



System i

Programming

i5/OS commands

Starting with RMVTRCFTR (Remove Trace Filter)

Version 6 Release 1





System i

Programming

i5/OS commands

Starting with RMVTRCFTR (Remove Trace Filter)

Version 6 Release 1

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 805.

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

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

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

Contents

Remove Trace Filter (RMVTRCFTR)	1	Parameters	23
Parameters	1	List identifier (LSTID)	23
Filter (FTR)	1	New list identifier (NEWLSTID)	24
Examples	1	Examples	24
Error messages	2	Error messages	24
Remove Work Station Entry (RMVWSE)	3	Rename LAN Adapter (RNMLANADPI)	27
Parameters	3	Parameters	27
Subsystem description (SBSD)	3	Adapter (ADPTNAME)	27
Work station name (WRKSTN)	4	New name (NEWNAME)	27
Work station type (WRKSTNTYPE)	4	Examples	27
Examples	5	Error messages	28
Error messages	5		
 		Rename Member (RNMM)	29
Run RNDCC Command (RNDCC)	7	Parameters	29
Parameters	7	Data base file (FILE)	29
RNDCC command (RNDCCCMD)	8	Member (MBR)	30
Domain name server (DMNNAMSVR)	9	New member (NEWMBR)	30
Domain name server port (PORT)	9	Examples	30
RNDCC configuration file (RNDCCCFGF)	9	Error messages	30
Source address (SRCADR)	9		
Key file (KEYFILE)	10	Rename Nickname (RNMNCK)	31
Key name (KEYNAME)	10	Parameters	31
Show debug information (DEBUG)	10	Nickname (NCK)	31
Output file (TOSTMF)	10	New nickname (NEWNCK)	32
Examples	11	Examples	32
Error messages	12	Error messages	32
Rename Object (RNM)	13	Rename Object (RNMOBJ)	33
Parameters	13	Parameters	35
Object (OBJ)	14	Object (OBJ)	35
New object (NEWOBJ)	14	Object type (OBJTYPE)	36
Examples	14	New object (NEWOBJ)	36
Error messages	14	ASP device (ASPDEV)	36
		System (SYSTEM)	37
Rename Directory Entry (RNMDIRE)	17	Examples	37
Parameters	17	Error messages	37
Old user identifier (OLDUSRID)	17		
New user identifier (NEWUSRID)	18	Rename TCP/IP Host Table Entry	
Forward from user identifier (FWDFRM)	19	(RNMTCPHTE)	43
Change network user identifier (NETUSRID)	19	Parameters	43
Examples	19	Internet address (INTNETADR)	44
Error messages	20	New internet address (NEWINTNETA)	44
		Examples	44
Rename Document Library Object		Error messages	45
(RNMDLO)	21		
Parameters	21	Rollback (ROLLBACK)	47
Document library object (DLO)	21	Parameters	47
New document library object (NEWDLO)	21	Examples	47
In folder (FLR)	21	Error messages	47
Examples	22		
Error messages	22	Start RPC Binder Daemon (RPCBIND)	49
		Parameters	49
Rename Distribution List (RNMDSTL)	23	Retrieve RPC registration (RTVRPCREG)	49

Examples	50
Error messages	50
Convert RPC Source (RPCGEN)	51
Parameters	51
From file (FROMFILE).	51
Option (OPTION)	51
Protocol (PROTOCOL).	52
To file (TOFILE).	52
Examples	53
Error messages	53

Replace Document (RPLDOC).	55
Parameters	55
Document (TODOC)	56
In folder (TOFLR)	56
Document file (DOCFIL)	56
Document member (DOCMBR).	56
Document part (DOCPART)	57
User identifier (USRID)	57
Document identifier (DOCID)	57
Document type (DOCTYPE).	58
Document character identifier (DOCCHRID)	58
Command character identifier (CMDCHRID)	59
Examples	59
Error messages	60

Request Order Assistance (RQSORDAST)	61
Parameters	61
Examples	61
Error messages	61

Reroute Job (RRTJOB).	63
Parameters	63
Routing data (RTGDTA)	63
Request data or command (RQSDTA)	64
Examples	64
Error messages	64

Resume Breakpoint (RSMBKP)	65
Parameters	65
Examples	65
Error messages	65

Resume Controller Recovery (RSMCTLCY)	67
Parameters	67
Controller (CTL).	67
Examples	68
Error messages	68

Resume Device Recovery (RSMDEVRCY)	69
Parameters	69
Device (DEV).	69
Examples	70
Error messages	70

Resume Line Recovery (RSMLINRCY)	71
Parameters	71
Line (LINE)	71
Examples	71
Error messages	71

Resume NWI Recovery (RSMNWIRCY)	73
Parameters	73
Network interface description (NWI)	73
Examples	73
Error messages	73

