE)	176
Examples	176
Error messages	176

Generate Message Catalog (GENCAT) 177

Parameters	177
Message catalog name (CLGFILE)	177
Source file path name (SRCFILE)	178
Text 'description' (TEXT)	178
Message catalog CCSID (CLGCCSID)	178
Source file CCSID (SRCCCSID)	178
Public authority for data (DTAAUT)	179
Public authority for object (OBJAUT)	180
Examples	180
Examples for MRGMSGCLG	180
Error messages	180

Generate Command Documentation (GENCMDDOC) 181

Parameters	181
Command (CMD)	182
To directory (TODIR)	182
To stream file (TOSTMF)	182
Replace file (REPLACE)	183
Generation options (GENOPT)	183
Examples	184
Error messages	184

Go to Menu (GO) 187

Parameters	187
Menu (MENU)	187
Return point (RTNPNT)	188
Examples	189
Error messages	189

Go To (GOTO) 191

Parameters	191
Command label (CMDLBL)	191
Examples	191
Error messages	192

Grant Access Code Authority (GRTACCAUT) 193

Parameters	193
Document access code (ACC)	193
User profile (USER)	194
Reference user profile (REFUSER)	194
Examples	194
Error messages	194

Grant Object Authority (GRTOBJAUT) 197

Parameters	198
Object (OBJ)	199
Object type (OBJTYPE)	199
ASP device (ASPDEV)	200
Users (USER)	200
Authority (AUT)	200
Authorization list (AUTL)	202
Reference object (REFOBJ)	202
Reference object type (REFOBJTYPE)	202
Reference ASP device (REFASPDEV)	203
Replace authority (REPLACE)	203
Examples	203
Error messages	204

Grant User Authority (GRTUSRAUT) 207

Parameters	207
User (USER)	208
Referenced user (REFUSER)	208
Examples	208
Error messages	208

Grant User Permission (GRTUSRPMN) 211

Parameters	211
To user profile (TOUSER)	211
For user profile (FORUSER)	211
Examples	212
Error messages	212

Grant Workstation Object Aut (GRTWSOAUT) 213

Parameters	213
Workstation object type (WSOTYPE)	214
Users (USER)	215
Authority (AUT)	216
Authorization list (AUTL)	217
Reference workstation object (REFWSO)	217
Examples	217
Error messages	218

Hold Communications Device (HLDCMNDEV) 219

Parameters	219
Device (DEV)	219
Option (OPTION)	220

Examples	220
Error messages	220

Hold Distribution Queue (HLDDSTQ) 223

Parameters	223
Distribution queue (DSTQ)	223
Priority (PTY)	223
Examples	224
Error messages	224

Hold Job (HLDJOB) 227

Parameters	227
Job name (JOB)	227
Hold spooled files (SPLFILE)	228
Duplicate job option (DUPJOB OPT)	228
Examples	228
Error messages	229

Hold Job Queue (HLDJOBQ) 231

Parameters	231
Job queue (JOBQ)	231
Examples	231
Error messages	232

Hold Job Schedule Entry (HLDJOBSCDE) 233

Parameters	233
Job name (JOB)	233
Entry number (ENTRYNBR)	234
Examples	234
Error messages	234

Hold Output Queue (HLDOUTQ) 237

Parameters	237
Output queue (OUTQ)	237
Examples	237
Error messages	238

Hold Reader (HLDRDR) 239

Parameters	239
Reader (RDR)	239
Examples	239
Error messages	239

Hold Spooled File (HLDSPLF) 241

Parameters	241
Spooled file (FILE)	241
Job name (JOB)	242
Spooled file number (SPLNBR)	242
Job system name (JOBSYSNAME)	242
Spooled file created (CRTDATE)	243
Select files for (SELECT)	243
ASP device (ASPDEV)	244
When to hold file (OPTION)	245
Examples	245
Error messages	245

Hold Writer (HLDWTR) 247

Parameters	247
----------------------	-----

Writer (WTR)	247
When to hold writer (OPTION)	247
Examples	248
Error messages	248

If (IF) 249

Parameters	250
Condition (COND)	250
Command (THEN)	250
Examples	251
Error messages	252

Install Linux Server (INSLNXSVR) . . . 253

Parameters	254
Network server description (NWSD)	256
Resource name (RSRCNAME)	257
Linux server distribution (LNXSVRDST)	257
Linux source directory (LNXSRCDIR)	257
TCP/IP port configuration (TCPPORTCFG)	258
Virtual Ethernet port (VRTETHPORT)	259
TCP/IP local domain name (TCPDMNNAME)	259
TCP/IP name server system (TCPNAMSVR)	260
Server storage space sizes (SVRSTGSIZE)	260
Storage space ASP (SVRSTGASP)	261
Server storage ASP device (STGASPDEV)	261
Language version (LNGVER)	262
Synchronize date and time (SYNCTIME)	262
Restricted device resources (RSTDDEVRSC)	262
Shutdown timeout (SHUTDTIMO)	263
Activation timer (ACTTMR)	263
Communications message queue (CMNMSGQ)	263
Storage path (STGPTH)	264
Virtual Ethernet path (VRTETHPTH)	264
Shutdown TCP port (SHUTDPORT)	264
Virtual Ethernet control port (VRTETHCTLP)	265
Remote system NWSCFG (RMTNWSCFG)	265
Service processor NWSCFG (SPNWSCFG)	265
Connection security NWSCFG (CNNNWSCFG)	265
Default IP security rule (DFTSECRULE)	266
IP security rule (IPSECRULE)	266
Initialize service processor (INZSP)	266
Enable unicast (ENBUNICAST)	267
Enclosure identifier (EID)	268
Service processor name (SPNAME)	268
SP internet address (SPINTNETA)	269
SP authentication (SPAUT)	269
SP certificate identifier (SPCERTID)	269
Remote system identifier (RMTSYSID)	270
Delivery method (DELIVERY)	271
CHAP authentication (CHAPAUT)	271
Boot device ID (BOOTDEVID)	272
Dynamic boot options (DYNBOOTOPT)	272
Remote interfaces (RMTIFC)	273
Text 'description' (TEXT)	275
Configuration file (CFGFILE)	275
Virtual PTP Ethernet port (VRTPTPPORT)	276
Examples	276
Error messages	277

Install Program Temporary Fix (INSPTF) 279

Parameters	279
Product description (LICPGM)	280
Device (DEV)	280
PTF apply type (INSTYP)	280
PTF omit list (OMIT)	281
HIPER PTFs only (HIPER)	281
End of media option (ENDOPT)	282
Restart type (RESTART)	282
Prompt for media (PMTMED)	283
Copy PTFs (CPYPTF)	283
Examples	283
Error messages	284

Install Windows Server (INSWNTSVR) 285

Parameters	286
Network server description (NWSD)	289
Installation type (INSTYPE)	290
Resource name (RSRCNAME)	290
Windows server version (WNTVER)	290
Windows source directory (WNTSRCDIR)	291
Install option (OPTION)	291
TCP/IP port configuration (TCPPORTCFG)	292
Virtual Ethernet port (VRTETHPORT)	293
TCP/IP local domain name (TCPDMNNAME)	294
TCP/IP name server system (TCPNAMSVR)	294
Server message queue (MSGQ)	295
Event log (EVTLOG)	295
Server storage space sizes (SVRSTGSIZE)	296
Storage space ASP (SVRSTGASP)	297
Server storage ASP device (STGASPDEV)	297
Convert to NTFS (CVTNTFS)	298
To workgroup (TOWRKGRP)	298
To domain (TODMN)	299
Full Name (FULNAM)	300
Organization (ORG)	300
Language version (LNGVER)	300
Synchronize date and time (SYNCTIME)	300
Propagate domain user (PRPDMNUSR)	301
Windows license key (WNTLICKEY)	301
License mode (LICMODE)	301
Restricted device resources (RSTDDEVRSC)	302
Shutdown timeout (SHUTDTIMO)	302
Activation timer (ACTTMR)	303
Communications message queue (CMNMSGQ)	303
Storage path (STGPTH)	303
Virtual Ethernet path (VRTETHPTH)	304
Shutdown TCP port (SHUTDPORT)	305
Virtual Ethernet control port (VRTETHCTLP)	305
Remote system NWSCFG (RMTNWSCFG)	305
Service processor NWSCFG (SPNWSCFG)	305
Connection security NWSCFG (CNNNWSCFG)	306
Default IP security rule (DFTSECRULE)	306
IP security rule (IPSECRULE)	307
Initialize service processor (INZSP)	307
Enable unicast (ENBUNICAST)	308
Enclosure identifier (EID)	308
Service processor name (SPNAME)	309
SP internet address (SPINTNETA)	309

SP authentication (SPAUT)	310
SP certificate identifier (SPCERTID)	310
Remote system identifier (RMTSYSID)	311
Delivery method (DELIVERY)	312
CHAP authentication (CHAPAUT)	312
Boot device ID (BOOTDEVID)	313
Dynamic boot options (DYNBOOTOPT)	314
Remote interfaces (RMTIFC)	315
Text 'description' (TEXT)	317
Keyboard layout (KBDTYPE)	317
Configuration file (CFGFILE)	317
Cluster name (CLU)	318
Cluster configuration (CLUCFG)	318
Virtual PTP Ethernet port (VRTPTPPORT)	319
Examples	320
Error messages	322

Initialize DLFM (INZDLFM)	323
Parameters	323
Clear existing databases (CLEARDB)	323
Examples	323
Error messages	323

Initialize Distribution Queue (INZDSTQ)	325
Parameters	325
Distribution queue (DSTQ)	326
Clear queue entries (CLEAR)	326
Examples	326
Error messages	327

Initialize NWS Configuration (INZNWSCFG)	329
Parameters	329
Network server configuration (NWSCFG)	329
Processing option (OPTION)	330
SP authentication (SPAUT)	330
Examples	331
Error messages	331

Initialize Optical (INZOPT)	333
Parameters	333
Volume identifier (VOL)	333
New volume identifier (NEWVOL)	334
Device (DEV)	334
Volume full threshold (THRESHOLD)	334
Check for an active volume (CHECK)	334
End of media option (ENDOPT)	335
Clear (CLEAR)	335
Volume type (TEXT)	335
Volume type (TYPE)	335
Coded character set ID (CCSID)	336
Media format (MEDFMT)	336
Examples	336
Error messages	337

Initialize iSeries Access (INZPCS)	339
Error messages for INZPCS	339
Parameters	339

Keyboard type (KBDTYPE)	340
ASCII code page number (ASCII)	342
EBCDIC code page number (EBCDIC)	342
Language feature code (LANGUAGE)	343
Examples	344
Error messages	345

Initialize Physical File Mbr (INZPFM)	347
Parameters	347
Physical file (FILE)	347
Member (MBR)	348
Initialize records as (RECORDS)	348
Total number of records (TOTRCDS)	348
Examples	348
Error messages	349

Initialize System (INZSYS)	351
Parameters	351
Message queue (MSGQ)	351
Examples	352
Error messages	352

Initialize Tape (INZTAP)	353
Parameters	353
Device (DEV)	353
New volume identifier (NEWVOL)	353
New owner identifier (NEWOWNID)	354
Volume identifier (VOL)	354
Check for active files (CHECK)	355
Tape density (DENSITY)	355
Code (CODE)	359
End of tape option (ENDOPT)	359
Clear (CLEAR)	360
Examples	360
Error messages	360

Iterate (ITERATE)	363
Parameters	363
Command label (CMDLBL)	363
Examples	363
Error messages	364

Leave (LEAVE)	365
Parameters	365
Command label (CMDLBL)	365
Examples	366
Error messages	366

Link/Unlink Data Definition (LNKDTADFN)	367
Parameters	367
Option (OPTION)	367
Data base file (FILE)	368
Data dictionary (DTADCT)	368
File definition (DFN)	368
Creation date (CRTDATE)	368
Examples	369
Error messages	369

Load or Unload Image Catalog (LODIMGCLG)	371
Parameters	371
Image catalog (IMGCLG)	372
Virtual device (DEV)	372
Option (OPTION)	372
Write protect (WRTPTC)	372
Examples	373
Error messages	374

Load/Unload/Mount IMGCLG Entry (LODIMGCLGE)	375
Parameters	375
Image catalog (IMGCLG)	376
Image catalog index (IMGCLGIDX)	376
Option (OPTION)	376
Image catalog volume (VOL)	376
Examples	376
Error messages	377

Load Optical Firmware (LODOPTFMW)	379
Parameters	379
From stream file (FROMSTMF)	379
Device type (TYPE)	379
Device model (MODEL)	379
Examples	380
Error messages	380

Load Program Temporary Fix (LODPTF)	381
Parameters	381
Product (LICPGM)	381
Device (DEV)	382
PTF numbers to select (SELECT)	382
PTF numbers to omit (OMIT)	382
Superseded PTFs (SPRPTF)	382
Release (RLS)	383
Sequence number (SEQNBR)	383
End of media option (ENDOPT)	383
Path identifier (PATHID)	384
Save file (SAVF)	384
Copy PTF cover letter (COVER)	385
Copy PTF save file (CPYSAVF)	385
Examples	385
Error messages	386

Load Q/A Database (LODQSTDB)	389
Parameters	389
Q/A database (QSTDB)	389
Lib containing Q/A database (LIB)	390
Examples	390
Error messages	390

Load and Run (LODRUN)	391
Parameters	393
Device (DEV)	393
Sequence number (SEQNBR)	393
Volume identifier (VOL)	394
Directory (DIR)	394

Examples	394
Error messages	395

Create Directory (MD)	397
Parameters	398
Directory (DIR)	398
Public authority for data (DTAAUT)	398
Public authority for object (OBJAUT)	399
Auditing value for objects (CRTOBJAUD)	400
Scanning option for objects (CRTOBJSCAN)	400
Restricted rename and unlink (RSTDRNMUNL)	401
Examples	401
Error messages	401

Create Directory (MKDIR)	403
Parameters	404
Directory (DIR)	404
Public authority for data (DTAAUT)	404
Public authority for object (OBJAUT)	405
Auditing value for objects (CRTOBJAUD)	406
Scanning option for objects (CRTOBJSCAN)	406
Restricted rename and unlink (RSTDRNMUNL)	407
Examples	407
Error messages	407

Monitor Message (MONMSG)	409
Parameters	410
Message identifier (MSGID)	410
Comparison data (CMPDTA)	411
Command to execute (EXEC)	411
Examples	411
Error messages	412

Add Mounted FS (MOUNT)	413
Parameters	413
Type of file system (TYPE)	414
File system to mount (MFS)	414
Directory to mount over (MNTOVRDIR)	414
Mount options (OPTIONS)	415
Coded character set ID (CCSID)	417
Code page (CODEPAGE)	417
Examples	418
Error messages	419

Move Object (MOV)	421
Parameters	422
Object (OBJ)	422
To directory (TODIR)	422
To object (TOOBJ)	422
From CCSID (FROMCCSID)	423
To CCSID (TOCCSID)	423
Data Format (DTAFMT)	424
From Code Page (FROMCODPAG)	424
To Code Page (TOCODEPAGE)	425
Examples	425
Error messages	426

Move Document (MOVDOC)	427
Parameters	427

From document (FROMDOC)	427
From folder (FROMFLR)	428
To folder (TOFLR)	428
Rename (RENAME)	428
System object name (SYSOBJNAM)	428
Examples	429
Error messages	429

Move Object (MOVE) 431

Parameters	432
Object (OBJ)	432
To directory (TODIR)	432
To object (TOOBJ)	432
From CCSID (FROMCCSID)	433
To CCSID (TOCCSID)	433
Data Format (DTAFMT)	434
From Code Page (FROMCODPAG)	434
To Code Page (TOCODEPAGE)	435
Examples	435
Error messages	436

Move Object (MOV OBJ) 437

Parameters	438
Object (OBJ)	439
Object type (OBJTYPE)	439
To library (TOLIB)	439
From ASP device (ASPDEV)	440
To ASP device (TOASPDEV)	440
Examples	441

Error messages	441
--------------------------	-----

Merge Message Catalog (MRGMSGCLG) 445

Parameters	445
Message catalog name (CLGFILE)	445
Source file path name (SRCFILE)	446
Message catalog CCSID (CLGCCSID)	446
Text 'description' (TEXT)	446
Source file CCSID (SRCCCSID)	446
Public authority for data (DTAAUT)	447
Public authority for object (OBJAUT)	448
Examples	448
Error messages	448

Merge Message File (MRGMSGF) 449

Parameters	449
From message file (FROMMSGF)	450
To message file (TOMSGF)	450
Replaced message file (RPLMSGF)	450
Message IDs to select (SELECT)	451
Message IDs to omit (OMIT)	451
Examples	451
Error messages	452

Appendix. Notices 455

Trademarks	456
Terms and conditions	457

End Job (ENDJOB)

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

Parameters
Examples
Error messages

The End Job (ENDJOB) command ends the specified job and any associated inline data files. The job can be on a job queue, it can be active, or it can have already completed running.

You can specify that the application program is 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

Before ending the job, you should verify that no logical unit of work is in an in doubt state due to a two-phase commit operation that is in progress. If it is, then the value of the Action if ENDJOB commitment option can greatly impact the ENDJOB processing. For example, if the Action if ENDJOB commitment option is the default value of WAIT, this job will be held up and will not complete its end of job processing until the commitment control operation is completed. This ensures database integrity on all related systems. For specific instructions on how to determine these conditions, and for a description of all the impacts of ending this job under these conditions, see the Backup and Recovery book.

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 ended, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. This restriction is enforced even when ending the current job.

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	Single values: * Other values: <i>Qualified job name</i>	Required, Positional 1
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
OPTION	How to end	* <u>CNTRLD</u> , *IMMED	Optional, Positional 2
DELAY	Controlled end delay time	1-999999, <u>30</u>	Optional, Positional 3
SPLFILE	Delete spooled files	* <u>NO</u> , *YES	Optional, Positional 4

Keyword	Description	Choices	Notes
LOGLMT	Maximum log entries	<i>Integer</i> , *SAME, *NOMAX	Optional
ADLINTJOBS	Additional interactive jobs	*NONE, *GRPJOB, *ALL	Optional
DUPJOB OPT	Duplicate job option	*SELECT, *MSG	Optional

Top

Job name (JOB)

Specifies the qualified job name of the job to be ended.

This is a required parameter.

Single values

- * The job from which this command is run is ended. Specifying this value is the only way to end the current job.

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 run.

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 provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOB OPT)** parameter for more information.

Top

How to end (OPTION)

Specifies whether the job ends immediately or in a controlled manner that lets the application program perform end-of-job processing. In either case, the system performs certain job cleanup processing.

*CNTRL D

The job ends in a controlled manner. This allows the program 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 on the DELAY parameter to complete cleanup before the job is ended.

*IMMED

The job ends immediately and the system performs end-of-job cleanup. System cleanup can take from a brief amount of time to several minutes. 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 QENDJOB LMT system value specifies the time limit. Other than by handling the SIGTERM signal, the program that is running is 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.

Note: When a SIGTERM signal handler is running during the immediate ending of a job, an ENDJOB command with OPTION(*IMMED) can be used to end the SIGTERM signal handler. This is only allowed if the SIGTERM signal handler has already had at least two minutes to run.

Top

Controlled end delay time (DELAY)

Specifies the amount of time (in seconds) allowed for the job to complete its cleanup processing during a controlled end. If the cleanup is not completed before the end of the delay time, the job is ended immediately. (Only system cleanup is performed.)

The delay time does not start until the job becomes active if the job is suspended because of one of the following conditions:

- The system request option 1 is selected.
- The job is held by the Hold Job (HLDJOB) command.
- The job is transferred by the Transfer Secondary Job (TFRSECJOB) command.
- The job is transferred by the Transfer to Group Job (TFRGRPJOB) command.

Note: This parameter is valid only when OPTION(*CNTRLD) is specified.

30 A maximum delay time of 30 seconds is allowed for cleanup before the job ends.

1-999999

Specify the maximum amount of delay time (in seconds) before the job ends.

Top

Delete spooled files (SPLFILE)

Specifies whether spooled output files created by this job are kept for normal processing or deleted. Regardless of whether the spooled files are deleted, the job logs are kept.

***NO** The spooled output files created by the job being ended are kept 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 job being ended and which are on output queues in the library name space of the thread issuing this command are deleted. The job log is not deleted. If the job has already ended and the spooled file action for the job is to detach the spooled files, the End Job (ENDJOB) command will not find the job and the spooled files will not be deleted.

Top

Maximum log entries (LOGLMT)

Specifies the maximum number of entries in the message queue of the job 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 a job that ends.

The value specified on this parameter can change the logging limit of the job even if the job is already ending or the job has already ended. The following are examples of how the logging limit can be changed:

1. If the value specified is greater than the number of messages written at the time the command is issued, messages continue to be written until the new limit is reached.
2. If the value specified is less than the number of messages already written to the spooled file, a message indicating that the limit has been reached is immediately put in the spooled file as the last entry. The remaining messages on the queue are ignored.
3. If 0 (zero) is specified before any messages are written to the spooled file, no job log is produced for the job that is ending. If the job has already ended and the job log has not yet been produced, the job log is removed regardless of the value of the job log output (LOGOUTPUT) job attribute. For more information on removing pending job logs, refer to the Remove Pending Job Log (QWTRMVJL) API.

***SAME**

The message logging limit does not change. If the logging limit does not change for this job on a previous command, *NOMAX is the value used by the system.

***NOMAX**

There is no limit to the number of messages logged; **all** messages on the job message queue are written to the job log.

integer-number

Specify the maximum number of messages that can be written to the job log.

Top

Additional interactive jobs (ADLINTJOBS)

Specifies whether the additional interactive jobs associated with the job specified in the **Job name (JOB)** parameter are ended.

***NONE**

Only the job specified in the JOB parameter is ended.

***GRPJOB**

If the job specified in the JOB parameter is a group job, all group jobs associated with the group are ended. If the job is not a group job, the job specified in the JOB parameter is ended.

***ALL** All interactive jobs running on the workstation associated with the job specified in the JOB parameter are ended. This includes group jobs and secondary jobs.

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: Ending a Job Immediately

```
ENDJOB JOB(JOB1) OPTION(*IMMED) SPLFILE(*YES)
```

This command ends a job named JOB1 immediately. Spooled output produced by the job is deleted; the job log is saved.

Example 2: Saving Spooled Output

```
ENDJOB JOB(001234/XYZ/JOB2) OPTION(*CNTRLD)
      DELAY(50) SPLFILE(*NO)
```

This command ends a job named 001234/XYZ/JOB2. Spooled output is saved for normal processing by the spooling writer. The job has 50 seconds to perform any cleanup routines, after which it is ended immediately.

Example 3: Removing a Pending Job Log

```
ENDJOB JOB(543210/ABCDE/JOB3) LOGLMT(0)
```

This command removes the pending job log for a completed job named 543210/ABCDE/JOB3. This has the same effect as the Remove Pending Job Log (QWTRMVJL) API.

Example 4: Ending the Current Job

```
ENDJOB JOB(*) OPTION(*IMMED)
```

This command ends the job in which the command is issued. To exit from a SIGTERM signal handling procedure after the required cleanup has been performed, immediately end the current job. Control may be returned to the command issuer, even though the immediate option has been used. To handle this situation, the program which issued this command should return.

Top

Error messages

*ESCAPE Messages

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.

CPF1351

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

CPF1352

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

CPF135D
ENDJOB OPTION(*IMMED) not allowed at this time.

CPF1360
&3/&2/&1 already ending because of ENDJOBABN.

CPF1361
Job &3/&2/&1 already ending with *IMMED option.

CPF1362
Job &3/&2/&1 has completed.

CPF1363
Job &3/&2/&1 is already ending *CNTRLD.

CPF8172
Spool control block for job &10/&9/&8 damaged.

Top

End Job Abnormal (ENDJOBABN)

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

Parameters
Examples
Error messages

The End Job Abnormal (ENDJOBABN) command ends a job that cannot be ended successfully by running the End Job (ENDJOB) command with *IMMED specified for the **How to end (OPTION)** parameter. The ENDJOBABN command cannot be issued against a job until 10 minutes have passed following the request for immediate ending. This allows sufficient time for normal job ending functions to be attempted.