Restore Object (RST)	75
Parameters	75
Device (DEV).	76
Objects (OBJ)	77
Name pattern (PATTERN)	78
Directory subtree (SUBTREE)	78
Output (OUTPUT)	79
Create parent directories (CRTPRNDIR)	79
Parent directory owner (PRNDIROWN)	79
Rebuild mounted file system (RBDMFS).	80
Volume identifier (VOL)	80
Label (LABEL)	81
Sequence number (SEQNBR)	81
End of media option (ENDOPT)	81
Optical file (OPTFILE)	82
Type of output information (INFTYPE)	82
System (SYSTEM)	82
Date when saved (SAVDATE)	82
Time when saved (SAVTIME)	83
Option (OPTION)	83
Allow object differences (ALWOBJDIF)	83
Force object conversion (FRCOBJCVN)	85
Object ID (OBJID)	85
Private authorities (PVTAUT)	85
Examples	86
Error messages	87

Restore Authority (RSTAUT)	91
Parameters	92
User profile (USRPRF).	92
Saved from ASP device (SAVASPDEV)	92
Restore to ASP device (RSTASPDEV)	93
Examples	93
Error messages	94

Restore Configuration (RSTCFG)	97
Parameters	97
Objects (OBJ)	98
Device (DEV).	99
Object types (OBJTYPE)	99
Volume identifier (VOL)	100
Sequence number (SEQNBR)	101
End of media option (ENDOPT)	101
Save file (SAVF)	101
System Resource Management (SRM)	102
Optical file (OPTFILE)	102
Allow object differences (ALWOBJDIF)	102
Output (OUTPUT).	103

File to receive output (OUTFILE)	104
Output member options (OUTMBR)	104
Objects to omit (OMITOBJ)	105
Examples	106
Error messages	106

Restore Deferred Objects

(RSTDFROBJ)	109
Parameters	109
Defer ID (DFRID)	109
Output (OUTPUT)	110
File to receive output (OUTFILE)	110
Output member options (OUTMBR)	110
Type of output information (INFTYPE)	111
Examples	111
Error messages	111

Restore Document Lib Object

(RSTDLO)	113
Parameters	114
Document library object (DLO)	115
Saved folder (SAVFLR)	115
New object name (RENAME)	116
Restore into folder (RSTFLR)	116
Device (DEV)	116
System object name (SYSOBJNAM)	117
Volume identifier (VOL)	117
Sequence number (SEQNBR)	117
End of media option (ENDOPT)	118
Label (LABEL)	118
Save file (SAVF)	119
Optical file (OPTFILE)	119
Object name generation (NEWOBJ)	119
Date when saved (SAVDATE)	120
Time when saved (SAVTIME)	120
Allow object differences (ALWOBJDIF)	120
Saved from ASP number (SAVASP)	121
Restore to ASP number (RSTASP)	121
Output (OUTPUT)	121
File to receive output (OUTFILE)	122
Output member options (OUTMBR)	122
Examples	123
Error messages	124

Restore Library (RSTLIB) 127

Parameters	128
Saved library (SAVLIB)	129
Device (DEV)	131
Volume identifier (VOL)	131
Sequence number (SEQNBR)	132
Label (LABEL)	132
End of media option (ENDOPT)	133
Starting library (STRLIB)	133
Save file (SAVF)	134
Media definition (MEDDFN)	134
Optical file (OPTFILE)	135
Libraries to omit (OMITLIB)	135
Option (OPTION)	135
Data base member option (MBROPT)	136
Defer ID (DFRID)	136

Spooled file data (SPLFDTA)	137
Private authorities (PVTAUT)	137
Date when saved (SAVDATE)	137
Time when saved (SAVTIME)	137
Allow object differences (ALWOBJDIF)	138
Force object conversion (FRCOBJCVN)	139
Restore to library (RSTLIB)	140
Restore to ASP device (RSTASPDEV)	140
Restore to ASP number (RSTASP)	141
Output (OUTPUT)	141
File to receive output (OUTFILE)	141
Output member options (OUTMBR)	142
Type of output information (INFTYPE)	142
Objects to omit (OMITOBJ)	143
Examples	144
Error messages	146

Restore Licensed Program

(RSTLICPGM)	151
Parameters	151
Product (LICPGM)	152
Device (DEV)	152
Optional part to be restored (OPTION)	153
Type of object to be restored (RSTOBJ)	153
Language for licensed program (LNG)	154
Output (OUTPUT)	154
Release (RLS)	154
Replace release (REPLACERLS)	155
Volume identifier (VOL)	155
Sequence number (SEQNBR)	155
End of media option (ENDOPT)	156
Save file (SAVF)	156
Library (LIB)	157
Language library (LNGLIB)	157
Folder (FLR)	157
Code home directory (CODHOMEDIR)	157
Language home directory (LNGHOMEDIR)	158
Force object conversion (FRCOBJCVN)	158
Examples	159
Error messages	160

Restore Object (RSTOBJ) 161

Parameters	162
Objects (OBJ)	164
Saved library (SAVLIB)	165
Device (DEV)	165
Object types (OBJTYPE)	166
Volume identifier (VOL)	166
Sequence number (SEQNBR)	167
Label (LABEL)	168
End of media option (ENDOPT)	168
Save file (SAVF)	168
Option (OPTION)	169
File member (FILEMBR)	169
Data base member option (MBROPT)	170
Defer ID (DFRID)	170
Spooled file data (SPLFDTA)	171
Private authorities (PVTAUT)	171
Date when saved (SAVDATE)	171
Time when saved (SAVTIME)	171

Allow object differences (ALWOBJDIF)	172
Force object conversion (FRCOBJCVN)	173
Restore to library (RSTLIB)	174
Output (OUTPUT)	174
Media definition (MEDDFN)	175
Optical file (OPTFILE)	175
Restore to ASP device (RSTASPDEV)	175
Restore to ASP number (RSTASP)	176
File to receive output (OUTFILE)	176
Output member options (OUTMBR)	176
Type of output information (INFTYPE)	177
Libraries to omit (OMITLIB)	177
Objects to omit (OMITOBJ)	178
Examples	178
Error messages	180

Restore Performance Collection (RSTPFCOL) 183

Parameters	183
Collection (COL)	183
Saved library (SAVLIB)	184
Collection type (COLTYPE)	184
Save file (SAVF)	185
Restore to library (RSTLIB)	185
Examples	185
Error messages	186

Restore S/36 File (RSTS36F) 187

Parameters	189
To file (TOFILE)	189
To library (TOLIB)	189
Device (DEV)	190
Set identifier (SET)	190
User specified DBCS data (IGCDTA)	190
File label (FROMLABEL)	191
Creation date (CRTDATE)	191
Sequence number (SEQNBR)	191
Volume identifier (VOL)	191
End of tape option (ENDOPT)	191
Physical file (PHYFILE)	192
Data base member option (MBROPT)	192
Date differentiated file (DATDIFF)	192
Examples	193
Error messages	194

Restore S/36 Library Members (RSTS36LIBM) 197

Parameters	198
To member (TOMBR)	199
To library (TOLIB)	199
Device (DEV)	199
S/36 source members (SRCMBRS)	199
Data base member option (MBROPT)	200
User specified DBCS data (IGCDTA)	200
File label (FROMLABEL)	200
Creation date (CRTDATE)	200
Sequence number (SEQNBR)	200
Volume identifier (VOL)	201
End of tape option (ENDOPT)	201
Physical file (PHYFILE)	201

S/36 object members (OBJMBRS)	202
Examples	202
Error messages	202

Restore System Information (RSTSYSINF) 205

Parameters	205
Device (DEV)	206
Volume identifier (VOL)	206
Sequence number (SEQNBR)	207
End of media option (ENDOPT)	207
Save file (SAVF)	207
Optical file (OPTFILE)	208
Output (OUTPUT)	208
File to receive output (OUTFILE)	208
Output member options (OUTMBR)	209
Examples	209
Error messages	210

Restore User Profiles (RSTUSRPRF) 211

Parameters	212
Device (DEV)	212
User profile (USRPRF)	213
Volume identifier (VOL)	213
Sequence number (SEQNBR)	214
End of media option (ENDOPT)	214
Save file (SAVF)	215
Mail (MAIL)	215
Allow object differences (ALWOBJDIF)	215
User profiles to omit (OMITUSRPRF)	216
Security data (SECDTA)	217
Omit security data (OMITSECDTA)	217
Output (OUTPUT)	218
Optical file (OPTFILE)	218
Saved from ASP device (SAVASPDEV)	218
File to receive output (OUTFILE)	219
Output member options (OUTMBR)	219
Examples	220
Error messages	220

Return from Subroutine (RTNSUBR) 223

Parameters	223
Return value (RTNVAL)	223
Examples	223
Error messages	224

Retrieve Auth List Entry (RTVAUTLE) 225

Parameters	225
Authorization list (AUTL)	226
User (USER)	226
CL var for *ALL (10) (ALL)	226
CL var for *CHANGE (10) (CHANGE)	226
CL var for *USE (10) (USE)	226
CL var for *EXCLUDE (10) (EXCLUDE)	227
CL var for *OBJALTER (10) (OBJALTER)	227
CL var for *OBJEXIST (10) (OBJEXIST)	227
CL var for *OBJMGT (10) (OBJMGT)	227
CL var for *OBJOPR (10) (OBJOPR)	227
CL var for *OBJREF (10) (OBJREF)	227
CL var for *READ (10) (READ)	228