When the ENDJOBABN command is issued, most of the end-of-job processing is bypassed (including spooling of the job log, the end of job display for interactive jobs, and the end-of-job processing for the specific functions that are being performed). The part of the end-of-job processing that is attempted is allowed only five minutes to complete. If it does not do so in five minutes, the job is forced to end at that point. Because some of the job cleanup is not performed, the ENDJOBABN command should only be used when a job that is in the process of immediate ending does not finish ending and resources in use by the job are needed by another job or by the system. When the ENDJOBABN command is used, some resources in use by the ended job may be left unavailable until the next IPL.

Use of the ENDJOBABN command causes the next system end to be marked as ABNORMAL. Certain system functions are then called during the subsequent IPL to clear up conditions that may have resulted from running the ENDJOBABN command. This does not, however, cause any machine recovery functions to be called, nor do any access paths need to be rebuilt. Some storage in use by the job may become unavailable after the ENDJOBABN command is run and that available storage can be reclaimed by using the Reclaim Storage (RCLSTG) command.

Bypassing the job log writing process causes the job to have the status of JOBLOG PENDING (as shown on the DSPJOB status attributes display) after it has been ended with the ENDJOBABN command. The job log writing is not performed until the next IPL. However, the contents of the job log can be printed or shown by using the Display Job Log (DSPJOBLOG) command.

When the ENDJOBABN command is run, the following functions are performed successfully:

- Journaling entries
- Commitment control

Before ending the job abnormally, you should verify that no logical unit of work is in an in doubt state due to a two-phase commit operation that is in progress. If it is, then pending committable changes at this system will not be committed or rolled back. Therefore, database integrity may not be maintained on all related systems. For specific instructions on how to determine these conditions, and for a description of all the impacts of ending this job abnormally under these conditions, see the Commitment control article in the Information Center.

- Making database files available for use by others
- Releasing file locks

This command fails to end a job or takes more than five minutes to do so in the following situations:

- When the job runs under a subsystem monitor that is hung, is abnormally slow, or has ended abnormally (the subsystem monitor performs part of the ending function).
- When the machine interface (MI) instruction running in the job is hung or is abnormally slow. The job cannot end until the MI instruction that is currently running completes or reaches a point of interruption.

Restrictions:

1. 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 ended, 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.
2. After the ENDJOBABN command is run, subsequent ENDJOBABN commands cannot be issued against the job.
3. Users cannot end a reader, writer, subsystem monitor, or system job.
4. Users cannot run the ENDJOBABN command until ten minutes after immediate ending of the job is started. Immediate ending of the job is started in the following ways:
 - When the End Job (ENDJOB) command with OPTION(*CNTRLD) is specified and the delay time ends.
 - When the ENDJOB command with OPTION(*IMMED) is issued.
 - When the End Subsystem (ENDSBS) command with OPTION(*CNTRLD) is issued against the subsystem in which the job is running and the delay time ends.
 - When the ENDSBS command with OPTION(*IMMED) is issued against the subsystem in which the job is running.
 - When the End System (ENDSYS) command with OPTION(*IMMED) is issued, or OPTION(*CNTRLD) is issued and the delay time ends.
 - When the Power Down System (PWRDWN SYS) command with OPTION(*IMMED) is issued.
5. If the job defines a handler for the asynchronous signal SIGTERM, the immediate ending of the job was delayed to allow the SIGTERM signal handler to run. For more information, refer to system value QENDJOBLMT. An ENDJOBABN command is not allowed while the SIGTERM signal handler is running. If the SIGTERM signal handler has run for at least 2 minutes, use ENDJOB command with OPTION(*IMMED) to stop the SIGTERM signal handler.

Top

Parameters

Keyword	Description	Choices	Notes
JOB	Job name	<i>Qualified job name</i>	Required, Positional 1
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
DUPJOB OPT	Duplicate job option	*SELECT , *MSG	Optional

Top

Job name (JOB)

Specifies the qualified job name of the job to be ended.

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 run.

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 provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOB OPT)** parameter for more information.

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

```
ENDJOBABN JOB(000310/SMITH/PAYROLL)
```

This command ends the batch job 000310/SMITH/PAYROLL after the failure of an earlier attempt to end it with the ENDJOB command. The ENDJOBABN command can be issued only after waiting at least ten minutes for the job to end after issuing the ENDJOB command.

Top

Error messages

*ESCAPE Messages

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.

CPF1351

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

CPF1359

ENDJOBABN not allowed at this time for job &3/&2/&1.

CPF1360

&3/&2/&1 already ending because of ENDJOBABN.

CPF1362

Job &3/&2/&1 has completed.

[Top](#)

End Journal (ENDJRN)

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

Parameters
Examples
Error messages

The End Journal (ENDJRN) command is used to end the journaling of changes for an object or list of objects. The object types which are supported through this interface are Data Areas (*DTAARA), Data Queues (*DTAQ), Byte Stream Files (*STMF), Directories (*DIR), and Symbolic Links (*SYMLNK). Only objects of type *STMF, *DIR, or *SYMLNK that are in the "root" (/), QOpenSys, and user-defined file systems are supported.

All objects of the supported types that are currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

1. Access Paths - End Journal Access Path (ENDJRNAP)
2. Physical Files - End Journal Physical File (ENDJRNPF)
3. Other Objects - End Journal Object (ENDJRNOBJ)

Restrictions:

- Objects specified on the command cannot be in use for any reason at the time the command is running.
- If OBJ(*ALL) is specified, a journal name must be specified (JRN parameter).
- If a journal name and a list of object names are specified, all objects must be currently journaled to the indicated journal.
- The specified journal must be a local journal.
- At least one of parameter OBJ or OBJFID must be specified.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Name	<i>Path name</i> , <u>*</u>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
OBJFID	File identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
SUBTREE	Directory subtree	<u>*NONE</u> , *ALL	Optional
PATTERN	Name pattern	Values (up to 20 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value</i> , <u>*</u>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
JRN	Journal	<i>Path name</i> , <u>*OBJ</u>	Optional

Top

Objects (OBJ)

Specifies a maximum of 300 object path names for which changes will no longer be journaled. Only objects whose path name identifies an object of type *STMF, *DIR, *SYMLNK, *DTAARA or *DTAQ are supported.

Single values

***ALL** All objects of the supported type that are currently being journaled to the indicated journal are to stop having their changes journaled. If *ALL is specified parameter OBJFID must not be specified.

Element 1: Name

'object-path-name'

Specify the path name of the object for which changes are no longer journaled.

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

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

Element 2: Include or omit

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

*INCLUDE

The objects that match the object name pattern are to stop having their changes journaled unless overridden by an *OMIT specification.

*OMIT

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

Top

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) for which changes are no longer journaled. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies on object of type *STMF, *DIR, *SYMLNK, *DTAARA or *DTAQ are supported.

file-identifier

Objects identified with the FID are no longer journaled.

Top

Directory subtree (SUBTREE)

Specifies whether the objects in directory subtrees are to stop having their changes journaled.

Note: This parameter is ignored unless object-path-name is a directory object.

Note: This parameter is ignored if the OBJFID parameter is specified.

*NONE

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

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

Top

Name pattern (PATTERN)

Specifies a maximum of 20 patterns to be used to include or omit objects for the end journal operation.

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

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

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

Note: This parameter is ignored if the OBJFID parameter is specified.

Element 1: Name pattern

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

name-pattern

Specify the pattern to either include or omit objects for the end journal operation. Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern.

Element 2: Include or omit

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

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

*INCLUDE

The objects that match the object name pattern are to stop having their changes journaled unless overridden by an *OMIT specification.

***OMIT**

The objects that match the object name pattern are not to be included with the objects that are to

stop having their changes journaled. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Journal (JRN)

Specifies the journal to which changes are currently being journaled.

***OBJ** The journal is determined by the system from the specified object path name or object file identifier.

journal-path-name

Specify the path name of the journal to which changes are currently being journaled.

Top

Examples

Example 1: End All Non-Database Journaling

```
ENDJRN OBJ(*ALL) JRN('/qsys.lib/mylib.lib/myjrn.jrn')
```

This command stops the journaling of all changes to all objects of type *DIR, *STMF, *SYMLNK, *DTAARA and *DTAQ to journal /qsys.lib/mylib.lib/myjrn.jrn.

Example 2: End Journaling with Omit of Directory

```
ENDJRN OBJ('/mypath' *INCLUDE) ('mypath/myobject' *OMIT))
```

This command stops the journaling of all changes to all first-level objects in directory /mypath except object /mypath/myobject. Object /mypath/myobject will continue to be journaled.

Example 3: End Journaling with Pattern Selection

```
ENDJRN OBJ('/mypath' *INCLUDE) ('mypath/mysubdir' *OMIT))  
SUBTREE(*ALL) PATTERN('*.*txt' *INCLUDE))
```

This command stops the journaling of all changes to all objects in directory /mypath of type *DIR, *STMF, and *SYMLNK that match pattern '*.txt'. Any objects within directory /mypath/mysubdir will continue to be journaled.

Example 4: End Journaling using File Identifiers

```
ENDJRN OBJFID(00000000000000000007E09BDB000000009  
0000000000000000009E09BDB00000000A)
```

This command stops the journaling of all changes to the objects of type *DIR, *STMF, *SYMLNK, *DTAARA or *DTAQ represented by the specified file identifiers.

Example 5: End Journaling on a set of Data Areas

```
ENDJRN OBJ('/qsys.lib/mylib.lib/mydata*.dtaara'))  
JRN('/qsys.lib/mylib.lib/myjrn.jrn')
```

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

Top

Error messages

*ESCAPE Messages

CPFA0D4

File system error occurred. Error number &1.

CPF700B

&1 of &2 objects have ended journaling.

CPF705A

Operation failed due to remote journal.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.

Top

End Journal Access Path (ENDJRNAP)

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

Parameters
Examples
Error messages

The End Journal Access Path (ENDJRNAP) command is used to end the journaling of the access paths of a journaled file.

All access paths currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

- Integrated file system objects - End Journal (ENDJRN)
- Physical files - End Journal Physical File (ENDJRNPF)
- Other objects - End Journal Object (ENDJRNOBJ)

Restrictions:

- The access paths for the files specified on the command cannot be in use for any reason at the time the command is running.
- Overrides are not applied to the files listed in the FILE parameter.
- If FILE(*ALL) is specified, a journal name must be specified.
- If a journal name and a list of file names are specified, then all the access paths for the listed files must be currently journaled to the indicated journal.
- Journaling entries for any physical file does not end by the running of this command.
- The specified journal must be a local journal.

Top

Parameters

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

Top

Journalled file (FILE)

Specifies a maximum of 50 database files for which the journaling of access paths are to be ended.

This is a required parameter.

Single values

***ALL** All current journaling of access paths to the indicated journal are ended.

Qualifier 1: Journalled file

file-name

Specify the name and library of the database file for which access paths for the journal entry are ended.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Journal (JRN)

Specifies the journal to which journaling of the access paths for the indicated files are being ended.

Single values

***FILE** The journal name is determined by the system from the specified file names.

Qualifier 1: Journal

journal-name

Specify the name of the journal.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Examples

```
ENDJRNAP FILE(MYLIB/MYFILE)
```

This command ends the journaling for all access paths of the file MYFILE in the library MYLIB.

Error messages

*ESCAPE Messages

CPF6972

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

CPF7008

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

CPF703C

DDL transaction prevents journaling operation.

CPF703D

DDL transaction prevents journaling operation.

CPF703E

DDL transaction prevents journaling operation.

CPF7032

ENDJRNPf or ENDJRNAP command failed.

CPF7033

Start or end journaling failed for member &3.

CPF7034

Logical damage of file &1 in &2.

CPF705A

Operation failed due to remote journal.

CPF708D

Journal receiver found logically 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.

CPF9812

File &1 in library &2 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.

End Journal Object (ENDJRNOBJ)

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

Parameters
Examples
Error messages

The End Journal Object (ENDJRNOBJ) command is used to end journaling of changes for an object or list of objects.

All objects, of object types *DTAARA and *DTAQ, that are currently being journaled to a specific journal may also have journaling stopped.

For other ways to end journaling see the following commands:

- Access paths - End Journal Access Path (ENDJRNAP)
- Integrated file system objects - End Journal (ENDJRN)
- Physical files - End Journal Physical File (ENDJRNPf)

Restrictions:

- Objects specified on the command cannot be in use for any reason at the time the command is running.
- If OBJ(*ALL) or OBJTYPE(*ALL) is specified, a journal name must be specified (JRN parameter).
- If a journal name and a list of object names are specified, all objects must be currently journaled to the indicated journal.
- The specified journal must be a local journal.

Top

Parameters

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

Top

Object (OBJ)

Specifies a maximum of 300 objects for which changes are no longer to be journaled.

This is a required parameter.

Single values

***ALL** All objects of the specified object types that are currently being journaled to the indicated journal are to stop having their changes journaled.

Qualifier 1: Object

object-name

Specify the name of the object for which journaling is to be ended.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Object type (OBJTYPE)

Specifies the object type for which journaling is to be ended.

This is a required parameter.

***DTAARA**

Data area objects are to have their journaling ended.

***DTAQ**

Data queue objects are to have their journaling ended.

***ALL** All objects of the object types that are supported on this command are to have their journaling ended.

Note: If OBJTYPE(*ALL) is specified, then OBJ(*ALL) must also be specified.

Top

Journal (JRN)

Specifies the qualified name of the journal to which changes in the objects are currently being journaled.

Single values

***OBJ** The journal is determined by the system from the specified object name and object type.

Qualifier 1: Journal

journal-name

Specify the name of the journal to which the indicated objects are currently being journaled.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Examples

Example 1: End Journaling All Data Areas and Data Queues

```
ENDJRNOBJ OBJ(*ALL) OBJTYPE(*ALL) JRN(MYLIB/MYJRN)
```

This command stops journaling all changes to all objects of type *DTAARA and *DTAQ to journal MYJRN in library MYLIB.

Example 2: End Journaling for Specific Data Area

```
ENDJRNOBJ OBJ(DTALIB/MYDTAARA) OBJTYPE(*DTAARA)
```

This command stops the journaling of all changes to data area MYDTAARA in library DTALIB.

Top

Error messages

*ESCAPE Messages

CPF700B

&1 of &2 objects have ended journaling.

CPF705A

Operation failed due to remote journal.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9873

ASP status is preventing access to object.

CPF9875

Resources exceeded on ASP &1.

[Top](#)

End Journal Physical File (ENDJRNPf)

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

Parameters
Examples
Error messages

The End Journal Physical File (ENDJRNPf) command is used to end journaling of changes for a specific physical file and all of its members.

All physical files currently being journaled to a specific journal may also have journaling stopped.

When the file for which journaling is ended is a distributed file, an attempt is made to distribute the ENDJRNPf command if journaling was successfully ended locally. Even if the distribution request fails, the local file is not journaled. In addition, if a journal and file name are specified, and the file is distributed, an attempt to distribute the ENDJRNPf request is made even if the file is not journaled locally.

For other ways to end journaling see the following commands:

- Access paths - End Journal Access Path (ENDJRNPf)
- Integrated file system objects - End Journal (ENDJRNPf)
- Other objects - End Journal Object (ENDJRNPf)

Restrictions:

- Members in the files specified on the command cannot be in use for any reason at the time the command is running.
- Overrides are not applied to the files listed in the FILE parameter.
- If FILE(*ALL) is specified, a journal name must be specified.
- If a journal name and a list of file names are specified, all files must be currently journaled to the indicated journal.
- The specified journal must be a local journal.
- In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	Journaled physical file	Single values: *ALL Other values (up to 50 repetitions): <i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Journaled physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
JRN	Journal	Single values: *FILE Other values: <i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	

Top

Journalled physical file (FILE)

Specifies a maximum of 50 physical files for which changes will no longer be journalled.

This is a required parameter.

Single values

***ALL** All physical files currently being journalled to the specified journal no longer have their changes journalled.

Qualifier 1: Journalled physical file

file-name

Specify the name of the physical database file for which changes will no longer be journalled.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Journal (JRN)

Specifies the name of the journal to which changes in the indicated files are currently being journalled.

Single values

***FILE** The journal is determined by the system from the specified file names.

Qualifier 1: Journal

journal-name

Specify the name of the journal to which changes in the specified files are currently being journalled.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Examples

ENDJRNPF FILE(MYLIB/MYFILE)

This command stops the journaling of all changes to all members of file MYFILE in library MYLIB. Changes made after this command is run are not journaled.

Top

Error messages

*ESCAPE Messages

CPF6959

Object &1 is not currently journaled.

CPF6970

Access paths built over file &1 are being journaled.

CPF7002

File &1 in library &2 not a physical file.

CPF703B

Implicit end of access path journaling failed.

CPF703C

DDL transaction prevents journaling operation.

CPF703D

DDL transaction prevents journaling operation.

CPF703E

DDL transaction prevents journaling operation.

CPF7031

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

CPF7032

ENDJRNPF or ENDJRNAP command failed.

CPF7033

Start or end journaling failed for member &3.

CPF7034

Logical damage of file &1 in &2.

CPF704C

Journaling ended locally but distributed requests failed.

CPF704D

ENDJRNPF command failed.

CPF705A

Operation failed due to remote journal.

CPF708D

Journal receiver found logically 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.
- CPF9810**
Library &1 not found.
- CPF9812**
File &1 in library &2 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9825**
Not authorized to device &1.
- CPF9830**
Cannot assign library &1.
- CPF9873**
ASP status is preventing access to object.
- CPF9875**
Resources exceeded on ASP &1.

Top

End Line Recovery (ENDLINRCY)

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

Parameters
Examples
Error messages

The End Line Recovery (ENDLINRCY) command ends automatic error recovery procedures for a specific line. If any type of failure occurs after this command is run, an inquiry message is sent to the system operator.

Use the Resume Line Recovery (RSMLINRCY) command to reestablish error recovery procedures for the line.

Top

Parameters

Keyword	Description	Choices	Notes
LINE	Line	<i>Name</i>	Required, Positional 1

Top

Line (LINE)

Specifies the name of the communications line whose recovery is to be stopped.

This is a required parameter.

Top

Examples

```
ENDLINRCY LINE(NYC2)
```

This command ends error recovery procedures for the line named NYC2.

Top

Error messages

*ESCAPE Messages

CPF2704

Line description &1 not found.

CPF5917

Not authorized to line description &1.

CPF5932

Cannot access line &1.

CPF5933

Line &1 not varied on.

CPF5935

Error occurred during command processing.

[Top](#)

End Job Log Server (ENDLOGSVR)

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

Parameters
Examples
Error messages

The End Job Log Server (ENDLOGSVR) command is used to end the job log server. The job log server writes job logs for jobs that are in a job log pending state. If more than one job log server job is active at the time this command is issued, all of the job log server jobs will be ended.

Restrictions:

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

Usage notes:

- If you only want to stop the production of a particular job log because, for example, it is very long or consuming too many resources, use the Change Job (CHGJOB) or End Job (ENDJOB) command instead of this command.

To determine which job a server is currently working on, display the server job's job log using the Display Job Log (DSPJOBLOG) command. Informational message, CPI1307, "Writing job log for &3/&2/&1" will be logged in the server job's job log while it is writing the job log for the job named in the message.

If you want to be able to write the complete job log for the job named in CPI1307 at a later time, change the job using the CHGJOB command specifying *PND on the LOGOUTPUT parameter.

If you do not need the job log and do not want to save it so it could be rewritten later, use the ENDJOB command on the job named in the CPI1307 message, specifying LOGLMT(0).

Top

Parameters

Keyword	Description	Choices	Notes
OPTION	How to end	*CNTRL, *IMMED	Optional, Positional 1

Top

How to end (OPTION)

Specifies whether the job log server is ended in a controlled manner or immediately.

*CNTRL

The server is ended in a controlled manner. Job logs which are in the process of being written are allowed to continue until they have completed.

*IMMED

The server is ended immediately. Jobs which are currently being processed by the job log server will not be removed from the system and the job logs currently being written will not be completed. The incomplete job logs will be rewritten from the beginning when a new job log server is started.

Top

Examples

ENDLOGSVR

This command ends the job log server in a controlled manner. Any job logs which are currently being written to spooled files by the job log server are allowed to complete.

[Top](#)

Error messages

*ESCAPE Messages

CPF134B

Job log server not ended.

[Top](#)

End Mode (ENDMOD)

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

Parameters
Examples
Error messages

The End Mode (ENDMOD) command ends (deactivates) a single mode or all active modes for a specific advanced program-to-program communications (APPC) remote location. The mode remains inactive until a Start Mode (STRMOD) command is run to start the mode. This command can be used to end all the sessions for a particular remote location and to cause an active switched connection to disconnect. The user can also specify how activities that have been requested on the remote system but have not yet been performed are to be handled.

The APPC Programming book, SC41-5443 has more information on the ENDMOD command.

Restriction: This command cannot be used to end (deactivate) Client Access/400 mode (QPCSUPP) at a remote location.

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location	<i>Communications name</i>	Required, Positional 1
DEV	Device	<i>Name, *LOC</i>	Optional
MODE	Mode	<i>Communications name, *NETATR, *ALL</i>	Optional
LCLLOCNAME	Local location	<i>Communications name, *LOC, *NETATR</i>	Optional
RMTNETID	Remote network identifier	<i>Communications name, *LOC, *NETATR, *NONE</i>	Optional
CPLPNDRQS	Complete pended requests	<i>*NO, *YES</i>	Optional

Top

Remote location (RMTLOCNAME)

Specifies the remote location name for which one or more modes are to be ended.

This is a required parameter.

Top

Device (DEV)

Specifies the device description name.

The possible values are:

***LOC** The device description is determined by the system.

device-name

Specify a device description name.

Mode (MODE)

Specifies the mode that is to be ended.

The possible values are:

*NETATR

The mode in the network attributes is used.

***ALL** All modes currently in use by the remote location are ended.

BLANK

The mode name (consisting of 8 blank characters) is used.

mode-name

Specify a mode name.

Note: SNASVCMG and CPSVCMG are reserved names and cannot be specified.

Top

Local location (LCLLOCNAME)

Specifies the local location name.

The possible values are:

***LOC** The local location name is determined by the system.

*NETATR

The LCLLOCNAME value specified in the system network attributes is used.

local-location-name

Specify the name of your location. The local location name is specified if you want to indicate a specific local location name for the remote location.

Top

Remote network identifier (RMTNETID)

Specifies the remote network ID used with the remote location.

The possible values are:

***LOC** The system selects the remote network ID.

*NETATR

The remote network identifier specified in the network attributes is used.

***NONE**

No remote network identifier (ID) is used.

remote-network-id

Specify the name of the remote network ID used.

Complete pended requests (CPLPNDRQS)

Specifies if the remote location can complete pending work, or if the pended work should be ended before being allowed to start.

The possible values are:

***NO** The requested activities currently in progress at the remote location can complete. Activities that have been requested but not started at the remote location will not be performed.

***YES** All requested activities are allowed to complete before the mode is ended.

Top

Examples

```
ENDMOD RMTLOCNAME (APPCRLOC) MODE (APPCMOD)
```

This command ends a mode named APPCMOD for remote location APPCRLOC.

Top

Error messages

*ESCAPE Messages

CPF598B

The &1 command failed for one or more modes.

Top

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)
Threadsaf: 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 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 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

*ESCAPE Messages

CPFA0B1

Requested operation not allowed. Access problem.

CPFA1B8

*IOSYSCFG authority required to use &1.

Top

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 i5/OS through iSeries Navigator's Application Administration support. The Change Function Usage Information (QSYCHFUI) API, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
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> , *SUSPEND, *STOP	Optional
DTAOPT	Data option	<u>*LIB</u> , *MGTCOL, *DLT	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 iSeries, 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.

CPFB94A

Collector communications error. Reason code &1.