CL var for *ADD (10) (ADD)	228
CL var for *UPD (10) (UPDATE)	228
CL var for *DLT (10) (DELETE)	228
CL var for *EXECUTE (10) (EXECUTE)	228
CL var for *AUTLMGT (10) (AUTLMGT)	229
Examples	229
Error messages	229

Retrieve Backup Options (RTVBCKUP) 231

Parameters	231
Backup options (BCKUPOPT)	231
CL var for DEV (43) (DEV)	232
CL var for TAPSET (34) (TAPSET)	232
CL var for CLRTAP (4) (CLRTAP)	232
CL var for SBMJOB (4) (SBMJOB)	232
CL var for CHGONLY (4) (CHGONLY)	232
CL var for PRTRPT (4) (PRTRPT)	232
CL var for LIB (10) (LIB)	232
CL var for FLR (10) (FLR)	233
CL var for DIR (10) (DIR)	233
CL var for SECDDTA (4) (SECDDTA)	233
CL var for CFG (4) (CFG)	233
CL var for MAIL (4) (MAIL)	233
CL var for CAL (4) (CAL)	233
CL var for EXITPGM (10) (EXITPGM)	233
CL var for EXITPGMLIB (10) (EXITPGMLIB)	234
Examples	234
Error messages	234

Retrieve Binder Source (RTVBNDSRC) 235

Parameters	235
Module (MODULE)	236
Service program (SRVPGM)	236
Export source file (SRCFILE)	237
Export source member (SRCMBR)	237
Replace or add records (MBROPT)	238
Examples	238
Error messages	238

Retrieve Configuration Source (RTVCFGSRC) 241

Parameters	241
Configuration description (CFGD)	241
Type (CFGTYPE)	242
Source file (SRCFILE)	243
Source member (SRCMBR)	243
Retrieve option (RTVOPT)	243
Member option (MBROPT)	244
Text 'description' (TEXT)	244
Examples	244
Error messages	244

Retrieve Configuration Status (RTVCFGSTS) 247

Parameters	247
Configuration description (CFGD)	247
Type (CFGTYPE)	247
CL variable for status code (STSCDE)	248

Examples	249
Error messages	249

Retrieve C Locale Description (RTVCLDSRC) 251

Error messages for RTVCLDSRC	251
Parameters	251
Locale name (CLD)	251
Source file (SRCFILE)	252
Source member (SRCMBR)	252
Text description (TEXT)	253
Examples	253
Error messages	253

Retrieve Cleanup (RTVCLNUP) 255

Parameters	255
CL var for ALWCLNUP (4) (ALWCLNUP)	255
CL var for STRTIME (10) (STRTIME)	255
CL var for USRMSG (5) (USRMSG)	256
CL var for SYSMMSG (5) (SYSMMSG)	256
CL var for CRITSYSMSG (5) (CRITSYSMSG)	256
CL var for SYSPRT (5) (SYSPRT)	256
CL var for SYSLOG (5) (SYSLOG)	256
CL var for JOBQ name (10) (JOBQ)	256
CL var for JOBQ library (10) (JOBQLIB)	257
CL var for RUNPTY (2 0) (RUNPTY)	257
CL var for JRNRCVSIZ (7 0) (JRNRCVSIZ)	257
CL var for CALITM (5) (CALITM)	257
Examples	257
Error messages	258

Retrieve CL Source (RTVCLSRC) 259

Parameters	259
Program (PGM)	259
Source file (SRCFILE)	260
Source member (SRCMBR)	260
Retrieve included source (RTVINCSRC)	260
Examples	261
Error messages	261

Retrieve Current Directory (RTVCURDIR) 263

Parameters	263
CL var for RTNDIR (9999) (RTNDIR)	263
CL var for DIRNAMLEN (7 0) (DIRNAMLEN)	263
Examples	264
Error messages	264

Retrieve Directory Information (RTVDIRINF) 265

Parameters	265
Directory (DIR)	265
Directory subtree (SUBTREE)	266
Information file prefix (INFFILEPFX)	266
Information library (INFLIB)	266
Examples	266
Error messages	267

Retrieve DLO Authority (RTVDLOAUT) 269

Parameters	269
Document library object (DLO)	269
Folder (FLR).	270
Document identifier (DOCID)	270
LADN timestamp (LADNTSP)	270
System object name (SYSOBJNAM)	271
Starting user authority entry (STRUSRAUTE).	271
CL var for OWNER (10) (OWNER)	271
CL var for AUTL (10) (AUTL)	271
CL var for SENSITIV (20) (SENSITIV)	272
CL var for PUBAUT (10) (PUBAUT).	272
CL var for CHKOUTUSR (32) (CHKOUTUSR)	272
CL var for ACC (220) (ACC)	273
CL var for USRAUT (1020) (USRAUT)	273
CL var for GRPAUT (340) (GRPAUT)	273
CL var for PGP (10) (PGP)	273
Examples.	273
Error messages	274