[Top](#)

End Performance Trace (ENDPFRTTC)

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

Parameters
Examples
Error messages

The End Performance Trace (ENDPFRTTC) command will stop the collection of performance trace data in the QPM_STRPFRTTC trace table and optionally write performance trace data to a data base file. The QPM_STRPFRTTC 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 (STRPFRTTC) command. However, it will end and try to process any active trace in the QPM_STRPFRTTC 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:

- To use this command, you must have:
 - job control (*JOBCTL) special authority.
- 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	Name, <u>*FILE</u>	Optional, Positional 1
OPNID	Open file identifier	Simple name, <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 database reader. 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)

Specifies the database reader to be ended.

This is a required parameter.

name Specify the name of the reader to be ended.

Top

When to end reader (OPTION)

Specifies when the ended reader should stop processing.

*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.

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.

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

ENDRMPTSPT 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)
Threadsaf: 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

*ESCAPE Messages

CPFA1B8

*IOSYSCFG authority required to use &1.

[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 information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

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	<i>Integer</i> , *PRV	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.

integer

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 QENDJOB LMT 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 (BCHTIMLMT)

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 Subroutine (ENDSUBR)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The End Subroutine (ENDSUBR) command specifies the end of a subroutine, and must be paired with a previous Subroutine (SUBR) command. When the ENDSUBR command is processed, control is returned to the command immediately following the Call Subroutine (CALLSUBR) command that called the subroutine. If a value is to be returned, such as an error code, the returned value must be able to be stored into a 4-byte signed integer CL variable. The **Return value (RTNVAL)** parameter can be a variable or constant. If no RTNVAL parameter is defined, the value will default to zero.

Restrictions:

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

Top

Parameters

Keyword	Description	Choices	Notes
RTNVAL	Return value	<i>Integer, 0</i>	Optional

Top

Return value (RTNVAL)

Specifies the value to be returned from the subroutine. This can be a variable or constant that can be stored in a 4-byte signed integer CL variable. If the Call Subroutine (CALLSUBR) command that called the subroutine specified a **CL variable for returned value**, the variable is set to this value.

0 The subroutine returns a zero.

integer-constant

Specify the constant integer value for the return value.

CL-variable-name

Specify the name of the CL variable to contain the return value from the called subroutine. This must be a signed integer CL variable.

Top

Examples

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

This CL subroutine is identified by a SUBR command that contains the subroutine name, SUBR1, and is ended by the ENDSUBR command.

```
PGM
:
SUBR SUBR1
:
ENDSUBR RTNVAL(-1)
ENDPGM
```

This CL subroutine has an ENDSUBR command that returns a -1.

[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	* <u>CNTRLD</u> , *IMMED	Optional, Positional 1
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
CONFIRM	Confirm	* <u>ENVVAR</u> , *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 QENDJOB LMT 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 system 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	*IMMED, *CNTRL	Optional, Positional 1
DELAY	Controlled end delay time	1-86400, <u>30</u>	Optional, Positional 2
ENDSVR	End application servers	*YES, *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 system. To display the current TCP/IP traffic on the system, 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 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 system 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 system 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 system 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>	Optional, Positional 1
ALIASNAME	Alias name	<i>Simple name</i>	Optional

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.

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

character-value

Specify the internet address associated with the interface to be ended.

Top

Alias name (ALIASNAME)

Specifies the name 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 alias name specified must be for an IPv4 interface. If the alias name is for an IPv6 interface, an error message will be returned.

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

simple-name

Specify the name of the interface to be ended.

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.

Example 3: Ending a Token-Ring Interface using an Alias Name

```
ENDTCPIFC ALIASNAME(TEST_NETWORK)
```

This command causes the TCP/IP protocol stack to deactivate (end) the interface associated with the alias name TEST_NETWORK.

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.

TCP1B71
&1 interface not ended.

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 the local system and establish a point-to-point TCP/IP session.
2. Dial mode sessions (*DIAL) allow the local system to contact a remote system that supports point-to-point TCP/IP.

Note: Profiles of linetype *PPP can 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 job &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 system 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.

*EDRSQ

The Extended Dynamic Remote SQL (EDRSQ) 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.

***IBMHELP**

The IBM Online Help and Eclipse Information Center (IBMHELP) server is ended.

***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 (STRTCPSVR) 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.

***SRVSPTPRX**

The Service and Support Proxy (SRVSPTPRX) server is 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 system 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 Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_TRACE, can also be used to change the list of users that are allowed to perform trace operations.
- If 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 PRTRC parameter on this command or the Print Trace (PRTRC) 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 (PRTRC)

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 PRTRC (Print Trace) command is not run as part of this command.

***YES** The PRTRC (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 Watch (ENDWCH)

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

Parameters
Examples
Error messages

The End Watch (ENDWCH) command ends a watch session that was started by a STRWCH (Start Watch) command or by Start Watch (QSCSWCH) API.

Restrictions:

- To use this command, you must have service (*SERVICE) special authority, or be authorized to the Service watch function of Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_WATCH, can also be used to change the list of users that are allowed to start and end watch operations.
- If ending a watch session that is watching for a message within a job log, the issuer of the command must be running under a user profile which is the same as the job user identity of the job being watched, or the issuer of the command must be running under a user profile which has job control (*JOBCTL) special authority. Job control (*JOBCTL) special authority is also required when ending a session where jobs with a generic user name are being watched.
- If ending a watch session that was started specifying *ALL for the watch job name, or a generic user name, you must have *ALLOBJ special authority, or be authorized to the Watch Any Job function of Operating System through iSeries Navigator's Application Administration support. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_WATCH_ANY_JOB, can also be used to change the list of users that are allowed to start and end watch operations.

Top

Parameters

Keyword	Description	Choices	Notes
SSNID	Session ID	Name, *PRV	Required, Positional 1

Top

Session ID (SSNID)

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

This is a required parameter.

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

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

Top

Examples

Example 1: End Most Recently Started Watch

```
ENDWCH  SSNID(*PRV)
```

This command ends the watch session started most recently by the same user who is running the ENDWCH command.

Example 2: End a Specific Watch Session

```
ENDWCH  SSNID(MYSESSION)
```

This command ends the watch session MYSESSION.

[Top](#)

Error messages

*ESCAPE Messages

CPF39EC

Cannot end watch session &1 started by &2 command.

CPF39E1

Watch session &1 not found.

CPF39E2

There is no active watch session for current user profile.

CPF39E6

The user does not have the required authority.

CPF39E8

Not enough authority to watch operations.

CPF39E9

*JOBCTL special authority required.

[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	*CNTRLD, *IMMED, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

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

This is a required parameter.

***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.

name Specify the name of the writer to end.

Top

When to end writer (OPTION)

Specifies when the writer should stop processing.

***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 i5/OS 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 i5/OS 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 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 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 i5/OS 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

*ESCAPE Messages

CPFA1B8

*IOSYSCFG authority required to use &1.

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 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)
 Threadsafes: No

Parameters
 Examples
 Error messages

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

Restrictions:

- 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.
- The user ID and address must be enrolled in the system distribution directory.
- 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): Element list	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: Qualified object name	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: Element list	Optional
	Element 1: User ID	Character value	
	Element 2: Address	Character value	
DOCFILE	Document file	Single values: *NONE Other values: Qualified object name	Optional
	Qualifier 1: Document file	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
DOCMBR	Document member	Name, *FIRST	Optional

Keyword	Description	Choices	Notes
DSTID	Distribution identifier	Character value, <u>*NONE</u>	Optional
DSTIDEXN	Distribution ID extension	1-99, <u>*NONE</u>	Optional
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	Character value, <u>*DFT</u>	Optional
DOCD	Document description	Character value, <u>*DFT</u>	Optional
AUTHOR	Author	Single values: <u>*NONE</u> , *USRID Other values (up to 50 repetitions): Character value	Optional
DOCCLS	Document class	Character value, <u>*NONE</u>	Optional
KWD	Keyword	Single values: <u>*NONE</u> Other values (up to 50 repetitions): Character value	Optional
SUBJECT	Subject	Single values: <u>*NONE</u> , *DOCD Other values (up to 50 repetitions): Character value	Optional
DOCDATE	Document date	Date, <u>*NONE</u> , *CURRENT	Optional
FILCAB	File cabinet location	Character value, <u>*NONE</u>	Optional
CPYLST	Copy list	Single values: <u>*NONE</u> Other values (up to 50 repetitions): Character value	Optional
EXPDATE	Expiration date	Date, <u>*NONE</u>	Optional
REFERENCE	Reference	Character value, <u>*NONE</u>	Optional
ACTDATE	Action due date	Date, <u>*NONE</u> , *CURRENT	Optional
STATUS	Document status	Character value, <u>*NONE</u>	Optional
CMPPATE	Completion date	Date, <u>*NONE</u> , *CURRENT	Optional
PROJECT	Project	Character value, <u>*NONE</u>	Optional
DOCCHRID	Document character identifier	Single values: <u>*SYSVAL</u> , *DEVVD Other values: Element list	Optional
	Element 1: Graphic character set	Integer	
	Element 2: Code page	Integer	
DOCLANGID	Language ID	Character value, <u>*JOB</u>	Optional
DOCNTRYID	Country or region ID	Character value, <u>*JOB</u>	Optional
PERSONAL	Personal	<u>*NO</u> , *YES	Optional
DSTEXPDATE	Distribution expiry indicator	Element list	Optional
	Element 1: Date	Date, <u>*NONE</u> , *CURRENT	
	Element 2: Time	Time, <u>*ENDOFDAY</u>	
CMDCHRID	Command character identifier	Single values: <u>*SYSVAL</u> , *DEVVD Other values: Element list	Optional
	Element 1: Graphic character set	Integer	
	Element 2: Code page	Integer	

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.

***FILE** The database file specified on the **Document file (DOCFILE)** parameter and the **Document member (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 (IDPFILE)** parameter and the **Profile member (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 (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.

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 (TODOC)** parameter.

***NONE**

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

***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.

*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.

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

***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 (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 (TODOC)** parameter.

*NONE

An authorization list is not specified.

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.

*NONE

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

0-2047 Specify the access codes that control who can use the document. A maximum of 50 access codes can be specified. Access code 0 gives use (*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.

*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 (IDPMBR)** parameter are ignored, except the **Command character identifier (CMDCHRID)** parameter and the **Document character identifier (DOCCHRID)** parameter.

*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 (IDPMBR)** parameter is ignored.

*DSTIDIDP

The IDP information associated with the distribution document is used. The **Profile member (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 (DOCFILE)** parameter and **Document member (DOCMBR)** parameter are used for the document profile information.

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.

Qualifier 2: Library

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

***CURLIB**

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

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 (IDPFILE)** parameter.

***FIRST**

The first member created in the database file is used.

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.

Single values

***CURRENT**

You are performing the request for yourself.

Element 1: User ID

character

Specify another user's user ID or your user ID. You must have been given permission to work on behalf of another user or have all object (*ALLOBJ) special authority.

Element 2: Address

character

Specify another user's address or your address. You must have been given permission to work on behalf of another user or have *ALLOBJ authority.

Top

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.

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

Qualifier 2: Library

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

***CURLIB**

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

name Specify the library where the database file is located.

Top

Document member (DOCMBR)

Specifies the document database file member that is used.

*FIRST

The first member created in the database file is used.

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.

*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 (TYPE)** parameter.

Top

Distribution ID extension (DSTIDEXN)

Specifies the extension of the distribution identifier (if any) specified on the **Distribution identifier (DSTID)** parameter. This 2-digit extension has a value ranging from 01 through 99 that uniquely identifies duplicate distributions. The default value is 01.

*NONE

There is no duplicate distribution. *NONE is equivalent to an extension of 01.

distribution-id-extension

Specify the extension associated with the distribution. This is used to uniquely identify duplicate distributions.

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.

*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.

- *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.

2-65,535

Specify a document type identifier value. 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 (SYSCOD)** parameter.

Top

System code (SYSCOD)

Specifies the text used with the value specified on the **Document type (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.

- *DFT** The system supplies a default system code. If the value specified on the **Document type (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 (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 (IDPFILE)** parameter is specified.

- *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 (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 (TYPE)** parameter, the default document name will be the distribution document name specified when the distribution was sent.

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.

*NONE

No author is identified for the document.

*USRID

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

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 (IDPFILE)** parameter is specified.

Top

Document class (DOCCLS)

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

*NONE

No class is assigned to the document.

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

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

Top

Keyword (KWD)

Specifies the keywords that describe the document.

*NONE

No keywords are defined for this 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 (IDPFILE)** parameter is specified.

Top

Subject (SUBJECT)

Specifies the subject or subjects of the document.

***NONE**

No subject is defined for the document.

***DOCD**

The document description is used as the subject for the 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 (IDPFILE)** parameter is specified.

Top

Document date (DOCDATE)

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

***NONE**

No date is assigned to the document.

***CURRENT**

The system assigns the current system date to the 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 (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 (TYPE)** parameter and *NONE is specified on the **Profile file (IDPFILE)** parameter.

***NONE**

No filing cabinet reference is defined for this document.

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 (IDPFILE)** parameter is specified.

Top

Copy list (CPYLST)

Specifies the names of the users that receive this document.

***NONE**

No copy list is included for this document.

name 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 (IDPFILE)** parameter is specified.

Expiration date (EXPDATE)

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

*NONE

No document expiration date is specified.

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 (IDPFILE)** parameter is specified.

Reference (REFERENCE)

Specifies a reference associated with the document.

*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 (IDPFILE)** parameter is specified.

Action due date (ACTDATE)

Specifies the date when the action requested is due.

*NONE

No action due date request is specified.

***CURRENT**

The current system action due date is used.

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 (IDPFILE)** parameter is specified.

Document status (STATUS)

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

*NONE

No status is included in this document.

status 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 (IDPFILE)** parameter is specified.

Completion date (CMPDATE)

Specifies the date when the requested action is completed.

*NONE

No completion date is included.

*CURRENT

The current system date is used as the completion date.

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

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

Top

Project (PROJECT)

Specifies the project associated with the document.

*NONE

No project field information is included in this document.

name Specify the name of the project associated with the document. A maximum of 10 characters can be specified.

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

***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 (DOCLANGID)** parameter to see a list of valid identifiers.

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

***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 (DOCCNTRYID)** parameter to see a list of valid identifiers.

Note: This parameter is ignored if the **Profile file (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.

***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.

***NONE**

No expiration date,

***CURRENT**

The current date is used.

date Specify the value to use as the expiration date. The date must be specified in the format specified by the system value QDATFMT.

Element 2: Time

*ENDOFDAY

An expiration time is requested by the end of the specified date. The time is set to 23:59:59.

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

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(*DEV), DOCCHRID(*DEV) 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)
 Threadsafes: 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, QFM^TSRC</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.

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.

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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseries/infocenter>, 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 does not exist, the public authority will be determined by the **os400.file.create.auth** Java property value. If this Java property has not been set, the public authority for a created stream file is set to *RW.
- 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.
- This command does not support proxy CL commands.

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

Keyword	Description	Choices	Notes
REPLACE	Replace file	*YES, *NO	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 information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter>, 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 i5/OS 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.

CPF6E67

Command documentation not generated for proxy command &1 in &2.

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	
QMOMDATA	QUSRADSM	QUSRPYMSVR	
QMOMPROC	QUSRBRM	QUSRDRARS	
QPFRDATA	QUSRDIRCL	QUSRSYS	
QRCL	QUSRDIRDB	QUSRVI	

1. 'xxxxx' is the number of a primary auxiliary storage pool (ASP).
2. A different library name, in the format QUSRVxRxMx, can be created by the user for each previous release supported by IBM to contain any user commands to be compiled in a CL program for the previous release. For the QUSRVxRxMx user library, VxRxMx is the version, release, and modification level of a previous release that IBM continues to support.

***ALL** All libraries in the system, including QSYS, are searched.

name Specify the name of the library to be searched.

Top

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. For a GOTO command in a subroutine, the branch target must be associated with the same subroutine.

Restrictions:

- This command is valid only within a CL procedure.
- Using GOTO to branch into or out of a subroutine is not allowed.

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:

- The access code must be defined to the system before you can grant access code authority.
- The user being granted access code authority must be enrolled in the system distribution directory.
- To use this command, you must have all object (*ALLOBJ) or security administrator (*SECADM) special 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 (USER)** parameter.

This is a required parameter.

Single values

*REFUSER

The access code authority being granted is based on the (referred to) user profile name; specified for the **Reference user profile (REFUSER)** parameter.

Other values (up to 300 repetitions)

1-2047 Specify 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.

This is a required parameter.

You can specify 300 values for this parameter.

name Specify the name of the user profile of a user to whom you are granting access code authority.

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 specified for the **Document access code (ACC)** parameter.

*NONE

No referred to user is used to grant access code authority.

name Specify the name of the user profile 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, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSL, *OUTQ, *OVL, *PAGDFN, *PAGESEG, *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, *CNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *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.

Single values

*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.

Other values (up to 10 repetitions)

***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.

Data authorities

***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 to be 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.

Restrictions:

- The following user profiles cannot be specified for either of the parameters on this command:
QANZAGENT, QAUTPROF, QCLUMGT, QCLUSTER, QCOLSRV, QDBSHR, QDBSHRDO, QDFTOWN, QDIRSRV, QDLFM, QDOC, QDSNX, QEJB, QEJBSVR, QGATE, QIBMHELP, QIPP, QLPAUTO, QLPINSTALL, QMGTC, QMSF, QNETSPLF, 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)

Specifies the user profile to whom authority is to be granted.

This is a required parameter.

name Specify the name of the user profile.

Top

Referenced user (REFUSER)

Specifies the user profile to be referred to for authority.

This is a required parameter.

name Specify the name of the user profile.

Top

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.

Top

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.

Top

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 all object (*ALLOBJ) special 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 (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 (TOUSER)** parameter works. The users must be enrolled in the system distribution directory before you run this command.

*CURRENT

You are granting permission to someone working on your behalf.

name Specify the name of the user profile on whose behalf the user specified on the **To user profile (TOUSER)** parameter works.

Top

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.

[Top](#)

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.

[Top](#)

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 i5/OS 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.

Single values

***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.

Other values (up to 10 repetitions)

***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.

Data authorities

*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	<u>*CNTRL</u> D, *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 provided either explicitly or through the selection display. Refer to the **Duplicate job option (DUPJOB OPT)** parameter for more information.

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)

Specifies the job queue that will have its current and future entries withheld from further processing.

This is a required parameter.

Qualifier 1: Job queue

name Specify the name of the job queue to be held.

Qualifier 2: Library

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

*CURLIB

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

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

[Top](#)

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.

Restrictions:

1. To hold entries, you must have job control (*JOBCTL) special authority; otherwise you can hold only those entries that you added.
2. To use this command, you must have:
 - Use (*USE) authority to object QDFTJOBSCD, type *JOBSCD, in library QUSRSYS and execute (*EXECUTE) authority to library QUSRSYS.

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) ENTRYNBR(*ALL)
```

This command holds all entries in the job schedule.

Example 3: Holding an Individual Job Schedule Entry

```
HLDJOBSCDE JOB(PAYROLL) ENTRYNBR(*ONLY)
```

This command holds the entry PAYROLL in the job schedule.

Example 4: Holding a Generic Job Schedule Entry

```
HLDJOBSCDE JOB(PAY*) ENTRYNBR(*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)

Specifies the name of the output queue that will have its current and future spooled files withheld from further processing.

This is a required parameter.

Qualifier 1: Output queue

name Specify the name of the output queue to hold.

Qualifier 2: Library

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

*CURLIB

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

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

Top

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)

Specifies the spooling reader to be held.

This is a required parameter.

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)
Threadsafe: 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)

Specifies the spooled file to hold.

This is a required parameter.

***SELECT**

All spooled files that meet the selection values specified on the **Select files for (SELECT)** parameter are held. This value is mutually exclusive with values specified on the **Job name (JOB)** parameter, **Spooled file number (SPLNBR)** parameter, **Job system name (JOBSYSNAME)** parameter, and **Spooled file created (CRTDATE)** parameter.

name Specify the name of the spooled file to hold.

Top

Job name (JOB)

Specifies the job that created the file being held.

Single values

***** The job that issued this Hold Spooled File (HLDSPLF) command is the job that produced this file.

Qualifier 1: Job name

name Specify the name of the job that created the file being held.

Qualifier 2: User

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

Qualifier 3: Number

000000-999999

Specify the system-assigned job number.

Top

Spooled file number (SPLNBR)

Specifies the number of the spooled file that was created by the specified job.

***ONLY**

Only one spooled file in the job has the specified file name; therefore, the number of the spooled file is not necessary.

***LAST**

The spooled file with the highest number and the specified file name is used.

***ANY** The spooled file number is not used to determine which spooled file is used. Use this value when the job system name parameter or the spooled file create date and time parameter is to take precedence over the spooled file number when selecting a spooled file.

1-999999

Specify the number of the 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.

name Specify the name of the system where the job that created the spooled file ran.

Top

Spooled file created (CRTDATE)

Specifies the date and time the spooled file was created. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met.

Single values

***ONLY**

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

***LAST**

The spooled file with the latest create date and time of the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

Element 1: Creation date

date Specify the date the spooled file was created.

Element 2: Creation time

***ONLY**

There is one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.

***LAST**

The spooled file with the latest create time of the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.

time Specify the time the spooled file was created.

Top

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.

Element 1: User

***CURRENT**

Only files created by the user running this command are selected.

***ALL** Files created by all users are selected.

name Specify a user name. Only files created by that user name are selected.

Element 2: Print device

***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.

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.

Element 3: Form type

***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.

Element 4: User data

***ALL** Files with any user data tag specified are selected.

user-data

Specify the user data tag to select the file.

Element 5: ASP

***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.

1-32 Specify the auxiliary storage pool (ASP) of the files being selected.

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.

*
- 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.

***CURASPGRP**

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.

name Specify the name of the auxiliary storage pool device description. 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.

***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

*CURASGRP or *ASPGRPPRI specified and thread has no ASP group.

CPFB8ED

Device description &1 not correct for operation.

Top

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> , *CNTRLD, *PAGEEND	Optional, Positional 2

Top

Writer (WTR)

Specifies the spooling writer being held.

This is a required parameter.

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.

*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 topic "Built-in functions for CL" in the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter> 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 Linux Server (INSLNXSVR)

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

Parameters
Examples
Error messages

The Install Linux Server (INSLNXSVR) command installs the Linux server Base Operating System on an Integrated xSeries server. INSLNXSVR also installs Integrated Server Support code on the server.

Linux server installation occurs in two steps. During the first step, the INSLNXSVR command creates all necessary objects to manage the server. This includes a network server description, message queue, line descriptions, storage spaces and TCP/IP interfaces.

During the second step of the Linux server installation, the Integrated xSeries server is varied on to start the Linux server installation.

Further Linux server installation is performed using the file server console and the normal Linux server install process.

When INSLNXSVR completes normally, the Linux server is left in a varied on state.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG), all object (*ALLOBJ) and job control (*JOBCTL) special authorities to run this command.
- The file server must be varied off initially.
- The file server will also be varied off and varied back on during the second step of the install as Linux server installs and requires the server to reboot.

Usage Notes:

Any errors that occur during the first step of configuring the file server will result in the failure of this command.

After this command is run, if you need to manage the different resources created, use the following commands:

- To check out the status of the Linux server, use the Work with Configuration Status command; WRKCFGSTS CFGTYPE(*NWS).
- To manage the server just installed, use the Work with Network Server Descriptions command; WRKNWSD NWSD(network-server-name).
- To manage the line descriptions created by this command, use the Work with Line Descriptions command; WRKLIND LIND(nwsdname*). The line descriptions are named using the network server name (NWS parameter) specified on the INSLNXSVR command.
- To manage the TCP/IP interfaces created by this command, use the Work with TCP/IP Network Status (NETSTAT) command, option 1. Another option is to use the Configure TCP/IP (CFGTCP) command, option 1.
- To manage the network server configurations just created by this command, use the Work with NWS Configuration command; WRKNWSCFG NWSCFG(nwsdname*). The network server configurations are named using the network server name (NWS parameter) specified on the INSLNXSVR command.