Retrieve DLO Name (RTVDLONAM) 275

Parameters	275
Document library object (DLO)	275
Folder (FLR).	276
Document identifier (DOCID)	276
LADN timestamp (LADNTSP)	277
System object name (SYSOBJNAM)	277
Object class (OBJCLS)	277
CL var for RTNDLO (12) (RTNDLO)	277
CL var for RTNFLR (63) (RTNFLR)	278
CL var for RTNDOCID (24) (RTNDOCID).	278
CL var for RTNLADNTSP (16) (RTNLADNTSP)	278
CL var for RTNOBJNAM (10) (RTNOBJNAM)	278
CL var for RTNOBJCLS (8) (RTNOBJCLS)	278
CL var for RTNASP (2 0) (RTNASP).	278
CL var for RTNOVRFLW (1) (RTNOVRFLW).	278
Examples.	279
Error messages	279

Retrieve Document (RTVDOC) 281

Parameters	281
From document (FROMDOC)	282
From folder (FROMFLR)	282
File to receive output (OUTFILE).	282
Output member options (OUTMBR).	283
Type of data for output (OUTDTATYP).	283
User identifier (USRID)	284
Document identifier (DOCID)	285
Check out (CHKOUT)	285
Command character identifier (CMDCHRID).	286
Examples.	286
Error messages	287

Retrieve Disk Information (RTVDSKINF) 289

Parameters	289
ASP device (ASPDEV)	289
Examples.	290
Error messages	290

Retrieve Data Area (RTVDTAARA) . . . 291

Parameters	292
Data area specification (DTAARA)	292
CL variable for returned value (RTNVAR).	293
Examples.	293
Error messages	294

Retrieve Group Attributes (RTVGRPA) 297

Parameters	297
CL var for GRPJOB (10) (GRPJOB)	297
CL var for GRPJOB list (1056) (GRPJOBL)	298
CL var for GRPJOBcnt (3 0) (GRPJOBcnt)	298
CL var for MSGQ (10) (MSGQ)	298
CL var for MSGQLIB (10) (MSGQLIB)	298
CL var for PRVGRPJOB (16) (PRVGRPJOB)	298
CL var for CTLCDE (3 0) (CTLCDE).	299
Examples.	299
Error messages	300

Retrieve Image Catalog (RTVIMGCLG) 301

Parameters	301
Image catalog (IMGCLG)	301
Virtual device (DEV)	301
CL var for DEVL0D (10) (DEVL0D)	302
CL var for IMGCLGL0D (10) (IMGCLGL0D)	302
Examples.	302
Error messages	302

Retrieve Job Attributes (RTVJOBA) 305

Parameters	307
CL var for JOB (10) (JOB)	308
CL var for USER (10) (USER)	308
CL var for NBR (6) (NBR)	308
CL var for CURUSER (10) (CURUSER)	309
CL var for TYPE (1) (TYPE)	309
CL var for SUBTYPE (1) (SUBTYPE).	309
CL var for SYSLIBL (165) (SYSLIBL).	309
CL var for CURLIB (10) (CURLIB)	309
CL var for USRLIBL (2750) (USRLIBL)	310
CL var for ASPGRP (10) (ASPGRP)	310
CL var for LOGLVL (1) (LOGLVL)	310
CL var for LOGSEV (2 0) (LOGSEV).	310
CL var for LOGTYPE (10) (LOGTYPE)	310
CL var for LOGCLPGM (10) (LOGCLPGM)	310
CL var for LOGOUTPUT (10) (LOGOUTPUT)	311
CL var for JOBMSGQMX (2 0) (JOBMSGQMX)	311
CL var for JOBMSGQFL (10) (JOBMSGQFL)	311
CL var for INQMSGRPY (10) (INQMSGRPY)	311
CL var for STSMMSG (7) (STSMMSG)	311
CL var for BRKMSG (7) (BRKMSG)	311
CL var for DEVRCYACN (13) (DEVRCYACN)	311
CL var for RTNCDE (5 0) (RTNCDE)	312
CL var for ENDSTS (1) (ENDSTS)	312
CL var for PRTDEV (10) (PRTDEV)	312
CL var for OUTQ (10) (OUTQ)	312
CL var for OUTQLIB (10) (OUTQLIB)	312
CL var for SPLFACN (10) (SPLFACN)	313
CL var for PRITXT (30) (PRITXT)	313
CL var for PRTKEYFMT (10) (PRTKEYFMT)	313
CL var for SRTSEQ (10) (SRTSEQ)	313
CL var for SRTSEQLIB (10) (SRTSEQLIB)	313