Top

Parameters

Keyword	Description	Choices	Notes
NWSD	Network server description	<i>Name</i>	Required, Key, Positional 1
RSRCNAME	Resource name	<i>Name</i> , *ISCSI	Required, Positional 2
LNXSVRDST	Linux server distribution	<i>Character value</i>	Required, Positional 3
LNXSRCDIR	Linux source directory	<i>Path name</i> , *DFT	Optional
TCPPOPCFG	TCP/IP port configuration	Single values: *NONE Other values (up to 4 repetitions): <i>Element list</i>	Optional
	Element 1: Port	1, 2, 3, 4	
	Element 2: Linux internet address	<i>Character value</i>	
	Element 3: Linux subnet mask	<i>Character value</i>	
	Element 4: Linux gateway address	<i>Character value</i>	
VRTETHPORT	Virtual Ethernet port	Single values: *NONE Other values (up to 4 repetitions): <i>Element list</i>	Optional
	Element 1: Port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Linux internet address	<i>Character value</i>	
	Element 3: Linux subnet mask	<i>Character value</i>	
	Element 4: Associated port	<i>Name</i> , *NONE	
TCPDMNNAME	TCP/IP local domain name	<i>Character value</i> , *SYS	Optional
TCPNAMSVR	TCP/IP name server system	Single values: *SYS, *NONE Other values (up to 3 repetitions): <i>Character value</i>	Optional
SVRSTGSIIZE	Server storage space sizes	<i>Element list</i>	Optional
	Element 1: Install source size	200-2047, <u>200</u>	
	Element 2: System size	1024-1024000, <u>5000</u>	
SVRSTGASP	Storage space ASP	<i>Element list</i>	Optional
	Element 1: Install source ASP	1-255, <u>1</u>	
	Element 2: System ASP	1-255, <u>1</u>	
STGASPDEV	Server storage ASP device	<i>Element list</i>	Optional
	Element 1: Install source ASP device	<i>Name</i>	
	Element 2: System ASP device	<i>Name</i>	
LNGVER	Language version	<i>Integer</i> , *PRIMARY	Optional
SYNCTIME	Synchronize date and time	*YES, *NO	Optional
RSTDDEVRSRSC	Restricted device resources	Single values: *NONE, *ALL Other values (up to 10 repetitions): <i>Name</i> , *ALLTAPE, *ALLOPT	Optional
SHUTDTIMO	Shutdown timeout	2-45, <u>15</u>	Optional
ACTTMR	Activation timer	30-1800, <u>120</u>	Optional

Keyword	Description	Choices	Notes
CMNMSGQ	Communications message queue	Single values: *SYSOPR Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Communications message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
STGPTH	Storage path	<i>Element list</i>	Optional
	Element 1: Network server host adapter	<i>Communications name</i>	
VRTETHPTH	Virtual Ethernet path	Values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Port	*VRTETHPTP, *VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Network server host adapter	<i>Communications name</i>	
SHUTDPORT	Shutdown TCP port	1024-65535, 8700	Optional
VRTETHCTLP	Virtual Ethernet control port	1024-65535, 8800	Optional
RMTNWSCFG	Remote system NWSCFG	<i>Name, *DFT</i>	Optional
SPNWSCFG	Service processor NWSCFG	<i>Name, *DFT</i>	Optional
CNNNWSCFG	Connection security NWSCFG	<i>Name, *DFT</i>	Optional
DFTSECRULE	Default IP security rule	<i>Character value, *NONE, *GEN</i>	Optional
IPSECRULE	IP security rule	1-16, *DFTSECRULE, *NONE	Optional
INZSP	Initialize service processor	*MANUAL, *SYNC, *AUTO, *NONE	Optional
ENBUNICAST	Enable unicast	*NO, *YES	Optional
EID	Enclosure identifier	Single values: *AUTO Other values: <i>Element list</i>	Optional
	Element 1: Serial number	<i>Character value</i>	
	Element 2: Manufacturer type and model	<i>Character value</i>	
SPNAME	Service processor name	<i>Character value, *SPINTNETA</i>	Optional
SPINTNETA	SP internet address	<i>Character value</i>	Optional
SPAUT	SP authentication	Single values: *DFT Other values: <i>Element list</i>	Optional
	Element 1: User name	<i>Character value</i>	
	Element 2: User password	<i>Character value</i>	
SPCERTID	SP certificate identifier	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: Component	*COMMONNAME, *EMAIL, *ORGUNIT	
	Element 2: Compare value	<i>Character value</i>	
RMTSYSID	Remote system identifier	Single values: *EID Other values: <i>Element list</i>	Optional
	Element 1: Serial number	<i>Character value</i>	
	Element 2: Manufacturer type and model	<i>Character value</i>	
DELIVERY	Delivery method	<i>Character value, *DYNAMIC, *MANUAL</i>	Optional
CHAPAUT	CHAP authentication	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: CHAP name	<i>Character value</i>	
	Element 2: CHAP secret	<i>Character value</i>	

Keyword	Description	Choices	Notes
BOOTDEVID	Boot device ID	Single values: *SINGLE Other values: <i>Element list</i>	Optional
	Element 1: Bus	0-255	
	Element 2: Device	0-31	
	Element 3: Function	0-7	
DYNBOOTOPT	Dynamic boot options	<i>Element list</i>	Optional
	Element 1: Vendor ID	<i>Character value, *DFT</i>	
	Element 2: Alternate client ID	<i>Character value, *ADPT</i>	
RMTIFC	Remote interfaces	<i>Element list</i>	Optional
	Element 1: SCSI interface	<i>Element list</i>	
	Element 1: Adapter address	<i>Hexadecimal value</i>	
	Element 2: Internet address	<i>Character value</i>	
	Element 3: Subnet mask	<i>Character value</i>	
	Element 4: Gateway address	<i>Character value</i>	
	Element 5: iSCSI qualified name	<i>Character value, *GEN</i>	
	Element 2: LAN interface	<i>Element list</i>	
	Element 1: Adapter address	<i>Hexadecimal value</i>	
	Element 2: Internet address	<i>Character value</i>	
	Element 3: Subnet mask	<i>Character value</i>	
	Element 4: Gateway address	<i>Character value</i>	
	TEXT	Text 'description'	
CFGFILE	Configuration file	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Configuration file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
VRTPTPORT	Virtual PTP Ethernet port	<i>Element list</i>	Optional
	Element 1: Internet address	<i>Character value, *GEN</i>	
	Element 2: Linux internet address	<i>Character value, *GEN</i>	

Top

Network server description (NWS D)

Specifies the network server description to be created. The network server description is created using the values specified in this command. The Create Network Server Description (CRTNWS D) command will be used to create a *NWS D object of TYPE(*LINUX). The name is also used as the computer name of the Linux server that is installed as well as the TCP host name for the Linux server.

This is a required parameter.

name Specify the name of the network server. The network server name can be up to eight characters. The following characters are allowed in NWS D names:

- Alphabetical characters A through Z
- Digits 0 through 9

Top

Resource name (RSRCNAME)

Specifies the resource name that identifies the hardware that the description uses.

This is a required parameter.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. Specify the resource name of the communications adapter. The resource name represents the File Server IOA.

*ISCSI

This network server resource is an iSCSI attached server.

name Specify the resource name. The maximum length of the resource name is 10 characters.

Top

Linux server distribution (LNXSVRDST)

Specifies the version of Linux server to install on this network server.

This is a required parameter.

character-value

Specify the Linux server distribution version to install. You can use F4 while prompting the command to see the complete list of allowed values for this parameter.

Top

Linux source directory (LNXSRCDIR)

Specifies the directory associated with the Linux server CD-ROM image that is used as the source for the install. The directory name may reference an optical volume ('/QOPT/volume'), a folder ('/QDLS/folder'), or an integrated file system (IFS) directory ('/dir1/dir2'). An example of a CD-ROM volume path name would be '/QOPT/NTSRV40A'.

To find out the name of a volume on an optical device, use the command: DSPOPT VOL(*MOUNTED) DEV(device-name). If you do not know the name of the optical device, use the command: WRKCFGSTS CFGTYPE(*DEV) CFGD(*OPT)

To find out the name of a path in an IFS directory, use the Work with Object Links (WRKLNK) command. WRKLNK will show the directory object path names on the system.

***DFT** The default directory path name to use is determined by searching the QOPT file system (/QOPT directory in IFS). The path name for the first volume in QOPT that contains a DOSUTILS directory with a LOADLIN.EXE file in it is used as the default.

Note: Using *DFT on a system with an optical media library is not recommended. Every volume located in the '/QOPT' path will be searched for a valid Linux install source and will cause each volume to be retrieved and mounted. A specific optical volume path should be specified.

path-name

Specify the Linux server source directory path name to use for the install.

Note: The specified directory must contain a DOSUTILS directory with a LOADLIN.EXE file in it.

You can use F4 while prompting the command to see a list of path names for optical volumes that are allowed values for this parameter. Folder or IFS path names that are not optical volumes are also allowed, but are not listed when using F4.

Top

TCP/IP port configuration (TCPPOPTCFG)

Specifies the Linux TCP/IP configuration values that are specific to a port on the network server. This information consists of four parts including the identification of the network server port, the internet address, the subnet mask and the default gateway assigned to the port.

Single values

*NONE

Specifies that there is no Linux TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the network server port number to be configured. Specify one of the following values:

- 1 Network server port number 1 is configured.
- 2 Network server port number 2 is configured.
- 3 Network server port number 3 is configured.
- 4 Network server port number 4 is configured.

Element 2: Linux internet address

character-value

Specify the Linux internet address for the port in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 3: Linux subnet mask

character-value

Specifies the subnet mask for the Linux internet address in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Linux gateway address

character-value

Specifies the default gateway address for the Linux internet address in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual Ethernet port (VRTETHPORT)

Specify the TCP/IP configuration for the virtual Ethernet used by the file server.

Note: The VRTETHPORT parameter is only available for Integrated xSeries servers.

Single values

*NONE

Specifies that there is no Linux TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the network server virtual Ethernet port number to be configured. Specify one of the following values:

***VRTETHn**

The network server virtual Ethernet port 'n' is configured, where 'n' can have a value of 0 through 9.

Element 2: Linux internet address

character-value

Specify the Linux internet address for the port in the form, nnn.nnn.nnn.nnn , where nnn is a decimal numbers ranging from 0 through 255. The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 3: Linux subnet mask

character-value

Specifies the subnet mask for the Linux internet address in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Associated port

Specifies the resource name that describes the port that is used to establish a connection between a Linux network server and the network.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. The resource name is on the port. For example, the resource name may be CMN01 on a Ethernet port.

*NONE

An associated port resource name is not associated with the line.

name Specify the resource name.

Top

TCP/IP local domain name (TCPDMNNAME)

Specifies the local domain name associated with the network server.

A domain name can be a text string having 2 to 255 characters. Domain names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The following characters are allowed in domain names:

- Alphabetical characters A through Z
- Alphabetical characters a through z
- Digits 0 through 9
- Minus sign (-)
- Period (.). Periods are only allowed when they separate labels of domain style name (refer to RFC 1034).

Uppercase and lowercase characters are allowed, but no significance is attached to the case. The case is maintained as entered. The first and last character of the host name must be an alphabetic character or a digit.

***SYS** Specifies that the local domain name for the network server should be the same value as is configured for the system.

character-value

Specify a TCP domain name to be associated with the network server.

Top

TCP/IP name server system (TCPNAMSVR)

Specifies the internet address of the name server system that is used by the network server. Typically, this is the same value as it is for the system.

Single values

***SYS** The name server system used by the network server should be the same as for the system.

***NONE**

There is no name server to be used by the network server.

Other values (up to 3 repetitions)

character-value

Specify an internet address for the name server system to be used by the network server. Up to three remote name server systems can be specified. The name server systems are used in the order they are specified.

Top

Server storage space sizes (SVRSTGSIIZE)

Specifies the size of the server storage spaces, in megabytes.

Element 1: Install source size

Specifies the size of the storage space that holds the files used to install the Linux server.

Notes:

1. The contents of the DOSUTILS directory of the Linux server installation media and the Integrated Server Support code are copied to the install source drive. The size specified for the install source drive must be large enough to hold this data.

200 A storage space of 200 megabytes is used for install files.

200-2047

Specifies the install source size value in megabytes.

Element 2: System size

Specifies the size, in megabytes, of the storage space that the Linux server operating system is installed on.

5000 A storage space of 5000 megabytes is used for system storage.

1024-1024000

Specifies the system size value, in megabytes.

Top

Storage space ASP (SVRSTGASP)

Specifies the auxiliary storage pool (ASP) identifiers for the storage space that will contain the files used to install the Linux server and the storage space that will contain the Linux server operating system.

Note: You cannot specify values for both the SVRSTGASP and STGASPDEV parameters.

Note: Null (omitted) values are specified with the characters *N, which mean that no value was specified. The value specified for the corresponding Server storage ASP device (STGASPDEV) element will be used if specified. Otherwise, the default value will be used. *N is needed only when another value following the omitted element is being specified.

Element 1: Install source ASP

Specifies the auxiliary storage pool for the storage space that holds the files used to install the Linux server.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

1-255 Specify the ASP identifier of the ASP to hold install files. Valid values depend on how many ASPs are defined on the system.

Element 2: System ASP

Specifies the auxiliary storage pool for the storage space that holds the Linux server operating system.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

1-255 Specify the ASP identifier of the ASP to hold the operating system. Valid values depend on how many ASPs are defined on the system.

Top

Server storage ASP device (STGASPDEV)

Specifies the auxiliary storage pool (ASP) device for the storage space that will contain the files used to install the Linux server and the storage space that will contain the Linux server operating system.

Note: You cannot specify values for both the SVRSTGASP and STGASPDEV parameters.

Note: The ASP must have been activated (by varying on the ASP device) and have a status of 'Available'.

Element 1: Install source ASP device

Specifies the independent auxiliary storage pool device for the storage space that holds the files used to install the Linux server.

name Specify the device name of the ASP to use for the network server storage space.

Element 2: System ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the Linux server operating system.

name Specify the device name of the ASP to use for the network server storage space.

Top

Language version (LNGVER)

Specifies the installed language environment used to display Integrated Server Support text and messages. Note that a smaller set of languages are available for messages displayed during installation than for messages displayed after installation.

*PRIMARY

The installed language environment for Integrated Server Support text and messages is based on the language feature of the system's primary language.

integer

Specifies the language feature number that will be used to select the Integrated Server Support text and messages.

Top

Synchronize date and time (SYNCTIME)

Specifies whether the system should synchronize the network server date and time with the host system date and time.

The QTIMZON system value must be set to the correct value for time synchronization to work correctly.

*YES The system synchronizes the file server date and time with the host system date and time at every vary on and at least every 30 minutes thereafter.

*NO The system synchronizes the file server date and time with the host system date and time when the network server description is varied on, but will not keep the date and time synchronized while the network server description is in a varied on state.

Top

Restricted device resources (RSTDDEVRSC)

Specifies the system's device resource names that are restricted and cannot be used by the Linux server.

Note: Only tape and optical device resources can be restricted.

Single values

*NONE

No device resources are restricted from the network server. Therefore, any tape or optical device resources that exist on the system can be used.

*ALL All tape and optical resources are restricted from being used by the network server.

Other values (up to 10 repetitions)

*ALLOPT

All optical resources are restricted from being used by the network server.

Note: This value can only be specified once.

*ALLTAPE

All tape resources are restricted from being used by the network server.

Note: This value can only be specified once.

name Specify the name of the restricted device resource that cannot be used by the network server.

Top

Shutdown timeout (SHUTDTIMO)

Specifies the server shutdown time-out value, in minutes. This is used to limit the amount of time that the servers operating system is allowed to shutdown before the server is varied offline.

15 The network server default shutdown time-out value is used.

2-45 Specify the time (in minutes) to wait. The system waits until the network servers operating system has shutdown successfully, or until the specified time passes before varying the network server offline.

Top

Activation timer (ACTTMR)

Specifies the amount of time (in seconds) the system will wait for the connection to be established to the remote server's service processor and to power on the remote server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

120 The activate time of 120 seconds is used.

30-1800
Specify the number of seconds to wait.

Top

Communications message queue (CMNMSGQ)

Specifies the name of a message queue to receive communications status messages.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Single values

*SYSOPR

Causes messages to be placed in the system operator message queue (QSYSOPR) in library QSYS.

Qualifier 1: Communications message queue

name Specify the name of a message queue to receive communications status messages.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be used.

Top

Storage path (STGPTH)

Specifies the storage paths the storage spaces can use. This information consists of the **Network server host adapter (NWSH)** description.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Element 1: Network server host adapter

communications-name

Specify the name of an existing **Network server host adapter (NWSH)** description.

Top

Virtual Ethernet path (VRTETHPTH)

Specifies the virtual Ethernet paths the Ethernet line descriptions can use. This information consists of two parts including the virtual Ethernet port and the **Network server host adapter (NWSH)** description. You can enter up to five values for this parameter. You must enter at least one virtual Ethernet path which is the path to be used by the *VRTETHPTP line description.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

You can specify 5 values for this parameter.

Element 1: Port

***VRTETHPTP**

The network server virtual Ethernet point to point port is configured.

***VRTETHn**

The virtual Ethernet port 'n' is configured, where 'n' can have a value of 0 through 9.

Element 2: Network server host adapter

communications-name

Specify the name of an existing **Network server host adapter (NWSH)** description. The network server host adapter name does not need to be unique for each VRTETHPTH parameter on this NWSD.

Top

Shutdown TCP port (SHUTDPORT)

Specifies the TCP port to use for shutdown.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

8700 Use the TCP port number of 8700.

1024-65535

Specify the port number identifying the port that is to be used for shutdown.

Top

Virtual Ethernet control port (VRTETHCTLP)

Specifies the TCP port to use for virtual Ethernet control.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

8800 Use the TCP port number of 8800.

1024-65535

Specifies the port number identifying the port that is to be used for virtual Ethernet control.

Top

Remote system NWSCFG (RMTNWSCFG)

Specifies the remote system network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default remote system network server configuration name of 'nwsdnameRM' where nwsdname is the name of the network server description.

name Specify the name of an existing remote system network server configuration.

Top

Service processor NWSCFG (SPNWSCFG)

Specifies the service processor network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default service processor network server configuration name of 'nwsdnameSP' where nwsdname is the name of the network server description.

name Specify the name of an existing service processor network server configuration.

Top

Connection security NWSCFG (CNNNWSCFG)

Specifies the connection security network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default connection security network server configuration name of 'nwsdnameCN' where nwsdname is the name of the network server description.

name Specify the name of an existing connection security network server configuration.

Top

Default IP security rule (DFTSECRULE)

Specifies the default IP Security (IPSec) rule used between the hosting and remote system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the connection security network server configuration specified for the **Connection security NWSCFG (CNNNWSCFG)** parameter already exists.

A value other than *DFTSECRULE must be specified on the IP security rule (IPSECRULE) parameter in this case.

***NONE**

IP Security rules are not configured.

***GEN** The system will automatically generate a random pre-shared key.

character-value

Specify the pre-shared key. A pre-shared key is a nontrivial string up to 32 characters long using only the following characters: A-Z a-z 0-9 + = % & () , _ - . : ;. These characters are part of the Syntactic Character Set (character set number 640).

Top

IP security rule (IPSECRULE)

Specify the relative entry of the IP security rules (IPSECRULE) parameter, defined in the existing connection security network server configuration that will be used as the initial IP security setting between the hosting and remote system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFTSECRULE**

Use the value specified on the **Default IP security rule (DFTSECRULE)** parameter.

Note: This value is not valid when the **Connection security NWSCFG** specified already exists.

***NONE**

Remote interface will not use any security rule.

1-16 Remote interface will use security rule specified

Top

Initialize service processor (INZSP)

Specifies how the remote system's service processor is secured.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: Some service processors do not support secure connections. Use *NONE for these service processors. Additional information can be found at Integrated xSeries solutions at <http://www.ibm.com/servers/eserver/iserries/integratedxseries>.

***MANUAL**

Security parameters are manually configured on remote system's service processor. *MANUAL provides the highest security. To use this option it is required that the remote system's service processor is pre-configured with a user name, password and certificate. Certificate management will be required. This method is appropriate when connecting to the service processor via public networks to protect the password.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***AUTO**

Parameters are automatically configured on the remote system's service processor. *AUTO provides security without requiring pre-configuration of the remote system's service processor. The remote system's service processor will automatically generate a certificate. The connection is secure once initialized. This option is appropriate if the interconnecting network is physically secure or is protected by a firewall.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***SYNC**

This network server configuration will synchronize the self-signed certificate with the service processor.

Note: This value is only valid when the service processor has already been configured by another network server configuration with *AUTO specified for the Initialize service processor (INZSP) parameter. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

***NONE**

The password is sent in the clear. Use this only if the interconnecting network is physically secure.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Top

Enable unicast (ENBUNICAST)

Specifies whether unicast packet distribution is to be used. Unicast is a transmission method where packets are sent directly to the specified **Service processor name (SPNAME)** or **SP internet address (SPINTNETA)** parameter. The system identification for the Enclosure identifier (EID) parameter is automatically retrieved if *AUTO is specified and the system hardware supports it.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***NO** Disable unicast

*YES Enable unicast.

Top

Enclosure identifier (EID)

Specifies the identifying serial number, type and model of the enclosure containing the service processor. They are required to locate the remote system on the network when ENBUNICAST(*NO) is specified. Look for these values on the label of the system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

*AUTO

Automatically retrieve the identifier when ENBUNICAST(*YES) is specified.

Element 1: Serial number

character-value

Specify the remote system's machine serial number.

Element 2: Manufacturer type and model

character-value

Specify the remote system's machine type and model in the form tttmmm where ttt is the machine type and mmm is the machine model number.

Note: The remote system's machine type and model may be omitted if the systems serial number is unique on the network.

Top

Service processor name (SPNAME)

Specifies the remote system's service processor host name.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Note: This parameter is required when ENBUNICAST(*YES) is specified.

*SPINTNETA

The remote system is identified by the value specified for the **SP internet address (SPINTNETA)** parameter.

host-name

Specify the remote system's service processor host name.

Top

SP internet address (SPINTNETA)

Specify the remote system's service processor internet address. Internet addresses are expressed in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

This parameter is required when SPNAME(*SPINTNETA) is specified.

Note: This parameter is ignored when ENBUNICAST(*NO) is specified.

character-value

Specify the internet address of the service processor.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

*DFT The default service processor userid and password are used.

Note: This value is only valid when INZSP(*NONE) is specified on this command or in the existing network server configuration.

Element 1: User name

user-name

Specify the remote system's service processor user name.

Element 2: User password

password-value

Specify remote system's service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

SP certificate identifier (SPCERTID)

The SP certificate identifier specifies one of three possible fields that identifies the service processor's certificate. This parameter is specified to provide additional validation that the certificate is from the service processor. The contents of the selected field must exactly match the value of the field that was entered when the certificate was generated or requested from a certificate authority.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

This parameter is required when INZSP(*MANUAL) is specified.

Single values

*NONE

Service processor certificate is not configured.

Element 1: Component

***COMMONNAME**

Selects the certificates common name specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "ASM Domain Name" field used to generate a self-signed certificate or generate a certificate signing request.

***EMAIL**

Selects the certificate's e-mail address specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "Email Address" field used to generate a self-signed certificate or generate a certificate signing request.

***ORGUNIT**

Selects the certificate's organizational unit specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "Organizational Unit" field used to generate a self-signed certificate or generate a certificate signing request.

Element 2: Compare value

character-value

Specify the certificates component compare value. Specify no more than 255 characters of text, enclosed in apostrophes.

Top

Remote system identifier (RMTSYSID)

Specifies the identifying serial number, type and model of the remote system. When specified, they are used to locate the remote system on the network. Look for these values on the label of the remote system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

*EID Use the service processor identifier.

Element 1: Serial number

character-value

Specify the remote system's machine serial number.

Element 2: Manufacturer type and model

character-value

Specify the remote system's machine type and model in the form tttmmm where ttt is the machine type and mmm is the machine model number.

Note: The remote system's machine type and model may be omitted if the systems serial number is unique on the network.