CL var for LANGID (3) (LANGID)	313
CL var for CNTRYID (2) (CNTRYID)	313
CL var for CCSID (5 0) (CCSID)	314
CL var for DFTCCSID (5 0) (DFTCCSID)	314
CL var for CHRIDCTL (10) (CHRIDCTL)	314
CL var for DECFMT (1) (DECFMT)	314
CL var for DATFMT (4) (DATFMT)	314
CL var for DATSEP (1) (DATSEP)	314
CL var for TIMSEP (1) (TIMSEP)	314
CL var for DATE (6) (DATE)	315
CL var for CYMDDATE (7) (CYMDDATE)	315
CL var for DATETIME (20) (DATETIME)	315
CL var for DAYOFWEEK (4) (DAYOFWEEK)	315
CL var for TIMZON (10) (TIMZON)	315
CL var for TIMZONABBR (10) (TIMZONABBR)	315
CL var for TIMZONFULL (50) (TIMZONFULL)	316
CL var for TIMOFFSET (3 0) (TIMOFFSET)	316
CL var for YEAROFS (3 0) (YEAROFS)	316
CL var for ACGCDE (15) (ACGCDE)	316
CL var for SWS (8) (SWS)	316
CL var for RUNPTY (2 0) (RUNPTY)	316
CL var for TIMESLICE (7 0) (TIMESLICE)	317
CL var for PURGE (10) (PURGE)	317
CL var for DFTWAIT (7 0) (DFTWAIT)	317
CL var for SBMMMSGQ (10) (SBMMMSGQ)	317
CL var for SBMMMSGQLIB (10) (SBMMMSGQLIB)	317
CL var for DDMCNV (5) (DDMCNV)	317
CL var for TSEPOOL (10) (TSEPOOL)	318
CL var for THDRSCAFN (20) (THDRSCAFN)	318
CL var for RSCAFNGRP (10) (RSCAFNGRP)	318
Examples	318
Error messages	319

Retrieve Journal Entry (RTVJRNE) 321

Parameters	322
Journal (JRN)	323
Journalized physical file (FILE)	324
Objects (OBJ)	325
Objects (OBJPATH)	327
Directory subtree (SUBTREE)	328
Name pattern (PATTERN)	329
Range of journal receivers (RCVRNG)	330
Starting large sequence number (FROMENTLRG)	331
Starting date and time (FROMTIME)	331
Ending large sequence number (TOENTLRG)	332
Ending date and time (TOTIME)	332
Search (SEARCH)	333
Journal codes (JRNCD)	333
Journal entry types (ENTTYP)	334
Job name (JOB)	334
Program (PGM)	335
User profile (USRPRF)	335
Commit cycle large identifier (CCIDLRG)	335
Dependent entries (DEPENT)	335
File identifier (OBJFID)	336
Object journal identifier (OBJJID)	336
Entry format (ENTFMT)	337
Null value indicators length (NULLINDLEN)	338
Format minimized data (FMTMINDTA)	338
Include entries (INCENT)	339
Starting sequence number (FROMENT)	339

Ending sequence number (TOENT)	339
Commit cycle identifier (CMTCYCID)	340
CL var for RTNSEQLRG (20) (RTNSEQLRG)	340
CL var for RTNJRNCD (1) (RTNJRNCD)	340
CL var for RTNENTTYP (2) (RTNENTTYP)	341
CL var for RTNRCV (10) (RTNRCV)	341
CL var for RTNRCVLIB (10) (RTNRCVLIB)	341
CL var for RTNJRNE (1) (RTNJRNE)	341
CL var for RTNSEQNBR (10 0) (RTNSEQNBR)	347
Examples	347
Error messages	348

Retrieve Library Description (RTVLIBD) 351

Parameters	351
Library (LIB)	352
CL var for TYPE (10) (TYPE)	352
CL var for ASP (2 0) (ASP)	352
CL var for ASPDEV (10) (ASPDEV)	352
CL var for ASPGRP (10) (ASPGRP)	353
CL var for CRTAUT (10) (CRTAUT)	353
CL var for CRTOBJAUD (10) (CRTOBJAUD)	353
CL var for JRNSTS (1) (JRNSTS)	353
CL var for JRN (10) (JRN)	353
CL var for JRNLIB (10) (JRNLIB)	354
CL var for JRNIMG (1) (JRNIMG)	354
CL var for JRNOMTE (1) (JRNOMTE)	354
CL var for INHERIT (1) (INHERIT)	354
CL var for JRNSTRDATE (13) (JRNSTRDATE)	355
CL var for STRJRNRCV (10) (STRJRNRCV)	355
CL var for JRNRCVLIB (10) (JRNRCVLIB)	355
CL var for RCVLIBASP (10) (RCVLIBASP)	355
CL var for RCVLIBGRP (10) (RCVLIBGRP)	355
CL var for INHRULES (1280) (INHRULES)	356
CL var for TEXT (50) (TEXT)	358
Examples	358
Error messages	358