Top

Delivery method (DELIVERY)

Specifies how the parameters necessary to configure the remote system are delivered.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

*DYNAMIC

Parameters are dynamically delivered to the remote system using DHCP.

*MANUAL

Parameters are manually configured on the remote system using the BIOS utilities (System BIOS or Adapter BIOS - ALT-Q).

Top

CHAP authentication (CHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the host system iSCSI target to authenticate the remote system initiator node.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

*NONE

CHAP authentication is not enabled.

Element 1: CHAP name

name Specify the CHAP name. CHAP names are translated to ASCII code page 850 using the job's coded character set identifier (CCSID).

Element 2: CHAP secret

character-string

Specify the secret you want to use for the Challenge Handshake Authentication Protocol as a value up to 24-characters long using only the following invariant characters: A-Z a-z 0-9 + = % & () , _ - . : ;. These characters are part of the Syntactic Character Set (character set number 640).

Boot device ID (BOOTDEVID)

Specifies the PCI Function Address (Bus/Device/Function) of the iSCSI adapter in the remote system that will be used to boot from. This information can be accessed using the BIOS utilities (System BIOS or Adapter BIOS - ALT-Q).

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***SINGLE**

The single iSCSI adapter is used on the remote system

Note: Remote systems with more than one iSCSI adapter installed in the server are required to specify which adapter will be used to boot from.

Element 1: Bus

0-255 Specify the bus number of the remote system's iSCSI adapter that will be used to boot.

Element 2: Device

0-31 Specify the device number of the remote system's iSCSI adapter that will be used to boot.

Element 3: Function

0-7 Specify the function number of the remote system's iSCSI adapter that will be used to boot.

Dynamic boot options (DYNBOOTOPT)

Specifies the configuration of the internal DHCP server.

This is an advanced function.

This parameter is used to configure the internal DHCP Server that is part of the iSCSI Target Host Bus Adapter firmware and it's required to provide IP address and diskless boot parameters for the remote iSCSI Initiator.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

This parameter is only valid when DELIVERY(*DYNAMIC) is specified.

Element 1: Vendor ID

The client and server are pre-configured to a fixed vendor ID (IBM_ISAN). Network administrators can configure clients to define their own identifying values to convey hardware, operating system or other identifying information. DHCP option 60 described in the IETF RFC 2132 is used for this function.

IBM_ISAN

The default vendor ID of IBM_ISAN will be used.

character-value

Specify the vendor ID of the remote system's iSCSI adapter that will be used.

Element 2: Alternate client ID

Used by clients to specify their unique identifier to the server. Each client's identifier must be unique among all other client identifiers used on the effective DHCP network to which the client is attached (that is, the client's local subnet and any remote subnets reachable using DHCP relay). Vendors and system administrators are responsible for choosing client identifiers that meet this requirement for uniqueness. DHCP option 61 described in the IETF RFC 2132 is used for this function.

*ADPT

The default Client ID consists of the adapter address for the remote system's iSCSI adapter. This value will be used to identify the remote system.

character-value

Specify the Client ID of the remote system's iSCSI adapter that will be used to boot.

Top

Remote interfaces (RMTIFC)

Specifies the remote system's interfaces. This information is used to identify and configure the remote system's interfaces. Each adapter has two functions to support a SCSI and a LAN interface.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Element 1: SCSI interface

Element 1: Adapter address

hexadecimal-value

Specify the 12-character hexadecimal adapter address for the remote system's SCSI interface.

Element 2: Internet address

character-value

Specify the internet address for the remote system's SCSI interface in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value

Specify the subnet mask for the remote system's SCSI interface in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

character-value

Specify the gateway address for the remote system's SCSI interface in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 5: iSCSI qualified name

***GEN** The system will automatically generate the iSCSI qualified name.

name Specify the iSCSI qualified name for the remote system's SCSI interface. iSCSI qualified names are translated to UTF-8 using the job's coded character set identifier (CCSID). The following characters are allowed in iSCSI qualified names:

- Alphabetical characters A through Z converted to lower case (refer to RFC 3722)
- Alphabetical characters a through z
- Digits 0 through 9
- Period (.)
- Dash (-)
- colon (:)

Element 2: LAN interface

Element 1: Adapter address

hexadecimal-value

Specify the 12-character hexadecimal adapter address for the remote system's LAN interface.

Element 2: Internet address

character-value

Specify the internet address for the remote system's LAN interface port in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

character-value

Specify the subnet mask for the remote system's LAN interface in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

character-value

Specify the gateway address for the remote system's LAN interface in the form, nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: TCP/IP name server system

character-value

Specify an internet address for the domain name server to be used by the the remote system's adapter.

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the objects created by this command.

*BLANK

Text is not specified.

character-value

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

Top

Configuration file (CFGFILE)

Specifies the source file that contains configuration data, to be used in activating or further defining the server.

Single values

*NONE

No configuration file is specified.

Qualifier 1: Configuration file

name Specify the name of a source file containing the configuration data member(s) for the server. At the time the server is activated, all members in the file will be processed.

Qualifier 2: Library

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

*CURLIB

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

name Specify the name of the library to be used.

Top

Virtual PTP Ethernet port (VRTPTPPORT)

Specifies the TCP/IP configuration for the virtual point-to-point Ethernet port.

Notes:

1. The VRTPTPPORT parameter must be used for Integrated xSeries servers running on the host system.
2. The subnet mask that is used for both sides of the virtual point-to-point Ethernet port is 255.255.255.0 by default. Therefore, the internet addresses that are chosen for both sides of the virtual point-to-point Ethernet port must have the same values for the first three parts of the internet addresses.

Element 1: Internet address

Specifies the internet address for the host side of the virtual point-to-point Ethernet connection. Specify one of the following values:

***GEN** Specify *GEN to let the INSLNXSVR command configure a virtual point-to-point Ethernet port with a generated internet address.

character-value

Specify the host internet address for the virtual point-to-point Ethernet port in the form, xxx.yyy.zzz.nnn, where xxx, yyy, zzz, and nnn are decimal numbers ranging from 0 through 255. The internet address selected must be unique across all NWSO objects and the host TCP/IP configuration. The xxx.yyy.zzz value in the internet address must match the xxx.yyy.zzz value for the internet address of the Linux server side of the point-to-point Ethernet port.

Element 2: Linux internet address

Specifies the internet address for the Linux side of the virtual point-to-point Ethernet connection. Specify one of the following values:

***GEN** Specify *GEN to let the INSLNXSVR command configure a virtual point-to-point Ethernet port with a generated internet address.

character-value

Specify the Linux server internet address for the virtual point-to-point Ethernet port in the form, xxx.yyy.zzz.nnn, where xxx, yyy, zzz, and nnn are decimal numbers ranging from 0 through 255. The internet address selected must be unique across all NWSO objects and the host TCP/IP configuration. The xxx.yyy.zzz value in the internet address must match the xxx.yyy.zzz value for the internet address of the host side of the point-to-point Ethernet port.

Top

Examples

Example 1: Install Linux Using Non-iSCSI Hardware

```
INSLNXSVR NWSO(RHEL3MAR) RSRNAME(LIN03) LNXSVRDST(*RHEL4)
          SVRSTGSI(200 12000) LNGVER(2920)
          TEXT('RedHat 4 server')
```

This command will install a Linux server with a Red Hat 4 distribution on non-iSCSI hardware. It will contain a 12 gb system drive and exist on resource name LIN03. The name RHEL3MAR will be associated with the network server description on the iSeries side. The language version used will be 2920.

Example 2: Install Linux Using iSCSI Hardware

```
INSLNXSVR  NWSD(RHEL3MAR)  RSRcname(*iSCSI)  LNxsVRDST(*SLES8)
           SVRSTGSize(200 12000)  SVRSTGASP(2)  LNGVER(2920)
           SHUTDTIMO(2)  ACTTMR(300)  ENBUNICAST(*YES)
           SPINTNETA('9.5.2.32')  SPAUT(JSMITH ())
           TEXT('RedHat 4 server')
```

This command will install a Linux server with a SuSE 8 distribution on iSCSI hardware. It will contain a 12 gb which resides on user ASP 2. The name RHEL3MAR will be associated with the network server description on the iSeries side. The language version used will be 2920. The shutdown timeout will be set to 2 minutes. The system will wait 300 seconds for the connection to be made to the remote server's service processor. Unicast packet distribution will be used over iSCSI. The remote server's service processor internet address will be set to 9.5.2.32. SP authentication will be made with username JSMITH.

Top

Error messages

*ESCAPE Messages

NTA1007

Network server &1 must be varied off.

NTA1199

Vary on or off of the Linux server not successful.

NTA119A

Linux server installation not successful.

NTA1024

Storage space assigned to server &1 missing, damaged or not valid.

NTA1030

Internal error occurred.

Top

Install Program Temporary Fix (INSPTF)

Where allowed to run: All environments (*ALL)
 Threadsafte: 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
CPYPTF	Copy PTFs	*SRVATT, *YES, *NO	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 valid only 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 valid only if a tape or optical device name is specified on the DEV parameter. For optical devices, *UNLOAD is the only special value supported, *REWIND and *LEAVE will be ignored.

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

Copy PTFs (CPYPTF)

Specifies whether to copy PTF save files and cover letters into *SERVICE when PTFs are loaded. PTF save files must be in *SERVICE when distributing PTFs to other systems or when using the Save System Information (SAVSYSINF) command.

Note: This parameter is ignored if DEV(*SERVICE) or DEV(*NONE) is specified.

*SRVATT

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

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

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

Top

Examples

Example 1: Omitting PTFs

```
INSPTF LICPGM>(*ALL) DEV(*SERVICE) INSTYP(*IMMDLY)
      OMIT((5722999 MF12345 V5R4M0) (5722SS1 SI12345 V5R4M0))
```