Retrieve Member Description (RTVMBRD) 361

Parameters	361
File (FILE)	362
Member (MBR)	363
CL var for RTNSYSTEM (4) (RTNSYSTEM)	363
CL var for RTNLIB (10) (RTNLIB)	364
CL var for RTNMBR (10) (RTNMBR)	364
CL var for FILEATR (3) (FILEATR)	364
CL var for FILETYPE (5) (FILETYPE)	364
CL var for SRCTYPE (10) (SRCTYPE)	364
CL var for SRCCHGDATE (13) (SRCCHGDATE)	364
CL var for CRTDATE (13) (CRTDATE)	365
CL var for EXPDATE (7) (EXPDATE)	365
CL var for TEXT (50) (TEXT)	365
CL var for NBRCURRCD (10 0) (NBRCURRCD)	365
CL var for NBRDLTRCD (10 0) (NBRDLTRCD)	365
CL var for SHARE (4) (SHARE)	366
CL var for DTASPCSIZE (15 0) (DTASPCSIZE)	366
CL var for ACCPHSIZE (12 0) (ACCPHSIZE)	366
CL var for CHGDATE (13) (CHGDATE)	366
CL var for SAVDATE (13) (SAVDATE)	366

CL var for RSTDATE (13) (RSTDATE)	367
CL var for NBRDTAMBRS (2 0) (NBRDTAMBRS)	367
CL var for USEDATE (7) (USEDATE)	367
CL var for USECOUNT (5 0) (USECOUNT)	367
CL var for RESETDATE (7) (RESETDATE).	367
Examples.	367
Error messages	369

Retrieve Message (RTVMSG). 371

Parameters	371
Message identifier (MSGID)	372
Message file (MSGF)	372
Message data field values (MSGDTA)	372
CL var for 1st level text (MSG)	372
CL var for MSGLEN (5 0) (MSGLEN)	372
CL var for 2nd level text (SECLVL)	373
CL var for SECLVLEN (5 0) (SECLVLEN)	373
CL var for SEV (2 0) (SEV)	373
CL var for ALROPT (9) (ALROPT)	373
CL var for LOGPRB (1) (LOGPRB)	373
Convert to CCSID (CCSID)	373
Message data CCSID (MDTACCSID)	374
CL var for text CCSID (5 0) (TXTCCSID)	374
CL var for data CCSID (5 0) (DTACCSID)	374
Examples.	375
Error messages	375

Retrieve Network Attributes (RTVNETA) 377

Parameters	377
CL var for SYSNAME (8) (SYSNAME)	378
CL var for PNDSYSNAME (8) (PNDSYSNAME)	378
CL var for LCLNETID (8) (LCLNETID).	378
CL var for LCLCPNAME (8) (LCLCPNAME).	379
CL var for LCLLOCNAME (8) (LCLLOCNAME)	379
CL var for DFTMODE (8) (DFTMODE).	379
CL var for NODETYPE (8) (NODETYPE)	379
CL var for DTACPR (10 0) (DTACPR)	379
CL var for DTACPRINM (10 0) (DTACPRINM)	380
CL var for MAXINTSSN (5 0) (MAXINTSSN)	380
CL var for RAR (5 0) (RAR)	380
CL var for NETSERVER (85) (NETSERVER)	380
CL var for ALRSTS (10) (ALRSTS)	381
CL var for ALRPRIFP (10) (ALRPRIFP)	381
CL var for ALRDFTFP (10) (ALRDFTFP)	381
CL var for ALRLOGSTS (7) (ALRLOGSTS)	381
CL var for ALRBCKFP (16) (ALRBCKFP)	382
CL var for ALRRQSFP (16) (ALRRQSFP)	382
CL var for ALRCTLD (10) (ALRCTLD)	382
CL var for ALRHLCNT (5 0) (ALRHLCNT)	382
CL var for ALRFTR (10) (ALRFTR)	383
CL var for ALRFTRLIB (10) (ALRFTRLIB)	383
CL var for MSGQ (10) (MSGQ)	383
CL var for MSGQLIB (10) (MSGQLIB)	383
CL var for OUTQ (10) (OUTQ)	383
CL var for OUTQLIB (10) (OUTQLIB)	384
CL var for JOBACN (10) (JOBACN)	384
CL var for MAXHOP (5 0) (MAXHOP).	384
CL var for DDMACC (10) (DDMACC)	384
CL var for DDMACCLIB (10) (DDMACCLIB)	385