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 V5R4M0)) DEV(TAP01) INSTYP(*IMMONLY)
HIPER(*YES)
```

This command will search the media for PTFs for the V5R4M0 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

Install Windows Server (INSWNTSVR)

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

Parameters
Examples
Error messages

The Install Windows Server (INSWNTSVR) command installs the Windows server Base Operating System on an Integrated xSeries server. INSWNTSVR also installs Integrated Server Support code on the server.

Windows server installation occurs in two steps. During the first step, the INSWNTSVR command creates all necessary objects to manage the server. This includes a network server description, message queue, line descriptions, storage spaces and TCP/IP interfaces.

For INSTYPE of *FULL, it also copies the install code for the Windows server Base Operating System from the Windows server installation media image.

During the second step of the Windows server installation, the Integrated xSeries server is varied on to start the Windows server installation.

For INSTYPE of *BASIC, the ServerGuide CD is inserted into the external servers locally attached CD-ROM or DVD drive and the server is varied on. At this point, the INSWNTSVR command ends. The ServerGuide CD in the server is booted from and continues with the process of configuring the server, including detecting and configuring any devices or adapters and preparing the system for the Windows installation.

Further Windows server installation is performed using the file server console and the normal Windows server install process.

When INSWNTSVR completes normally, the Windows server is left in a varied on state.

Restrictions:

1. You must have input/output system configuration (*IOSYSCFG), all object (*ALLOBJ) and job control (*JOBCTL) special authorities to run this command.
2. This command may require interactive input using the file servers console.
3. The file server must be varied off initially.
4. The file server will also be varied off and back on during the second step of the install as Windows server installs and requires the server to reboot.

Any errors that occur during the first step of configuring the file server will result in the failure of this command.

After this command is run, if you need to manage the different resources created, use the following commands:

- To check out the status of the Windows server, use the Work with Configuration Status command; WRKCFGSTS CFGTYPE(*NWS).
- To manage the server just installed, use the Work with Network Server Descriptions command; WRKNWSD NWSD(nwsdname).
- To manage the line descriptions created by this command, use the Work with Line Descriptions command; WRKLIND LIND(nwsdname*). The line descriptions are named using the network server name specified on the NWSD parameter of the INSWNTSVR command.

- To manage the TCP/IP interfaces created by this command, use the Work with TCP/IP Network Status (NETSTAT) command, option 1. Another option is to use the Configure TCP/IP (CFGTCP) command, option 1.
- To manage the network server configurations just created by this command, use the Work with NWS Configuration command; WRKNWSCFG NWS CFG(nwsdname*). The network server configurations are named using the network server name specified on the NWS D parameter of the INSWNTSVR command.

Top

Parameters

Keyword	Description	Choices	Notes
NWS D	Network server description	<i>Name</i>	Required, Key, Positional 1
INSTYPE	Installation type	*FULL, *BASIC	Required, Positional 2
RSRCNAME	Resource name	<i>Name</i> , *ISCSI	Required, Positional 3
WNTVER	Windows server version	<i>Character value</i>	Required, Positional 4
WNTSRCDIR	Windows source directory	<i>Path name</i> , *DFT	Optional
OPTION	Install option	<i>Character value</i>	Optional
TCPPTCFG	TCP/IP port configuration	Single values: *NONE Other values (up to 4 repetitions): <i>Element list</i>	Optional
	Element 1: Port	1, 2, 3, 4	
	Element 2: Windows internet address	<i>Character value</i>	
	Element 3: Windows subnet mask	<i>Character value</i>	
	Element 4: Windows gateway address	<i>Character value</i>	
VRTETHPORT	Virtual Ethernet port	Single values: *NONE Other values (up to 4 repetitions): <i>Element list</i>	Optional
	Element 1: Port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Windows internet address	<i>Character value</i>	
	Element 3: Windows subnet mask	<i>Character value</i>	
	Element 4: Associated port	<i>Name</i> , *NONE	
TCPDMNNAME	TCP/IP local domain name	<i>Character value</i> , *SYS	Optional
TCPNAMSVR	TCP/IP name server system	Single values: *SYS, *NONE Other values (up to 3 repetitions): <i>Character value</i>	Optional
MSGQ	Server message queue	Single values: *JOBLOG, *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Server message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
EVTLOG	Event log	Single values: *ALL, *NONE Other values (up to 3 repetitions): *SYS, *SEC, *APP	Optional

Keyword	Description	Choices	Notes
SVRSTGSIZE	Server storage space sizes	<i>Element list</i>	Optional
	Element 1: Install source size	500-2047, <u>*CALC</u>	
	Element 2: System size	1024-1024000, <u>*CALC</u>	
SVRSTGASP	Storage space ASP	<i>Element list</i>	Optional
	Element 1: Install source ASP	1-255, <u>1</u>	
	Element 2: System ASP	1-255, <u>1</u>	
STGASPDEV	Server storage ASP device	<i>Element list</i>	Optional
	Element 1: Install source ASP device	<i>Name</i>	
	Element 2: System ASP device	<i>Name</i>	
CVTNTFS	Convert to NTFS	<u>*YES</u> , *NO	Optional
TOWRKGRP	To workgroup	<i>Character value</i>	Optional
TODMN	To domain	<i>Character value</i>	Optional
FULNAM	Full Name	<i>Character value</i>	Optional
ORG	Organization	<i>Character value</i>	Optional
LNGVER	Language version	<i>Integer</i> , <u>*PRIMARY</u>	Optional
SYNCTIME	Synchronize date and time	<u>*YES</u> , *NO	Optional
PRPDMNUSR	Propagate domain user	<u>*YES</u> , *NO	Optional
WNTLICKEY	Windows license key	<i>Character value</i>	Optional
LICMODE	License mode	<i>Element list</i>	Optional
	Element 1: License type	<u>*PERSEAT</u> , *PERSERVER	
	Element 2: Client licenses	5-9999, <u>*NONE</u>	
	Element 3: Terminal services	<i>Character value</i>	
RSTDDEVRSC	Restricted device resources	Single values: <u>*NONE</u> , *ALL Other values (up to 10 repetitions): <i>Name</i> , *ALLTAPE, *ALLOPT	Optional
SHUTDTIMO	Shutdown timeout	2-45, <u>15</u>	Optional
ACTTMR	Activation timer	30-1800, <u>120</u>	Optional
CMNMSGQ	Communications message queue	Single values: <u>*SYSOPR</u> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Communications message queue	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
STGPTH	Storage path	<i>Element list</i>	Optional
	Element 1: Network server host adapter	<i>Communications name</i>	
VRTETHPTH	Virtual Ethernet path	Values (up to 5 repetitions): <i>Element list</i>	Optional
	Element 1: Port	<u>*VRTETHPTP</u> , *VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 2: Network server host adapter	<i>Communications name</i>	
SHUTDPORT	Shutdown TCP port	1024-65535, <u>8700</u>	Optional
VRTETHCTLP	Virtual Ethernet control port	1024-65535, <u>8800</u>	Optional
RMTNWSCFG	Remote system NWSCFG	<i>Name</i> , <u>*DFT</u>	Optional
SPNWSCFG	Service processor NWSCFG	<i>Name</i> , <u>*DFT</u>	Optional

Keyword	Description	Choices	Notes
CNNNWSCFG	Connection security NWSCFG	Name, <u>*DFT</u>	Optional
DFTSECRULE	Default IP security rule	Character value, <u>*NONE</u> , *GEN	Optional
IPSECRULE	IP security rule	1-16, <u>*DFTSECRULE</u> , *NONE	Optional
INZSP	Initialize service processor	<u>*MANUAL</u> , *SYNC, *AUTO, *NONE	Optional
ENBUNICAST	Enable unicast	<u>*NO</u> , *YES	Optional
EID	Enclosure identifier	Single values: *AUTO Other values: <i>Element list</i>	Optional
	Element 1: Serial number	<i>Character value</i>	
	Element 2: Manufacturer type and model	<i>Character value</i>	
SPNAME	Service processor name	<i>Character value</i> , <u>*SPINTNETA</u>	Optional
SPINTNETA	SP internet address	<i>Character value</i>	Optional
SPAUT	SP authentication	Single values: *DFT Other values: <i>Element list</i>	Optional
	Element 1: User name	<i>Character value</i>	
	Element 2: User password	<i>Character value</i>	
SPCERTID	SP certificate identifier	Single values: <u>*NONE</u> Other values: <i>Element list</i>	Optional
	Element 1: Component	*COMMONNAME, *EMAIL, *ORGUNIT	
	Element 2: Compare value	<i>Character value</i>	
RMTSYSID	Remote system identifier	Single values: <u>*EID</u> Other values: <i>Element list</i>	Optional
	Element 1: Serial number	<i>Character value</i>	
	Element 2: Manufacturer type and model	<i>Character value</i>	
DELIVERY	Delivery method	<i>Character value</i> , <u>*DYNAMIC</u> , *MANUAL	Optional
CHAPAUT	CHAP authentication	Single values: *NONE Other values: <i>Element list</i>	Optional
	Element 1: CHAP name	<i>Character value</i>	
	Element 2: CHAP secret	<i>Character value</i>	
BOOTDEVID	Boot device ID	Single values: <u>*SINGLE</u> Other values: <i>Element list</i>	Optional
	Element 1: Bus	0-255	
	Element 2: Device	0-31	
	Element 3: Function	0-7	
DYNBOOTOPT	Dynamic boot options	<i>Element list</i>	Optional
	Element 1: Vendor ID	<i>Character value</i> , <u>*DFT</u>	
	Element 2: Alternate client ID	<i>Character value</i> , <u>*ADPT</u>	

Keyword	Description	Choices	Notes
RMTIFC	Remote interfaces	<i>Element list</i>	Optional
	Element 1: SCSI interface	<i>Element list</i>	
	Element 1: Adapter address	<i>Hexadecimal value</i>	
	Element 2: Internet address	<i>Character value</i>	
	Element 3: Subnet mask	<i>Character value</i>	
	Element 4: Gateway address	<i>Character value</i>	
	Element 5: iSCSI qualified name	<i>Character value, *GEN</i>	
	Element 2: LAN interface	<i>Element list</i>	
	Element 1: Adapter address	<i>Hexadecimal value</i>	
	Element 2: Internet address	<i>Character value</i>	
	Element 3: Subnet mask	<i>Character value</i>	
	Element 4: Gateway address	<i>Character value</i>	
TEXT	Text 'description'	<i>Character value, *BLANK</i>	Optional
KBDTYPE	Keyboard layout	<i>Hexadecimal value, *DEFAULT</i>	Optional
CFGFILE	Configuration file	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Configuration file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
CLU	Cluster name	<i>Name, *NONE</i>	Optional
CLUCFG	Cluster configuration	Single values: *CLU Other values: <i>Element list</i>	Optional
	Element 1: Cluster domain name	<i>Character value</i>	
	Element 2: Quorum resource size	550-1024000, *CALC	
	Element 3: Quorum resource ASP	1-255, 1	
	Element 4: Quorum ASP device	<i>Name</i>	
	Element 5: Connection port	*VRTETH0, *VRTETH1, *VRTETH2, *VRTETH3, *VRTETH4, *VRTETH5, *VRTETH6, *VRTETH7, *VRTETH8, *VRTETH9	
	Element 6: Cluster internet address	<i>Character value</i>	
	Element 7: Cluster subnet mask	<i>Character value</i>	
VRTPTPORT	Virtual PTP Ethernet port	<i>Element list</i>	Optional
	Element 1: Internet address	<i>Character value, *GEN</i>	
	Element 2: Windows internet address	<i>Character value, *GEN</i>	

Top

Network server description (NWSD)

Specifies the name of the network server to be installed.

This is a required parameter.

The network server name can be up to eight characters. The following characters are allowed in NWSD names:

- Alphabetical characters A through Z
- Digits 0 through 9

The network server description specified is created using the values specified in this command. The Create Network Server Description (CRTNWSD) command will be used to create the NWSD. The name is also used as the computer name as well as the TCP host name for the Windows server.

communications-name

Specify the name of the network server description.

Top

Installation type (INSTYPE)

Specifies the type of install to perform.

This is a required parameter.

***FULL** A full installation of the file server will be controlled by the system.

Note: Any file server that will be upgrading, OPTION(*UPGRADE), to a new version of Windows must also specify *FULL.

***BASIC**

A basic installation of the external file server will be initiated by the system and completed using the ServerGuide CD.

Top

Resource name (RSRCNAME)

Specifies the resource name that identifies the hardware that the description uses.

This is a required parameter.

***ISCSI**

This network server resource is an iSCSI attached server using a network server host adapter device. Use the Work with Device Descriptions (WRKDEVVD) with *NWSH specified for the DEVVD parameter to help determine which Network Server Host Adapters are configured.

name Specify the resource name of the File Server IOA communications adapter to use. Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name.

Top

Windows server version (WNTVER)

Specifies the version of Windows server to install on this network server. You can use F4 while prompting the command to see the complete list of allowed values for this parameter.

This is a required parameter.

***WIN2000**

Support for Windows 2000 Server and Advanced Server versions.

*WIN2003

Support for Windows Server 2003 Web, Standard and Enterprise versions.

Top

Windows source directory (WNTSRCDIR)

Specifies the path name of the directory associated with the Windows server CD-ROM image that is used as the source for the install. The directory name may reference an optical volume ('/QOPT/volume'), a folder ('/QDLS/folder'), or an IFS directory ('/dir1/dir2'). An example of a CD-ROM volume path name would be '/QOPT/W2ASEL_EN'.

To find out the name of a volume on an optical device, use the command: DSPOPT VOL(*MOUNTED) DEV(device-name). If you do not know the name of the optical device, use the command: WRKCFGSTS CFGTYPE(*DEV) CFGD(*OPT)

To find out the name of a path in an IFS directory, use the Work with Object Links (WRKLNK) command. WRKLNK will show the directory object path names on the system.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

***DFT** The default directory path name to use is determined by searching the QOPT file system (/QOPT directory in IFS). The path name for the first volume in QOPT that contains an I386 directory with a WINNT.EXE file in it is used as the default.

Note: Using *DFT on a system with an optical media library is not recommended. Every volume located in the '/QOPT' path will be searched for a valid Windows install source and will cause each volume to be retrieved and mounted. A specific optical volume path should be specified.

path-name

Specify the Windows server source path name to use for the install.

Note: The specified directory must contain an I386 directory with a WINNT.EXE file in it.

You can use F4 while prompting the command to see a list of path names for optical volumes that are allowed values for this parameter. Folder or IFS path names that are not optical volumes are also allowed, but are not listed when using F4.

Top

Install option (OPTION)

Specifies the Windows server installation method. More information on each of type of install is available in the Windows server documentation and SETUP.TXT files. This information should be consulted prior to upgrading Windows server.

***INSTALL**

Install a new Windows server and the Integrated Server Support code.

This will create a new network server description, storage spaces, message queue, line descriptions, and TCP interfaces.

***UPGRADE**

Upgrade an existing Windows server and the Integrated Server Support code for a later release of Windows server.

This will use an existing network server description, system storage space, message queue, line descriptions, and TCP interfaces.

Notes:

1. A backup of all drives linked to the network server is strongly recommended before an *UPGRADE install is performed.
2. The install source drive (typically the D: drive) is deleted and recreated with the install source size specified on the **Server storage space sizes (SVRSTGSIZE)** parameter, **Install source size** element. Any user data on this drive will be lost!
3. An *UPGRADE install is the only supported upgrade path to a new release of Windows server. Upgrading the Windows server directly from an installation CD-ROM may cause the file server to become unusable and require that it be restored from a backup.

Top

TCP/IP port configuration (TCPPORTCFG)

Specifies the Windows TCP/IP configuration values that are specific to a port on the network server. This information consists of four parts including the identification of the network server port, the internet address, the subnet mask and the default gateway assigned to the port.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Single values

*NONE

Specifies that there is no Windows TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the network server port number to be configured. Specify one of the following values:

- 1 Network server port number 1 is configured.
- 2 Network server port number 2 is configured.
- 3 Network server port number 3 is configured.
- 4 Network server port number 4 is configured.

Element 2: Windows internet address

internet-address

Specify the Windows internet address for the port.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 3: Windows subnet mask

subnet-mask

Specifies the subnet mask for the Windows internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Windows gateway address

gateway-address

Specifies the default gateway address for the Windows internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual Ethernet port (VRTETHPORT)

Specify the TCP/IP configuration for the virtual Ethernet used by the file server.

Single values

*NONE

Specifies that there is no Windows TCP/IP port configuration.

Other values (up to 4 repetitions)

Element 1: Port

Specifies the network server virtual Ethernet port number to be configured. Specify one of the following values:

*VRTETHn

The network server virtual Ethernet port 'n' is configured, where 'n' has a value of 0 through 9.

Element 2: Windows internet address

internet-address

Specify the Windows internet address for the port.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 3: Windows subnet mask

subnet-mask

Specifies the subnet mask for the Windows internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Associated port

Specifies the resource name that describes the port that is used to establish a connection between a Windows network server and the network.

Note: Use the Work with Hardware Resources (WRKHDWRSC) command with *CMN specified for the TYPE parameter to help determine the resource name. The resource name is on the port. For example, the resource name may be CMN01 on a Ethernet port.

***NONE**

An associated port resource name is not associated with the line.

name Specify the associated port resource name.

Top

TCP/IP local domain name (TCPDMNNAME)

Specifies the local domain name associated with the network server.

A domain name can be a text string having 2 to 255 characters. Domain names consist of one or more labels separated by periods. Each label can contain up to 63 characters. The following characters are allowed in domain names:

- Alphabetical characters A through Z
- Alphabetical characters a through z
- Digits 0 through 9
- Minus sign (-)
- Period (.). Periods are only allowed when they separate labels of domain style name (refer to RFC 1034).

Uppercase and lowercase characters are allowed, but no significance attached to the case. The case is maintained as entered. The first and last character of the host name must be an alphabetic character or a digit.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

***SYS** Specifies that the local domain name for the network server should be the same value as is configured for the system.

character-string

Specify a TCP domain name to be associated with the network server.

Top

TCP/IP name server system (TCPNAMSVR)

Specifies the internet address of the name server system that is used by the network server. Typically, this is the same value as it is for the system.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Single values

***SYS** The name server system used by the network server should be the same as for the system.

***NONE**

There is no name server to be used by the network server.

Other values (up to 3 repetitions)

internet-address

Specify an internet address for the name server system to be used by the network server. Up to three remote name server systems can be specified. The name server systems are used in the order they are specified.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Server message queue (MSGQ)

Specifies the name of a message queue to receive server messages.

Single values

*JOBLOG

Causes Windows server event log messages from the server and informational messages to be placed on the joblog of the user administration monitor job. Errors requiring operator intervention are sent to the QSYSOPR message queue.

*NONE

Causes Windows server event log messages and informational messages to not be placed on any message queue. Errors requiring operator intervention are sent to the QSYSOPR message queue.

Qualifier 1: Server message queue

message-queue-name

Specify the name of a message queue to receive messages issued by the server, informational messages and errors requiring operator intervention.

This message queue should be monitored so that it does not become full. If it becomes full, messages will be rerouted to the joblog of the user administration job.

Care should be taken if QSYSOPR is specified as the message queue to receive all messages related to Windows server running on an Integrated xSeries server because the volume of Windows server event log messages is unpredictable.

If both a message queue name and library name is specified, and the message queue does not exist, the message queue will automatically be created with authority *EXCLUDE. If the library specified for the message queue does not exist, the command will fail.

The name of the message queue can be qualified by one of the following library values:

Qualifier 2: Library

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

*CURLIB

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

name Specify the name of the library to be used.

Top

Event log (EVTLOG)

Specifies whether or not messages from the event logs are received from the server.

Note: Event log messages are placed in the message queue that is identified by the **Server message queue (MSGQ)** parameter. The MSGQ value cannot be *NONE. See the MSGQ parameter for more information.

Single values

***ALL** All event log messages are received.

***NONE**

No event log messages are received.

Other values (up to 3 repetitions)

***SYS** The system event log messages are received.

Note: This value can only be specified once.

***SEC** The security event log messages are received.

Note: This value can only be specified once.

***APP** The application event log messages are received.

Note: This value can only be specified once.

Top

Server storage space sizes (SVRSTGSIZE)

Specifies the size of the server storage spaces, in megabytes.

Element 1: Install source size

Specifies the size of the storage space that holds the files used to install the Windows server.

Notes:

1. The contents of the I386 directory of the Windows server installation media and the Integrated Server Support code are copied to the install source drive. The size specified for the install source drive must be large enough to hold this data.
2. For an OPTION(*UPGRADE) install, a new install source drive size may be specified. The new drive that is created will replace the existing install source drive (typically the D: drive) resulting in the loss of all user data on this drive. A backup of all drives associated with this network server is recommended.

***CALC**

Specifies that the size should be calculated based on the space required to hold the install source located by the **Windows source directory (WNTSRCDIR)** parameter.

integer

Specifies the install source size value in megabytes. The size must be at least 500 megabytes but no larger than 2047 megabytes.

Element 2: System size

Specifies the size in megabytes of the storage space that the Windows server operating system is installed on.

Notes:

1. The minimum size allowed is determined by the Windows server version specified on the WNTVER parameter.
2. A value other than the default for the System size may not be specified when OPTION is *UPGRADE.

*CALC

Specifies that the size should be calculated based on the installed memory of the Integrated xSeries server (where it can be determined) and other elements such as Windows version requirements.

integer

Specifies the system size value, in megabytes up to a maximum of 1024000.

Top

Storage space ASP (SVRSTGASP)

Specifies the auxiliary storage pool (ASP) identifiers for the storage space that will contain the files used to install the Windows server and the storage space that will contain the Windows server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: Null (omitted) values are specified with the characters *N, which mean that no value was specified. The value specified for the corresponding Server storage ASP device (STGASPDEV) element will be used if specified. Otherwise, the default value will be used. *N is needed only when another value following the omitted element is being specified.

Element 1: Install source ASP

Specifies the auxiliary storage pool for the storage space that holds the files used to install the Windows server.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

integer

Specify a value ranging from 2 through 255 for the ASP identifier. Valid values depend on how many ASPs are defined on the system.

Element 2: System ASP

Specifies the auxiliary storage pool for the storage space that holds the Windows server operating system.

Note: A value other than the default for the System size may not be specified when OPTION is *UPGRADE.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

integer

Specify a value ranging from 2 through 255 for the ASP identifier. Valid values depend on how many ASPs are defined on the system.

Top

Server storage ASP device (STGASPDEV)

Specifies the auxiliary storage pool (ASP) device name for the storage space that will contain the files used to install the Windows server and the storage space that will contain the Windows server operating system.

Note: You cannot specify both a SVRSTGASP and STGASPDEV parameter value for the same element.

Note: The ASP must have been activated (by varying on the ASP device) and have a status of 'Available'.

Element 1: Install source ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the files used to install the Windows server.

name The device name of the ASP to use for the network server storage space.

Element 2: System ASP device

Specifies the independent auxiliary storage pool device name for the storage space that holds the Windows server operating system.

name The device name of the ASP to use for the network server storage space.

Top

Convert to NTFS (CVTNTFS)

Specifies if the primary partition of the system drive should be converted to the NT File System (NTFS).

Note: Conversion to NTFS may be automatically performed under some situations such as the File system limitations for system drives greater than 32000 megabytes. When one of these conditions exist, the Convert to NTFS (CVTNTFS) parameter is automatically set to *YES by this command.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

***YES** Windows server will convert the system drive to NTFS during the install.

***NO** Windows server will not convert the system drive to NTFS during the install.

Top

To workgroup (TOWRKGRP)

Specifies the workgroup in which the computer will participate. The Windows server will prompt for a value during the install if no parameter value is specified.

Characters allowed for this parameter include any upper-case character A-Z, lower-case a-z, 0-9, and any character in the ASCII code page 850 except:

- space ()
- quote (")
- asterisk (*)
- plus (+)
- comma (,)
- period (.)
- forward slash (/)
- colon (:)
- semicolon (;)
- less-than (<)
- equal (=)
- greater-than (>)

- question mark (?)
- open square bracket ([)
- backward slash (\)
- close square bracket (])
- vertical bar (|)

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

character-string

Specify the workgroup in which the computer will participate. The computer can be part of either a workgroup or a domain.

Top

To domain (TODMN)

Specifies the name of an existing server domain in which the computer will participate. Windows server will prompt for a value during the install if no parameter value is specified.

Characters allowed for this parameter include any upper-case character A-Z, lower-case a-z, 0-9, and any character in the ASCII code page 850 except:

- space ()
- quote (")
- asterisk (*)
- plus (+)
- comma (,)
- period (.)
- forward slash (/)
- colon (:)
- semicolon (;)
- less-than (<)
- equal (=)
- greater-than (>)
- question mark (?)
- open square bracket ([)
- backward slash (\)
- close square bracket (])
- vertical bar (|)

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

character-string

Specify the name of the server domain in which the computer will participate. The computer can be part of either a workgroup or a domain.

Top

Full Name (FULNAM)

Specifies the users' full name for the Windows server being installed. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

character-string

Specify the name of the user installing Windows server who holds the license.

Top

Organization (ORG)

Specifies the organization name for the Windows server being installed. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

character-string

Specify the name of the organization installing Windows server who holds the license.

Top

Language version (LNGVER)

Specifies the installed language environment used to display Integrated Server Support text and messages. Note that a smaller set of languages are available for messages displayed during installation than for messages displayed after installation.

*PRIMARY

The installed language environment for Integrated Server Support text and messages is based on the language feature of the system's primary language.

language-version

Specifies the language feature that will be used to select the Integrated Server Support text and messages.

Top

Synchronize date and time (SYNCTIME)

Specifies whether the system should synchronize the network server date and time with the host system date and time.

*YES The system synchronizes the file server date and time with the host system date and time at every vary on and at least every 30 minutes thereafter.

The QTIMZON system value must be set to the correct value for time synchronization to work correctly.

*NO The system synchronizes the file server date and time with the host system date and time when the network server description is varied on, but will not keep the date and time synchronized while the network server description is varied on.

Top

Propagate domain user (PRPDMNUSR)

Specifies if this server should be used to propagate and synchronize users to the Windows domain or active directory.

Note: When multiple network servers belong to the same Windows domain, only one needs to propagate users to the domain. Selecting a network server with a domain role of *DMNCTL will provide the fastest performance and may eliminate the need for the special QAS400NT userid. At least one network server should specify *YES for each Windows domain that you wish to propagate users to.

***YES** Send user updates to the Windows domain or active directory through this server.

***NO** Do not send user updates to the Windows domain or active directory through this server.

Top

Windows license key (WNTLICKEY)

Specifies the license key for Windows server. Windows server will prompt for a value during the install if no parameter value is specified.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

character-string

Specify the license key on the back of the jewel case that the Windows server CD came in, including any dashes ('-'). The length of the Windows license key (WNTLICKEY) is limited to 34 characters.

Top

License mode (LICMODE)

Determines the license mode to install the Windows server as.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

Element 1: License type

***PERSEAT**

Indicates that the end user purchased a client access license for each computer accessing the server.

***PERSERVER**

Indicates that the end user purchased client access licenses for each server, which allows a certain number of concurrent connections to the server.

Element 2: Client licenses

***NONE**

Indicates that no client licenses are installed. *NONE must be specified when *PERSEAT is specified.

integer

Number of client licenses purchased for the server being installed. A number greater than five must be specified when *PERSERVER is specified as the license type. The valid range is from 5 to 9999.

Element 3: Terminal services

***NONE**

Do not install the Terminal Server component for this server.

***TSENABLE**

Installs Terminal Services on Windows 2000.

***PERDEVICE**

Installs and configures the Windows Server 2003 Terminal Services to require that each connected device has a valid Terminal Server Client Access License (CAL). If the client has a Terminal Server CAL, it can access more than one Terminal Server.

***PERUSER**

Installs and configures Windows Server 2003 Terminal Server to provide one Terminal Server CAL for each active user.

Top

Restricted device resources (RSTDDEVRS)

Specifies the system's optical and tape device resource names that are restricted and cannot be used by the Windows server.

Note: Only tape and optical device resources can be restricted.

Single values

***NONE**

No device resources are restricted from the network server. Therefore, any tape or optical device resources that exist on the system can be used.

***ALL** All tape and optical resources are restricted from being used by the network server.

Other values (up to 10 repetitions)

***ALLOPT**

All optical resources are restricted from being used by the network server.

Note: This value can only be specified once.

***ALLTAPE**

All tape resources are restricted from being used by the network server.

Note: This value can only be specified once.

name Specify the restricted device resource name that cannot be used by the network server.

Top

Shutdown timeout (SHUTDTIMO)

Specifies the server shutdown time-out value in minutes. This is used to limit the amount of time that the servers operating system is allowed to shutdown before the server is varied offline.

15 The network server default shutdown time-out value is used.

integer

Specify the time (in minutes) to wait. Valid values range from 2 through 45. The system waits until the network servers operating system has shutdown successfully, or until the specified time passes before varying the network server offline.

Top

Activation timer (ACTTMR)

Specifies the amount of time (in seconds) the system will wait for the connection to be established to the remote server's service processor and to power on the remote server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

120 The activate time of 120 seconds is used.

integer

Specify, in seconds, a value ranging from 30 through 1800.

Top

Communications message queue (CMNMSGQ)

Specifies the name of a message queue to receive communications status messages.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Single values

*SYSOPR

Causes messages to be placed in the system operator message queue (QSYSOPR message queue in library QSYS).

Qualifier 1: Communications message queue

name Specify the name of a message queue to receive communications status messages.

Qualifier 2: Library

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

*CURLIB

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

name Specify the name of the library to be used.

Top

Storage path (STGPTH)

Specifies the storage path the storage spaces can use.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Element 1: Network server host adapter

communications-name

Specify the name of an existing network server host adapter (NWSH) device.

Top

Virtual Ethernet path (VRTETHPTH)

Specifies the virtual Ethernet paths the Ethernet line descriptions can use. This information consists of two parts including the virtual Ethernet port and the network server host adapter (NWSH) device. You must enter at least one virtual Ethernet path which is the path to be used by the *VRTETHPTP line description.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

You can specify 5 values for this parameter.

Element 1: Port

***VRTETHPTP**

The network server virtual Ethernet point to point port is configured.

***VRTETH0**

Virtual Ethernet port 0 is configured.

***VRTETH1**

Virtual Ethernet port 1 is configured.

***VRTETH2**

Virtual Ethernet port 2 is configured.

***VRTETH3**

Virtual Ethernet port 3 is configured.

***VRTETH4**

Virtual Ethernet port 4 is configured.

***VRTETH5**

Virtual Ethernet port 5 is configured.

***VRTETH6**

Virtual Ethernet port 6 is configured.

***VRTETH7**

Virtual Ethernet port 7 is configured.

***VRTETH8**

Virtual Ethernet port 8 is configured.

***VRTETH9**

Virtual Ethernet port 9 is configured.

Element 2: Network server host adapter

communications-name

Specify the name of an existing network server host adapter (NWSH) device. The network server host adapter name does not need to be unique for each VRTETHPTH parameter on this NWSH.

Shutdown TCP port (SHUTDPORT)

Specifies the TCP port to use for shutdown.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

8700 Use the TCP port number of 8700.

integer

Specify the port number identifying the port that is to be used for shutdown. Valid values range from 1024 through 65,535.

Top

Virtual Ethernet control port (VRTETHCTLP)

Specifies the TCP port to use for virtual Ethernet control.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

8800 Use the TCP port number of 8800.

integer

Specifies the port number identifying the port that is to be used for virtual Ethernet control. Valid values range from 1024 through 65,535.

Top

Remote system NWSCFG (RMTNWSCFG)

Specifies the remote system network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

*DFT Use the system generated default remote system network server configuration name of 'nwsdnameRM' where nwsdname is the name of the network server description.

name Specify the name of an existing remote system network server configuration.

Top

Service processor NWSCFG (SPNWSCFG)

Specifies the service processor network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

*DFT Use the system generated default service processor network server configuration name of 'nwsdnameSP' where nwsdname is the name of the network server description.

name Specify the name of an existing service processor network server configuration.

Connection security NWSCFG (C>NNNWSCFG)

Specifies the connection security network server configuration to use with this server.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

***DFT** Use the system generated default connection security network server configuration name of 'nwsdnameCN' where nwsdname is the name of the network server description.

name Specify the name of an existing connection security network server configuration.

Default IP security rule (DFTSECRULE)

Specifies the default IP Security (IPSec) rule used between the hosting and remote system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the connection security network server configuration specified for the **Connection security NWSCFG (C>NNNWSCFG)** parameter already exists.

A value other than *DFTSECRULE must be specified on the IP security rule (IPSECRULE) parameter in this case.

***NONE**

IP Security (IPSec) protocol security settings are not configured.

***GEN** Generate a random pre-shared key.

character-string

Specify the pre-shared key.

A pre-shared key is a nontrivial string up to 32 characters long.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

IP security rule (IPSECRULE)

Specify the relative entry of an existing IP security rules (IPSECRULE) parameter, defined in the existing connection security network server configuration that will be used as the initial IP security setting between the hosting and remote system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

*DFTSECRULE

Use the value specified on the **Default IP security rule (DFTSECRULE)** parameter.

Note: This value is not valid when the **Connection security NWSCFG** specified on the **CNNNWSCFG** parameter already exists.

*NONE

Remote interface will not use any security rule.

1-16 Remote interface will use security rule specified

Initialize service processor (INZSP)

Specifies how the remote system's service processor is secured.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Use this only if the interconnecting network is physically secure.

Note: Some service processors do not support secure connections. Use *NONE for these service processors. Additional information can be found at Integrated xSeries solutions at <http://www.ibm.com/servers/eserver/series/integratedxseries>.

*MANUAL

Security parameters are manually configured on remote system's service processor. *MANUAL provides the highest security.

To use this option, it is required that the remote system's service processor is pre-configured with a user name, password and certificate. Certificate management will be required. This method is appropriate when connecting to the service processor via public networks to protect the password.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

*AUTO

Parameters are automatically configured on the remote system's service processor.

*AUTO provides security without requiring pre-configuration of the remote system's service processor. The remote system's service processor will have certificates automatically regenerated when the certificates are near expiration. This option is appropriate if the interconnecting network is physically secure or is protected by a firewall.

Note: An administrator will need to regenerate the certificate using the Initialize NWS Configuration (INZNWSCFG) command when the service processor certificate has expired, or if a new certificate and password are desired at any time before the certificate expires.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***SYNC**

Synchronize the self-signed certificate from the service processor. This option is used if multiple service processor network server configurations are used for the same system or the service processor network server configuration has been restored from backup and the service processor's certificate must be synchronized. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

***NONE**

Provides no security.

Use this only if the interconnecting network is physically secure.

Note: Some service processors do not support secure connections. Use *NONE for these service processors. Additional information can be found at Integrated xSeries solutions at <http://www.ibm.com/servers/eserver/series/integratedxseries>.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Top

Enable unicast (ENBUNICAST)

Specifies whether unicast packet distribution is to be used. Unicast is a transmission method where packets are sent directly to the specified **Service processor name (SPNAME)** or **SP internet address (SPINTNETA)** parameter.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***NO** Disable unicast

***YES** Enable unicast.

Top

Enclosure identifier (EID)

Specifies the identifying serial number, type and model of the enclosure containing the service processor.

When specified, they are used to locate the system on the network.

Look for these values on the label of the system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Single values

***AUTO**

Automatically retrieve the identifier when ENBUNICAST(*YES) is specified.

Element 1: Serial number

character-string

Specify the machine serial number.

Element 2: Manufacturer type and model

character-string

Specify the machine type and model.

The value is entered in the form tttmmm where ttt is the machine type and mmm is the machine model number.

Top

Service processor name (SPNAME)

Specifies the remote system's service processor host name.

Note: This parameter is required when ENBUNICAST(*YES) is specified.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

***SPINTNETA**

The remote system is identified by the value specified for the **SP internet address (SPINTNETA)** parameter.

host-name

Specify the remote system's service processor host name.

Top

SP internet address (SPINTNETA)

Specifies the remote system's service processor internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Notes:

1. This parameter is ignored when ENBUNICAST(*NO) is specified.
2. This parameter is required when SPNAME(*SPINTNETA) is specified.

internet-address

Specify the internet address of the service processor.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password. This is used to authenticate and secure the service processor.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

Note: This value is only valid when INZSP(*NONE) is specified on this command or in the existing network server configuration.

Single values

*DFT The default service processors userid and password are used.

Element 1: User name**character-string**

Specify a name that represents the host configuration that owns the service processor. It is suggested that the remote system network server configuration name be used. If multiple remote system network server configurations use the same service processor at different times, each configuration must contain the same user name and password.

Element 2: User password**character-string**

Specify the service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

SP certificate identifier (SPCERTID)

The SP certificate identifier specifies one of three possible fields that identifies the service processor's certificate.

This parameter is specified to provide additional validation that the certificate is from the service processor. The contents of the selected field must exactly match the value of the field that was entered when the certificate was generated or requested from a certificate authority.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the service processor network server configuration specified for the **Service processor NWSCFG (SPNWSCFG)** parameter already exists.

This parameter is required when INZSP(*MANUAL) is specified and cannot have the value *NONE.

Single values

*NONE

Service processor certificate is not configured.

Element 1: Component

*COMMONNAME

Selects the certificate's common name specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "ASM Domain Name" field used to generate a self-signed certificate or generate a certificate signing request.

*EMAIL

Selects the certificate's e-mail address specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "Email Address" field used to generate a self-signed certificate or generate a certificate signing request.

*ORGUNIT

Selects the certificate's organizational unit specified when the certificate was generated or requested from a certificate authority. On the remote supervisor adapter II this correlates to the "Organizational Unit" field used to generate a self-signed certificate or generate a certificate signing request.

Element 2: Compare value

character-string

Specify the certificates component compare value. Enter no more than 255 characters of text, enclosed in apostrophes.

Top

Remote system identifier (RMTSYSID)

Specifies the identifying serial number, type and model of the remote system. When specified, they are used to locate the remote system on the network.

Look for these values on the label of the system.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

*EID Use the service processor identifier.

Element 1: Serial number

character-string

Specify the machine serial number.

Element 2: Manufacturer type and model

character-string

Specify the machine type and model.

Top

Delivery method (DELIVERY)

Specifies how the parameters necessary to configure the remote system are delivered.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

*DYNAMIC

Parameters are dynamically delivered to the remote system using Dynamic Host Configuration Protocol (DHCP).

*MANUAL

Parameters are manually configured on the remote system using the BIOS utilities (System BIOS or Adapter BIOS - CTRL-Q).

Top

CHAP authentication (CHAPAUT)

Specifies the Challenge Handshake Authentication Protocol (CHAP) for the host system iSCSI target to authenticate the remote system initiator node.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

*NONE

CHAP authentication is not enabled.

Element 1: CHAP name

*NWSCFG

The system will automatically generate a name for CHAP using the Network server configuration name.

character-string

Specify the name you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent

- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Element 2: CHAP secret

***GEN** The system will automatically generate a random CHAP secret.

character-string

Specify the secret you want to use for the Challenge Handshake Authentication Protocol.

Valid characters are upper case A through Z, lower case a through z, numbers 0 through 9, and the following special characters:

- Plus sign
- Equal sign
- Percent
- Ampersand
- Left parenthesis
- Right parenthesis
- Comma
- Underline
- Minus sign
- Period
- Colon
- Semicolon

Top

Boot device ID (BOOTDEVID)

Specifies the PCI Function Address (Bus/Device/Function) of the iSCSI adapter in the remote system that will be used to boot from.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Single values

***SINGLE**

The single iSCSI adapter is used on the remote system

Element 1: Bus

number

Specify the bus number of the remote system's iSCSI adapter that will be used to boot.
Valid values range from 0 through 255.

Element 2: Device

number

Specify the device number of the remote system's iSCSI adapter that will be used to boot.
Valid values range from 0 through 31.

Element 3: Function

number

Specify the function number of the remote system's iSCSI adapter that will be used to boot.
Valid values range from 0 through 7.

Top

Dynamic boot options (DYNBOOTOPT)

Specifies the internal Dynamic Host Configuration Protocol (DHCP) Server configuration.

Note: This is an advanced configuration function.

This parameter is used to configure the internal DHCP Server that is part of the iSCSI Target Host Bus Adapter firmware. It is used to provide IP address and diskless boot parameters for the remote iSCSI Initiator.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

This parameter is only valid when DELIVERY(*DYNAMIC) is specified.

Element 1: Vendor ID

The client and server are pre-configured to a default vendor ID. Network administrators can configure clients to define their own identifying values to convey hardware, operating system or other identifying information. DHCP option 60 described in the IETF RFC 2132 is used for this function.

***DFT** The default vendor ID will be used.

character-string

Vendor ID of the remote system's iSCSI adapter that will be used.

Element 2: Alternate client ID

Used by clients to specify their unique identifier to the server. Each client's identifier must be unique among all other client identifiers used on the effective DHCP network to which the client is attached (that is, the client's local subnet and any remote subnets reachable using DHCP relay). Vendors and system administrators are responsible for choosing client identifiers that meet this requirement for uniqueness. DHCP option 61 described in the IETF RFC 2132 is used for this function.

***ADPT**

The default Client ID consists of the adapter address for the remote system's iSCSI adapter. This value will be used to identify the remote system.

character-string

Specify the Client ID of the remote system's iSCSI adapter that will be used to boot.

Top

Remote interfaces (RMTIFC)

Specifies the remote system's interfaces. This information is used to identify and configure the remote system's interfaces. Each adapter has two functions to support a SCSI and a LAN interface.

Note: This parameter is only valid when *ISCSI is specified for the **Resource name (RSRCNAME)** parameter.

Note: This parameter is ignored when the remote system network server configuration specified for the **Remote system NWSCFG (NWSCFG)** parameter already exists.

Element 1: SCSI interface

Specifies the remote system's SCSI interfaces.

Element 1: Adapter address

***NONE**

No SCSI interface is configured for this adapter.

adapter-address

Specify the 12-character hexadecimal adapter address for the remote system's iSCSI interface.

Element 2: Internet address

internet-address

Specify the internet address for the remote system's SCSI interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

subnet-mask

Specify the subnet mask for the remote system's SCSI interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

gateway-address

Specify the gateway address for the remote system's SCSI interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 5: iSCSI qualified name

***GEN** The system will automatically generate the iSCSI qualified name.

name Specify the iSCSI qualified name for the remote system's SCSI interface.

The following characters are allowed in iSCSI qualified names:

- Alphabetical characters A through Z converted to lower case (refer to RFC 3722)
- Alphabetical characters a through z
- Digits 0 through 9
- Period (.)
- Dash (-)
- colon (:)

Element 2: LAN interface

Specifies the remote system's LAN interfaces.

Element 1: Adapter address

***NONE**

No LAN interface is configured for this adapter.

adapter-address

Specify the 12-character hexadecimal adapter address for the remote system's LAN or TCP Offload Engine (TOE) interface.

Element 2: Internet address

internet-address

Specify the internet address for the remote system's LAN interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 3: Subnet mask

subnet-mask

Specify the subnet mask for the remote system's LAN interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Element 4: Gateway address

gateway-address

Specify the gateway address for the remote system's LAN interface.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the network server configuration.

*BLANK

Text is not specified.

character-value

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

Top

Keyboard layout (KBDTYPE)

Specifies the keyboard layout identifier to install on the Windows server. The valid keyboard layout identifiers are listed in the TXTSETUP.SIF file in the I386 directory of the Windows server installation media.

Note: This parameter is configured using the ServerGuide when INSTYPE is *BASIC. Any values entered for this parameter will be discarded.

*DEFAULT

The default keyboard layout for the version of Windows server being installed is used.

keyboard-type

Specify the keyboard layout identifier to be used by the Windows server.

Top

Configuration file (CFGFILE)

Specifies the name of a source file containing configuration data, to be used in activating or further defining the server.

Single values

*NONE

No configuration file is specified.

Qualifier 1: Configuration file

configuration-file-name

Specify the name of a source file containing the configuration data member(s) for the server. At the time the server is activated, all members in the file will be processed.

The name of the configuration file can be qualified by one of the following library values:

Qualifier 2: Library

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

*CURLIB

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

name Specify the name of the library to be used.

Top

Cluster name (CLU)

Specifies the name of the cluster.

*NONE

Do not form or join a Windows Cluster.

name Specify the name of the cluster. Administrators will use this name for connections to the cluster. The cluster name must be different from the domain name, from all computer names on the domain, and from other cluster names on the domain.

Top

Cluster configuration (CLUCFG)

Specifies the parameters required to configure a new Windows Cluster.

Note: This parameter is only required when forming a new Windows cluster using the **Cluster name (CLU)** parameter.

Single values

*CLU Use the values already defined by a previous installation of the Windows Cluster service to join the existing cluster name.

Element 1: Cluster domain name

Specifies the domain to which the cluster belongs. If the cluster already exists, the cluster will be joined, otherwise, the cluster will be formed. If forming a cluster, the **Cluster configuration (CLUCFG)** parameter must be specified.

character-string

Specify the domain name to which the cluster belongs when forming a new cluster.

Element 2: Quorum resource size

Specifies the size in megabytes of the storage space used as the Windows quorum resource.

*CALC

Specifies that the size should be calculated to be the default value based on the **Windows server version (WNTVER)** parameter.

integer

Specifies the Windows quorum resource size in megabytes. The size must be at least 550 megabytes but no larger than 1024000 megabytes.

Element 3: Quorum resource ASP

Specifies the auxiliary storage pool for the storage space used as the Windows quorum resource.

Note: You cannot specify both a Quorum resource ASP and a Quorum ASP device value.

1 The storage space is created in auxiliary storage pool 1, the system auxiliary storage pool.

integer

Specify a value ranging from 2 through 255 for the ASP identifier. Valid values depend on how many ASPs are defined on the system.

Element 4: Quorum ASP device

Specifies the independent auxiliary storage pool device name for the storage space used as the Windows quorum resource.

Note: You cannot specify both a Quorum resource ASP and a Quorum ASP device value.

name Specify the name of the independent auxiliary storage pool device. Valid values depend on which IASPs are defined on the system.

Element 5: Connection port

Specifies the connection port used for the Cluster service communication.

*VRTETHn

The network server virtual Ethernet port 'n' is configured, where 'n' has a value of 0 through 9.

Element 6: Cluster internet address

Specifies the internet address for the cluster.

internet-address

Specify the Cluster internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 7: Cluster subnet mask

subnet-mask

Specifies the subnet mask for the Cluster internet address.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Top

Virtual PTP Ethernet port (VRTPTPPORT)

Specifies the TCP/IP configuration for the virtual point-to-point Ethernet port.

Notes:

1. The VRTPTPPORT parameter must be used for Integrated xSeries servers running on the host system.
2. The subnet mask that is used for both sides of the virtual point-to-point Ethernet port is 255.255.255.0 by default. Therefore, the internet addresses that are chosen for both sides of the virtual point-to-point Ethernet port must have the same values for the first three parts of the internet addresses.

Element 1: Internet address

Specifies the internet address for the host side of the virtual point-to-point Ethernet connection. Specify one of the following values:

*GEN Specify *GEN to let the INSWNTSVR command configure a virtual point-to-point Ethernet port with a generated internet address.

internet-address

Specify the host internet address for the virtual point-to-point Ethernet port.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Element 2: Windows internet address

Specifies the internet address for the Windows side of the virtual point-to-point Ethernet connection. Specify one of the following values:

***GEN** Specify *GEN to let the INSWNTSVR command configure a virtual point-to-point Ethernet port with a generated internet address.

internet-address

Specify the Windows server internet address for the virtual point-to-point Ethernet port.

The value is entered in the decimal form nnn.nnn.nnn.nnn , where nnn is a decimal number ranging from 0 through 255.

Note: The internet address selected must be unique across all NWSD objects and the system's TCP/IP configuration.

Top

Examples

Example 1: Installing a Windows Terminal Server

```
INSWNTSVR  NWSD(W2KSERV)  INSTYPE(*FULL)
            RSRNAME(LIN09)  DMNROLE(*SERVER)
            WNTVER(*WIN2000) WNTSRCDIR(*DFT) OPTION(*INSTALL)
            TCPPTCFG((1 '206.5.8.48' '255.255.255.128'
                       '206.5.8.1'))
            SVRSTGSI(*CALC 2500) SVRSTGASP(1 1)
            CVTNTFS(*YES) TOWRKGPR(XYZGROUP)
            FULNAM('John Smith') ORG('XYZ Corporation')
            WNTLICKY('VVVVV-WWWW-XXXX-YYYY-ZZZZ')
            LICMODE(*PERSEAT *NONE *TSENABLE)
            TEXT('Windows 2000 Terminal Server')
```

This command installs a Windows server named W2KSERV. W2KSERV is the network server description associated with the Windows 2000 Server operating system that will be installed on the Integrated xSeries Server resource LIN09.

A fully controlled install will be performed with a system drive size of 2500 MB that will automatically be converted to NTFS during the install. The Windows 2000 Server will join the XYZGROUP workgroup. A per seat license mode will be configured and Terminal Services will be installed on the server.

The TCP/IP local host name is the same as the server description name. The TCP/IP local domain name is the same as the OS/400 system and the same name servers will be used. The TCP/IP address 206.5.8.48 will automatically be configured for the first LAN adapter detected on the Integrated xSeries Server.

Example 2: Installing a Windows Cluster Node

```
INSWNTSVR  NWSD(W2KNODE1) INSTYPE(*FULL)
            RSRNAME(LIN03)  DMNROLE(*SERVER)
            WNTVER(*WIN2000) WNTSRCDIR(*DFT) OPTION(*INSTALL)
            TCPPTCFG((1 '206.5.8.60' '255.255.255.128'
                       '206.5.8.1'))
            VRTETHPR((*VRTETH5 '192.168.9.3')
```

```

        '255.255.255.0'))
TCPDMNNAME(xyzdomain.xyzcorp.com)
TCPNAMSVR('206.5.69.165' '206.5.8.8'
          '206.10.244.100')
SVRSTGSIIZE(*CALC 4000) SVRSTGASP(1 1)
CVTNTFS(*YES) TODMN(XYZDOMAIN)
FULNAM('John Smith') ORG('XYZ Corporation')
WNTLICKKEY('VVVVV-WWWW-XXXX-YYYY-ZZZZ')
LICMODE(*PERSEAT *NONE *NONE)
TEXT('Windows 2000 Cluster Node 1')
CLU(CLU1XYZ)
CLUCFG(XYZDOMAIN 600 1 *N *VRTETH5 '206.5.8.65'
       '255.255.255.128')

```

This command installs a Windows server named W2KNODE1. W2KNODE1 is the network server description associated with the Windows 2000 Server operating system that will be installed on the Integrated xSeries Server resource LIN03.

A fully controlled install will be performed with a system drive size of 4000 MB that will automatically be converted to NTFS during the install. The Windows 2000 Server will join the XYZDOMAIN domain. A per seat license mode will be configured on the server. The TCP/IP address 206.5.8.60 will automatically be configured for the first LAN adapter detected on the Integrated xSeries Server. A virtual ethernet LAN will be created on virtual ethernet 5 configured with TCP/IP address 192.168.9.3.

The TCP/IP local host name is the same as the server description name. The TCP/IP local domain name is xyzdomain.xyzcorp.com. The TCP/IP name servers will use 206.5.69.165 206.5.8.8 and 206.10.244.100.

A new Microsoft Cluster will be enabled by creating a quorum resource drive named XYZDOMAIN that is 600 MB is size. Virtual ethernet 5 will be used for the private communication between clustered nodes.

Example 3: Installing Windows Server 2003 on an iSCSI attached server

```

INSWNTSVR  NWS(WS03LAN) INSTYPE(*FULL)
           RSRNAME(*ISCSI) DMNROLE(*SERVER)
           WNTVER(*WIN2003) WNTSRCDIR(*DFT) OPTION(*INSTALL)
           TCPPTCFG((1 '206.5.8.68' '255.255.255.128'
                     '206.5.8.1'))
           SVRSTGSIIZE(*CALC 2500) SVRSTGASP(1 1)
           CVTNTFS(*YES) TOWRKG(RP(XYZGROUP)
           FULNAM('John Smith') ORG('XYZ Corporation')
           WNTLICKKEY('VVVVV-WWWW-XXXX-YYYY-ZZZZ')
           STGPTH(NWSHRG1)
           VRTETHPTH((*VRTETHPTP NWSHRG1))
           DFTSECRULE(*GEN)
           INZSP(*AUTO)
           ENBUNICAST(*NO)
           EID(1234567 418477U)
           SPAUT(spadmin spuid)
           RMTSYSID(*EID)
           DELIVERY(*DYNAMIC)
           CHAPAUT(mychapid mychapsecret)
           RMTIFC((020134304760 '206.5.8.92' '255.255.255.128'
                   '6.5.8.1')
                 (020134604750 '206.5.8.96' '255.255.255.128'
                   '6.5.8.1'))
           TEXT('Windows Server 2003 iSCSI Server')

```

This command installs a Windows server named WS03LAN. WS03LAN is an iSCSI attached server using Windows Server 2003. Network server host adapter (NWSH) device NWSHRG1 is configured for the storage and virtual Ethernet path.

A default security rule using the *AUTO IP Security mode and a generated pre-shared key. The service processor is automatically configured and is located using the enclosures serial number 1234567 and type/model 418477U.

The remote system is dynamically configured and secured using a generated CHAP name and secret. Remote SCSI and LAN interfaces for the remote systems iSCSI adapter are configured.

[Top](#)

Error messages

***ESCAPE Messages**

NTA1007

Network server &1 must be varied off.

NTA100E

Vary on or off of the Windows server not successful.

NTA1013

Network server installation not successful.

NTA1024

Storage space assigned to server &1 missing, damaged or not valid.

NTA1030

Internal error occurred.

[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 NWS Configuration (INZNWSCFG)

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

Parameters
Examples
Error messages

The Initialize NWS Configuration (INZNWSCFG) command initializes or resets a service processor (*SRVPRC) network server configuration when various service processor parameters are changed or need to be enabled.

Restrictions:

- This command is shipped with public exclude (*EXCLUDE) authority. When this command is shipped, authority is issued only to the security officer. The security officer can grant the use of this command to other users.
- You must have input/output system configuration (*IOSYSCFG) and security administrator (*SECADM) special authorities to use this command.

Top

Parameters

Keyword	Description	Choices	Notes
NWSCFG	Network server configuration	<i>Communications name</i>	Required, Positional 1
OPTION	Processing option	*INIT, *CHGSPAUT, *REGEN, *SYNC	Required, Positional 2
SPAUT	SP authentication	Single values: *DFT Other values: <i>Element list</i>	Required, Positional 3
	Element 1: User name	<i>Character value</i>	
	Element 2: User password	<i>Character value</i>	

Top

Network server configuration (NWSCFG)

Specifies the name of the network server configuration.

This *NWSCFG object must have been created with *SRVPRC specified for the **Configuration type (TYPE)** parameter on the Create NWS Configuration (CRTNWSCFG) command.

This is a required parameter.

name Specify the service processor network server configuration to be processed.

Top

Processing option (OPTION)

Specifies option to perform with the service processor network server configuration.

This is a required parameter.

***INIT** Initialize a new service processor.

Note: This option is used to configure a service processor that has never been configured before such as when a new system is put into service or has had a service processor field replacement performed.

Note: A new user name and password must be specified on the **SP authentication (SPAUT)** parameter unless INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command

***CHGSPAUT**

Change the service processor user name and password to the new values specified on the **SP authentication (SPAUT)** parameter.

***REGEN**

Request that the service processor regenerate a self signed certificate. This option is used if the service processor's certificate has expired, or if a new certificate and password are desired at any time before the certificate expires.

Note: This option is only valid when INZSP(*AUTO) is specified in the corresponding **CRTNWSCFG** command.

Note: A new password must be specified on the **SP authentication (SPAUT)** parameter unless INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command

***SYNC**

Synchronize the self-signed certificate from the service processor. This option is used if multiple service processor network server configurations are used for the same system or the service processor network server configuration has been restored from backup and the service processor's certificate must be synchronized. The current user name and password for the service processor must be specified on the **SP authentication (SPAUT)** parameter to perform this option.

Top

SP authentication (SPAUT)

Specifies the service processor user name and password. This is used to authenticate and secure the service processor.

Note: The *DFT value is only valid when INZSP(*NONE) was specified in the corresponding **CRTNWSCFG** command.

This is a required parameter.

Single values

***DFT** The default service processors userid and password are used.

Element 1: User name

character-string

Specify a name that represents the host configuration that owns the service processor. It is suggested that the remote system network server configuration name be used. If multiple remote system network server configurations use the same service processor at different times, each configuration must contain the same user name and password.

Element 2: User password

character-string

Specify the service processor password. Password must be at least 5 characters in length and contain at least one alphabetic character and one numeric or symbolic character.

Top

Examples

Example 1: Regenerate Service Processor Certificate

```
INZNWSCFG NWSCFG(MYCONFIG)
          OPTION(*REGEN)
          SPAUT(username password)
```

This command regenerates the remote system's service processor certificate then changes the user name and password.

Example 2: Change Service Processor User Name and Password

```
INZNWSCFG NWSCFG(MYCONFIG)
          OPTION(*CHGSPAUT)
          SPAUT(username password)
```

This command changes the user name and password used to secure the service processor.

Top

Error messages

*ESCAPE Messages

CPF0910

Password not valid for system.

CPF96CB

Network server configuration &1 not found.

CPF96CD

Network server configuration type &2 is not valid.

CPF96CE

Password matches the previous value.

CPF96CF

Errors processing network server configuration &1.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9899

Error occurred during processing of command.

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.

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.

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.

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 iSeries Access (INZPCS)

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

Parameters
Examples
Error messages

The Initialize iSeries Access (INZPCS) command allows you to establish an operating environment for iSeries Access applications by creating various control documents on the iSeries 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 iSeries Access (INZPCS command) failed.

IWS16DD

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

IWS16E1

INZPCS command successfully completed.

IWS16E2

Error retrieving data area &1 in library &2.

IWS16E3

Error creating data area QINZPCSDA in library QUSRSYS.

IWS16E4

Error updating data area QINZPCSDA in library QUSRSYS.

IWS16EE

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

Top

Parameters

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

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

Top

Keyboard type (KBDTYPE)

Specifies the keyboard type used.

The possible values are:

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

keyboard-type

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

Character Translation Table

Identifier

Language

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

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

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

Top

ASCII code page number (ASCII)

Specifies the ASCII code page number to use.

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

The possible values are:

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

code-page-number

Specify the ASCII code page number to use.

Top

EBCDIC code page number (EBCDIC)

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

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

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

code-page-number

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

Top

Language feature code (LANGUAGE)

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

The possible values are:

***DFT** The primary language of iSeries 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

[Top](#)

Error messages

*ESCAPE Messages

IWS16D0

Initialize iSeries Access (INZPCS command) failed.

IWS16DD

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

IWS16E1

INZPCS command successfully completed.

IWS16E2

Error retrieving data area &1 in library &2.

IWS16E3

Error creating data area QINZPCSDA in library QUSRSYS.

IWS16E4

Error updating data area QINZPCSDA in library QUSRSYS.

IWS16EE

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

[Top](#)

Initialize Physical File Mbr (INZPFM)

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

Parameters
Examples
Error messages

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

Restrictions:

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

Top

Parameters

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

Top

Physical file (FILE)

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

This is a required parameter.

Qualifier 1: Physical file

name Specify the name of physical file.

Qualifier 2: Library

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

**CURLIB*

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

name Specify the name of the library to be searched.

Member (MBR)

Specifies the file member to be initialized.

*FIRST

The first member of the specified file is used.

*LAST

The last member of the specified physical file is initialized.

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

Top

Initialize records as (RECORDS)

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

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

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

Top

Total number of records (TOTRCDS)

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

*NXTINCR

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

1-4294967288

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

Top

Examples

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

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

Top

Error messages

*ESCAPE Messages

CPF3130

Member &2 already in use.

CPF3131

Cannot initialize member &2 with default records.

CPF3132

TOTRCDS parameter value either missing or too small.

CPF3133

File &1 in library &3 contains no members.

CPF3134

Referential constraint error processing member &2.

CPF3136

File &1 in &3 not allowed on command.

CPF3137

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

CPF3138

Check constraint error processing member &2.

CPF3140

Initialize or copy of member &2 canceled.

CPF3141

Member &2 not found.

CPF3142

File &1 in library &3 not found.

CPF3143

Increments not allowed for member &2.

CPF3144

Member &2 not cleared or initialized.

CPF3148

New records need too much space for member &2.

CPF3156

File &1 in library &3 in use.

CPF3157

Triggers prevent requested operation.

CPF3159

Member &2 saved with STG(*FREE).

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

Top

Initialize System (INZSYS)

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

Parameters
Examples
Error messages

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

More information is available in the Install, upgrade, or delete i5/OS and related software book, SC41-5120.

Top

Parameters

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

Top

Message queue (MSGQ)

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

***SYSOPR**

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

message-queue-name

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

The possible library values are:

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

***CURLIB**

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

library-name

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

Top

Examples

INZSYS

This command initializes the conversions done during installation procedures.

[Top](#)

Error messages

*ESCAPE Messages

CPF372A

INZSYS or GO LICPGM currently running in another job.

CPF90E2

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

CPF90E3

Error occurred for file &1 in library &2.

CPF90E4

System function in use. Reason code &1.

CPF90E8

Error occurred for file &1 in library &2.

CPF90E9

Data exists for more than one previous release.

[Top](#)

Initialize Tape (INZTAP)

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

Parameters
Examples
Error messages

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

Top

Parameters

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

Top

Device (DEV)

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

This is a required parameter.

Top

New volume identifier (NEWVOL)

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

*NONE

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

*CTGID

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

character-value

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

Top

New owner identifier (NEWOWNID)

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

*BLANK

Text is not specified.

character-value

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

Top

Volume identifier (VOL)

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

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

*MOUNTED

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

character-value

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

Top

Check for active files (CHECK)

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

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

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

***FIRST**

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

Top

Tape density (DENSITY)

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

***DEVTYPE**

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

Device

Highest capacity density or format

3480 *FMT3480

3490E *FMT3490E

3570-Bxx
*FMT3570

3570-Cxx
*FMT3570E

3580-001
*ULTRIUM1

3580-002
*ULTRIUM2

3580-003
*ULTRIUM3

3590-Bxx
*FMT3590

3590-Exx
*FMT3590E

3590-Hxx
*FMT3590H

3592-J1A
*FMT3592A1

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
63B0 *VRT256K
7207-122
 *QIC4DC
7208-002
 *FMT2GB
7208-012
 *FMT5GB
7208-222
 *FMT7GB
7208-342
 *FMT20GB
7208-345
 *FMT60GB
9348 6250

***CTGTYPE**

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

character-value

Specify the density or format to use.

- 1600** The data density on the tape volume is 1,600 bits per inch, which is used for 1/2 inch reel tapes.
- 3200** The data density on the tape volume is 3,200 bits per inch, which is used for 1/2 inch reel tapes.
- 6250** The data density on the tape volume is 6,250 bits per inch, which is used for 1/2 inch reel tapes.
- *FMT3480**
The format of this tape is FMT3480. The data density on this tape volume is formatted to support a 3480 device. This density is used for 1/2 inch cartridge tapes.
- *FMT3490E**
The format of this tape is FMT3490E. The data density on this tape volume is formatted to support a 3490E device. This density is used for 1/2 inch cartridge tapes.
- *FMT3570**
The format of this tape is FMT3570. The data format is written on the tape volume with a 3570 device.
- *FMT3570E**
The format of this tape is FMT3570E. The data format is written on the tape volume with a 3570E device.
- *FMT3590**
The format of this tape is FMT3590. The data format is written on the tape volume with a 3590 device. This density is used for 1/2 inch cartridge tapes.
- *FMT3590E**
The format of this tape is FMT3590E. The data format is written on the tape volume with a 3590E device. This density is used for 1/2 inch cartridge tapes.
- *FMT3590H**
The format of this tape is FMT3590H. The data format is written on the tape volume with a 3590H device. This density is used for 1/2 inch cartridge tapes.
- *FMT3592A1**
The format of this tape is FMT3592A1. The data format is written on the tape volume with a 3592 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.

***ULTRIUM3**

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

***VRT32K**

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

***VRT64K**

The format of the volume is VRT64K. It is used to write data to a virtual volume using a

maximum data block size of 64KB. Volumes written using this format can only be duplicated to tape devices that support a maximum block size of 64KB or greater.

***VRT240K**

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

***VRT256K**

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

***VXA1**

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

***VXA2**

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

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.

Top

Code (CODE)

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

***EBCDIC**

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

***ASCII**

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

Top

End of tape option (ENDOPT)

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

***REWIND**

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

***UNLOAD**

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

Top

Clear (CLEAR)

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

- *NO** Existing data is not deleted. Even though the existing data is not deleted, the data on the volume is not accessible after the volume has been initialized for use.
- *YES** After the beginning of the tape has been initialized for use, the rest of the data on the tape is deleted. The *YES value is needed only if there are security concerns with the old data. If *YES is selected, the initialize operation can take a long time.

Top

Examples

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

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

Top

Error messages

*ESCAPE Messages

CPF67A0

Volume ID does not match cartridge ID

CPF6702

Error processing volume on device &1.

CPF6708

Command ended due to error.

CPF6715

Error at beginning of tape on device &1.

CPF6718

Cannot allocate device &1.

CPF6720

Incorrect volume &2 found on device &1.

CPF6721

Device &1 not a tape device.

CPF6722

End of tape found on device &1.

CPF6745

Device &1 not a media library device.

- CPF6750**
NEWVOL(*NONE) not valid for device &1.
- CPF6751**
Load failure occurred on device &4.
- CPF6754**
Active file &4 found on volume &2.
- CPF6760**
Device &1 not ready.
- CPF6762**
Wrong type of cartridge in device &1.
- CPF6763**
Wrong type of cartridge in device &1.
- CPF6768**
Volume on device &1 is write protected.
- CPF676B**
Volume on device &4 is write protected.
- CPF6772**
Volume on device &1 cannot be processed.
- CPF6774**
New volume &2 is a nonstandard labeled tape. Volume not prepared.
- CPF9814**
Device &1 not found.
- CPF9825**
Not authorized to device &1.

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

Leaves the innermost loop surrounding this LEAVE command.

simple-name

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

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 device. The status of the image catalog will be changed based on the value specified for the **Option (OPTION)** parameter as follows:

*LOAD

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

*UNLOAD

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

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

Restrictions:

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

Top

Parameters

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

Top

Image catalog (IMGCLG)

Specifies the image catalog to be loaded or unloaded.

This is a required parameter.

name Specify the name of the image catalog.

Top

Virtual device (DEV)

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

This is a required parameter.

name Specify the name of the virtual device.

Top

Option (OPTION)

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

*LOAD

The image catalog will be loaded into the virtual device.

*UNLOAD

The image catalog will be unloaded from the virtual device.

Top

Write protect (WRTPTC)

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

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

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

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

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

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

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

***NONE**

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

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

1. Optical dependent image catalogs

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

2. Optical reference image catalogs

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

3. Tape dependent image catalogs

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

4. Tape reference image catalogs

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

Top

Examples

Example 1: Loading an Image Catalog

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

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

Example 2: Loading an Image Catalog as read-only

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

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

Example 3: Unloading an Image Catalog

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

This command unloads image catalog **MYCLG** from device **OPTVRT01**.

Example 4: Loading a tape type Image Catalog

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

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

Top

Error messages

*ESCAPE Messages

CPFBC10

Image catalog &1 not loaded to device &2.

CPFBC11

Image catalog &1 not unloaded from device &2.

CPFBC45

Image catalog &1 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9820

Not authorized to use library &1.

[Top](#)

Load/Unload/Mount IMGCLG Entry (LODIMGCLGE)

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

Parameters
Examples
Error messages

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

*LOAD

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

*UNLOAD

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

*MOUNT

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

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

Restrictions:

- This command is shipped with public *EXCLUDE authority.
- The following authorities are required to load an image catalog entry:
 1. Execute (*EXECUTE) authority to library QUSRSYS.
 2. *USE authority to the image catalog being loaded or unloaded.
 3. *USE authority to the virtual device description.
 4. Execute (*X) authority to each directory in the image catalog path name.
- The following parameters are only supported for tape type image catalogs:
 1. Image catalog volume (VOL)

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, *NEXT, *VOL	Optional, Positional 2
OPTION	Option	<u>*MOUNT</u> , *LOAD, *UNLOAD	Optional, Positional 3
VOL	Image catalog volume	<i>Character value</i>	Optional

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.

*NEXT

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

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

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

Top

Option (OPTION)

Specifies the new status of the image catalog entry.

*MOUNT

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

*LOAD

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

*UNLOAD

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

Top

Image catalog volume (VOL)

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

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

Top

Examples

Example 1: Mounting an Image Catalog Entry

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

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

Example 2: Loading an Image Catalog Entry

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

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

Example 3: Unloading an Image Catalog Entry

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

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

Example 4: Mounting a Tape Image Catalog Entry by Volume

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

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

Top

Error messages

*ESCAPE Messages

CPFBC46

Catalog entry at index &1 not loaded.

CPFBC47

Catalog entry at index &1 not unloaded.

CPFBC48

Catalog entry at index &1 not mounted.

CPFBC0D

Catalog entry at index &1 not loaded.

CPFBC0E

Catalog entry at index &1 not unloaded.

CPFBC0F

Catalog entry at index &1 not mounted.

CPFBC45

Image catalog &1 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9820

Not authorized to use library &1.

Top

Load Optical Firmware (LODOPTFMW)

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

Parameters
Examples
Error messages

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

Restrictions:

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

[Top](#)

Parameters

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

[Top](#)

From stream file (FROMSTMF)

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

[Top](#)

Device type (TYPE)

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

[Top](#)

Device model (MODEL)

Specifies the model the firmware is for.

[Top](#)

Examples

Example 1: Specify a Media Library

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

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

Example 2: Specify an Optical Drive

```
LODOPTFMW FROMSTMF('/MyDir/MyFile') TYPE(399F)
           MODEL(001)
```

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

[Top](#)

Error messages

*ESCAPE Messages

OPT1114

Error accessing stream file.

OPT1121

Length beyond end of buffer.

OPT1681

An active optical device was not found.

OPT1815

Internal program error occurred.

OPT1821

Error occurred on optical device &1.

[Top](#)

Load Program Temporary Fix (LODPTF)

Where allowed to run: All environments (*ALL)
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. Specific PTFs can be selected or omitted when loading PTFs for the specified product. PTFs that are currently applied are not 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	*SRVATT, *YES, *NO, *ONLY	Optional
CPYSAVF	Copy PTF save file	*SRVATT, *YES, *NO	Optional

Top

Product (LICPGM)

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

This is a required parameter.

Top

Device (DEV)

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

*SERVICE

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

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

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

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

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 (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 or optical device name is specified on the **Device (DEV)** parameter.

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

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

***ONLY**

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

***SRVATT**

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

Top

Copy PTF save file (CPYSAVF)

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

***SRVATT**

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

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

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

Top

Examples

Example 1: Omitting PTFs

```
LODPTF LICPGM(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.

CPF35CC

Library required for PTF operation already exists.

CPF35CF

PTF &1-&2 not applied.

CPF35C1

LODPTF ended. No more storage available.

CPF35C9

PTF &1-&2 &3 not valid.

CPF35EB

Multiple releases of product &1 installed.

CPF35E3

Interface error detected.

CPF35FA

PTF &1-&2 not applied.

CPF35F4
Error occurred during cover letter processing.

CPF35F6
MPTFI for library &1 deleted and created.

CPF354A
Cannot specify *SELECT for the path identifier.

CPF354C
Cannot process PTF files on optical volume.

CPF354D
Device &1 not allowed.

CPF354E
No file selected.

CPF354F
Required PTF file cannot be processed.

CPF355B
Multiple releases for product &1 found on media.

CPF355C
No PTFs found in path identifier &1.

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

CPF3564
PTF &1-&2 damaged.

CPF358A
Release not valid.

CPF3586
List of PTFs not correct.

CPF3587
PTFs not loaded.

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

CPF3598
PTF function already in process.

CPF3606
Product &1 &2 not installed.

CPF361D
Apply order of PTFs cannot be determined.

CPF3612
Library &1 not found.

CPF3616
No PTFs loaded.

CPF3619
PTFs for release &1 found on device.

CPF3657
PTFs not loaded because error occurred.

CPF3693

Service function ended because error occurred.

CPF3924

PTF not loaded.

CPF3931

Required programs not found. PTF incomplete.

CPF3945

Records of PTF activity for licensed program are deleted.

CPF3992

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

CPF6602

PTF &1-&2 &3 not found.

CPF8191

Product definition &4 in &9 damaged.

CPF8193

Product load object &4 in &9 damaged.

Top

Load Q/A Database (LODQSTDB)

Where allowed to run:

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

Threadsafe: No

Parameters
Examples
Error messages

The Load Question and Answer Database (LODQSTDB) command allows you to load a Question and Answer (Q & A) database from an alternative medium (such as tape) to the system. More information is available in the Basic System 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 iSeries or System/36 naming conventions for devices:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Parameters

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

Device (DEV)

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

This is a required parameter.

The possible values are:

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

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

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

tape-device

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

diskette-device

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

Optical-device

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

Sequence number (SEQNBR)

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

The possible values are:

*FIRST

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

*SEARCH

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

sequence-number

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

Volume identifier (VOL)

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

The possible values are:

*MOUNTED

The volume currently mounted on the tape device is used.

volume-identifier

Specify the volume identifier mounted on the tape device.

Top

Directory (DIR)

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

The possible values are:

/ The root directory (/) is used.

directory-name

Specify the directory to search for a file named QTEMP.

Top

Examples

Example 1: Restoring a Program from Tape

```
LODRUN DEV(TAP01)
```

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

Example 2: Restoring the Program QINSTAPP from Tape

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

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

Example 3: Restoring the Program QINSTAPP from CD-ROM

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

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

Top

Error messages

None

[Top](#)

Create Directory (MD)

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

Parameters
Examples
Error messages

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

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

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

- CRTDIR
- MKDIR

For more information about integrated file system commands, see the Integrated file system 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 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 Create object auditing (QCRTOBJAUD) system value.

***NONE**

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

***USRPRF**

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

***CHANGE**

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

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

Top

Scanning option for objects (CRTOBJSCAN)

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

The integrated file system scan-related exit points are:

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

For details on these exit points, see the 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 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 Create object auditing (QCRTOBJAUD) system value.

***NONE**

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

***USRPRF**

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

***CHANGE**

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

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

Top

Scanning option for objects (CRTOBJSCAN)

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

The integrated file system scan-related exit points are:

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

For details on these exit points, see the 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. The label specified on a program-level MONMSG command cannot be a label associated with a subroutine. If a GOTO command is performed for a program-level MONMSG, the subroutine stack will be reset by the next Call Subroutine (CALLSUBR) command.

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

Restrictions:

- This command is valid only in 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, CALLSUBR, SUBR, RTNSUBR, ENDSUBR, GOTO, IF, or RETURN. Note that if another program sends a message that is monitored by this command, a return cannot be made to that program.

Top

Parameters

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

Top

Message identifier (MSGID)

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

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

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

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

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

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

This is a required parameter.

Comparison data (CMPDTA)

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

*NONE

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

comparison-data

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

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

Command to execute (EXEC)

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

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

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

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

Examples

Example 1: Monitoring Messages Sent by Any Command

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

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

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

Example 2: Monitoring Messages Sent by a Single Command

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

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

[Top](#)

Error messages

None

[Top](#)

Add Mounted FS (MOUNT)

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

Parameters
Examples
Error messages

The Add Mounted File System (MOUNT) command makes the objects in a file system accessible to the integrated file system name space. The file system to be made accessible can be either a user-defined file system (*UDFS) on the local system, 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 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.
4. The user must have execute (*X) authority to each directory in the path.

Top

Parameters

Keyword	Description	Choices	Notes
TYPE	Type of file system	*NFS, *UDFS, *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 *wsize* 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/iseries/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

CPFA09C

Not authorized to object. Object is &1.

CPFA0A2

Information passed to this operation was not valid.

CPFA0A9

Object not found. Object is &1.

CPFA1B8

*IOSYSCFG authority required to use &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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iseriess/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.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0B0

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

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0B2

No objects satisfy request.

CPFA0B8

&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:

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

Top

Parameters

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

Top

From document (FROMDOC)

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

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

This is a required parameter.

*SYSOBJNAM

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

name Specify the name of the document to be moved.

From folder (FROMFLR)

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

*NONE

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

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

To folder (TOFLR)

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

*NONE

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

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

Rename (RENAME)

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

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

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

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

*SAME

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

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

System object name (SYSOBJNAM)

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

name Specify the system object name.

Examples

Example 1: Adding a Folderless Document

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

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

Example 2: Moving a Document and Keeping its Name

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

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

Example 3: Moving and Renaming a Document

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

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

Example 4: Moving a Document and Making It Folderless

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

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

Error messages

*ESCAPE Messages

CPF8A13

Document &2 in folder &1 not moved.

Move Object (MOVE)

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

Parameters
Examples
Error messages

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

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

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

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

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

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

- MOV

For more information about integrated file system commands, see the Integrated file system 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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/series/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 the CL concepts and reference topic in the iSeries Information Center at <http://www.ibm.com/eserver/iserics/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.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A7

Path name too long.

CPFA0B0

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

CPFA0B1

Requested operation not allowed. Access problem.

CPFA0B2

No objects satisfy request.

CPFA0B8

&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.

When the object type of the object to be moved is an object type which can be journaled, the existence and content of a data area named QDFTJRN in the to-library will determine whether or not journaling is started for the object. With the exception noted below in the considerations for moving an object from QTEMP to a primary or secondary ASP, if the object was journaled in the from-library, it will continue to be journaled in the to-library.

Note: For additional information regarding journaling, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Restrictions:

1. For an object other than a *FILE object, you must have:
 - object management (*OBJMGT) authority for the object which is to be moved,
 - delete (*DELETE) and read (*READ) authorities for the library from which the object is to be moved, and
 - add (*ADD) and read (*READ) authorities for the library to which the object is to be moved.
2. For a *FILE object, you must have:
 - object management (*OBJMGT) and object operational (*OBJOPR) authority for the object which is to be moved,
 - delete (*DELETE), add (*ADD), and execute (*EXECUTE) authorities for the library from which the object is to be moved, and
 - add (*ADD) and read (*READ) authorities for the library to which the object is to be moved.
3. The following object types cannot be moved: Libraries, user profiles, edit descriptions, line descriptions, controller descriptions, device descriptions, journals, and journal receivers.
4. The following objects cannot be moved: the system operator message queue QSYSOPR, all workstation user message queues, and the system log QHST.
5. The library to which the object is to be moved must not already contain an object of the same name and type as the object to be moved.
6. The library to which the object is to be moved cannot be QTEMP.
7. The user space (*USRSPC), user index (*USRIDX), and user queue (*USRQ) user domain objects can only be moved into libraries that are permitted in the system value QALWUSRDMN (allow user domain objects in library). However, if the user object was created as a system domain object, it is not restricted.
8. As a general rule, objects cannot be moved to the to-library if the object and the to-library are in different auxiliary storage pools (ASPs). An error message is sent when the object cannot be moved. There are some specific exceptions to the general rule:

- You can move save files that are in a basic user ASP to libraries that are in the system ASP (ASP 1) if the save file's library is also in the system ASP.
- You can move objects in a secondary ASP to the primary ASP in the same ASP group if the to-library is QRPLxxxx (where 'xxxx' is the number of the primary ASP of the ASP group.)
- You can move an object from QTEMP to a primary or secondary ASP with the following considerations:
 - The 'move' is accomplished through a save and restore operation.
 - The size of the object must be less than 1 terabyte. (The Move Library to ASP (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.
 - The private authorities for the objects will be preserved.
 - After the object has been moved, the following attributes will differ from the original object:
 - The date last used will be set to blank.
 - The change date and time will be set to the current date and time.
 - The days used count will be set to zero.
 - The date use count reset will be set to blank.
 - The restore date and time will be set to the current date and time.
 - When the object type of the object to be moved is an object type which can be journaled, the existence and content of a data area named QDFTJRN in the to-library will determine whether or not journaling is started for the object. If the QDFTJRN data area does not exist or does not indicate that journaling should be started for a restore operation for the object type, the object will not be journaled even if the original object was journaled.

Note: For additional information regarding journaling, see the Journal Management information in the iSeries Information Center at <http://www.ibm.com/eserver/series/infocenter>.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OBJTYPE	Object type	*ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *JOBQ, *JOBQ, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBS, *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

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/series/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.

Top

To ASP device (TOASPDEV)

Specifies the auxiliary storage pool (ASP) device where storage is allocated for the to-library specified for the **To library (TOLIB)** parameter. If the to-library is in an ASP that is not part of the library name space associated with the thread, this parameter must be specified to ensure the correct object is moved. If this parameter is used when *CURLIB is specified for the TOLIB parameter, either TOASPDEV(*) must be specified or TOASPDEV(*ASPDEV) must be specified and the **From ASP device (ASPDEV)** parameter must be *.

*ASPDEV

The ASP device specified for the ASPDEV parameter will be searched to find the library.

* The ASPs that are currently part of the thread's library name space will be searched to find the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

*CURASPGRP

If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. If no ASP group is associated with the thread an error will be issued.

*SYSBAS

The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to find the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

name Specify the name of the primary or secondary ASP device to be searched to find the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched.

Note: To specify a specific auxiliary storage pool (ASP) device name, you must have use (*USE) authority for each ASP device in the ASP group.

Top

Examples

Example 1: Moving an Object from the General Purpose Library

```
MOV OBJ(QGPL/X) OBJTYPE(*PGM) TOLIB(MY)
```

The general purpose library (QGPL) is searched for the X program (*PGM) object. The X program object is moved to the MY library. After this command is run, the X program object is no longer in the QGPL library.

Example 2: Moving an Object from a Library in the Library List

```
MOV OBJ(*LIBL/Y) OBJTYPE(*FILE) TOLIB(Z)
-or-
MOV OBJ Y *FILE Z
```

The library list (*LIBL) is searched for the Y file object. If more than one file object with the same name exists in the libraries making up the library list, the first Y file object found in the library list is moved to the Z library. After this command is run, the Y file object is no longer in the library where it was found.

Example 3: Moving an Object from a Library in an Independent Auxiliary Storage Pool (ASP) to a Library in a different ASP.

```
MOV OBJ(INVENTORY/MONTHLY) OBJTYPE(*PGM)
TOLIB(WINVENTORY) ASPDEV(SALES) TOASPDEV(WSALES)
```

The INVENTORY library in the SALES independent auxiliary storage pool (ASP) is searched for the MONTHLY program object. The MONTHLY program object is moved to the WINVENTORY library in the WSALES ASP. After this command is run, the MONTHLY program object is no longer in the INVENTORY library in the SALES ASP. The SALES ASP and the WSALES ASP must have been activated (by varying on the ASP device) and have a status of 'Available'.

Top

Error messages

*ESCAPE Messages

CPFA030

Object already in use.

CPFB8ED

Device description &1 not correct for operation.

CPF0601

Not allowed to do operation to file &1 in &2.

CPF0602

File &1 already in library &2.

CPF0605

Device file &1 in &2 saved with storage freed.

CPF0610

File &1 in &2 not available.

CPF0678
Operation not performed for file name &1 in &2.

CPF1763
Cannot allocate one or more libraries.

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2112
Object &1 in &2 type *&3 already exists.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2135
Object &1 type *&3 already exists in library.

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF2160
Object type *&1 not eligible for requested function.

CPF216C
TOASPDEV value not allowed with TOLIB(*CURLIB).

CPF2173
Value for ASPDEV not valid with special value for library.

CPF218C
&1 not a primary or secondary ASP.

CPF2182
Not authorized to library &1.

CPF2183
Object &1 cannot be moved into library &3.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2193
Object &1 cannot be moved into library &4.

CPF22BC
Object &1 type &3 is not program defined.

CPF2451
Message queue &1 is allocated to another job.

CPF2512
Operation not allowed for message queue &1.

CPF32CF
Distributed file error, reason code &3.

CPF32C3
Distributed file error, level ID mismatch

CPF320B
Operation was not valid for database file &1.

CPF320C
File &1 not allowed in SQL collection &2.

CPF3201
File &1 in library &2 already exists.

CPF3202
File &1 in library &2 in use.

CPF3203
Cannot allocate object for file &1 in &2.

CPF322D
Operation not done for data base file &1.

CPF3220
Cannot do operation on file &1 in &2.

CPF3224
Not authorized to perform operation on file &1.

CPF323C
QRECOVERY library could not be allocated.

CPF323D
User does not have correct authority.

CPF323F
Move or rename of file &1 in library &2 not complete.

CPF3231
Cannot move file &1 from library &2.

CPF324B
Cannot allocate dictionary for file &1.

CPF324C
Concurrent authority holder operation prevents move, rename or restore.

CPF3245
Damage to file &1 member &6 prevents operation on file &3.

CPF325D
Field CCSID values not compatible.

CPF327C
File &1 cannot be moved into library &4.

CPF327E
Alternative name for file &1 not allowed.

CPF329D
Operation not successful for file &1 in library &2.

CPF3323
Job queue &1 in &2 already exists.

CPF3330
Necessary resource not available.

CPF3353
Output queue &1 in &2 already exists.

CPF3373
Job queue &1 in &2 not moved. Job queue in use.

CPF3374
Output queue &1 in &2 not moved. Output queue in use.

CPF3467
Output queue &1 deleted and then created again.

CPF3469
Operation not allowed for output queue.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF7010
Object &1 in &2 type *&3 already exists.

CPF7014
Object &1 cannot be moved to library &4.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

CPF9827
Object &1 cannot be created or moved into &2.

CPF9833
*CURASGRP 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

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) IBM 2006. Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1998, 2006. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2

IBM
Infoprint
InfoWindow
iSeries
LPDA
OfficeVision
i5/OS
Print Services Facility
RPG/400
SystemView
System/36
TCS
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE



Printed in USA