CL var for PCSACC (10) (PCSACC)	385
CL var for PCSACCLIB (10) (PCSACCLIB)	385
CL var for DFTNETTYPE (10) (DFTNETTYPE)	386
CL var for DFTCNLST (10) (DFTCNLST)	386
CL var for ALWANYNET (10) (ALWANYNET)	386
CL var for NWSDOMAIN (8) (NWSDOMAIN)	386
CL var for ALWVRTAPPN (10) (ALWVRTAPPN)	386
CL var for ALWHPRTWR (10) (ALWHPRTWR)	386
CL var for VRTAUTODEV (5 0) (VRTAUTODEV)	387
CL var for HPRPHTMR (40) (HPRPHTMR)	387
CL var for ALWADDCLU (10) (ALWADDCLU)	387
CL var for MDMCNTRYID (2) (MDMCNTRYID)	387
Examples.	387
Error messages	388

Retrieve Object Description (RTVOBJD) 389

Parameters	389
Object (OBJ)	391
Object type (OBJTYPE)	392
ASP device (ASPDEV)	392
CL var for RTNLIB (10) (RTNLIB)	393
CL var for OBJATR (10) (OBJATR)	393
CL var for USRDFNATR (10) (USRDFNATR)	393
CL var for TEXT (50) (TEXT)	393
CL var for OWNER (10) (OWNER)	393
CL var for PGP (10) (PGP)	393
CL var for ASP (2 0) (ASP)	394
CL var for LIBASP (5 0) (LIBASP)	394
CL var for OBJASPDEV (10) (OBJASPDEV)	394
CL var for OBJASPGRP (10) (OBJASPGRP)	394
CL var for LIBASPDEV (10) (LIBASPDEV)	395
CL var for LIBASPGRP (10) (LIBASPGRP)	395
CL var for OVFAFP (1) (OVFAFP)	395
CL var for CRTDATE (13) (CRTDATE)	395
CL var for CHGDATE (13) (CHGDATE)	395
CL var for SAVDATE (13) (SAVDATE)	396
CL var for SAVACTDATE (13) (SAVACTDATE)	396
CL var for RSTDATE (13) (RSTDATE)	396
CL var for CRTUSER (10) (CRTUSER)	396
CL var for CRTSYSTEM (8) (CRTSYSTEM)	396
CL var for OBJDMN (2) (OBJDMN)	396
CL var for USEUPD (1) (USEUPD)	397
CL var for USEDATE (7) (USEDATE)	397
CL var for USECOUNT (5 0) (USECOUNT)	397
CL var for RESETDATE (7) (RESETDATE).	397
CL var for STG (10) (STG)	397
CL var for CPR (1) (CPR)	397
CL var for SIZE (15 0) (SIZE)	398
CL var for SPCSIZE (15 0) (SPCSIZE)	398
CL var for SPCALIGN (1) (SPCALIGN)	398
CL var for SAVSIZE (15 0) (SAVSIZE)	398
CL var for SAVCMD (10) (SAVCMD)	398
CL var for SAVSEQNBR (4 0) (SAVSEQNBR)	398
CL var for SAVLRGSEQ (10 0) (SAVLRGSEQ)	399
CL var for SAVVOL (71) (SAVVOL)	399
CL var for SAVDEV (10) (SAVDEV)	399
CL var for SAVF (10) (SAVF)	399
CL var for SAVFLIB (10) (SAVFLIB)	399
CL var for SAVLABEL (17) (SAVLABEL)	400
CL var for SRCF (10) (SRCF)	400

CL var for SRCFLIB (10) (SRCFLIB)	400
CL var for SRCMBR (10) (SRCMBR)	400
CL var for SRCDATE (13) (SRCDATE)	400
CL var for SYSLVL (9) (SYSLVL)	400
CL var for COMPILER (16) (COMPILER)	401
CL var for OBJLVL (8) (OBJLVL)	401
CL var for ALWAPICHG (1) (ALWAPICHG)	401
CL var for APICHG (1) (APICHG)	401
CL var for USRCHG (1) (USRCHG)	401
CL var for LICPGM (16) (LICPGM)	402
CL var for PTF (10) (PTF)	402
CL var for APAR (10) (APAR)	402
CL var for OBJAUD (10) (OBJAUD)	402
CL var for OBJSIG (1) (OBJSIG)	402
CL var for SYSSIG (1) (SYSSIG)	403
CL var for MLTSIG (1) (MLTSIG)	403
CL var for JRNSTS (1) (JRNSTS)	403
CL var for JRN (10) (JRN)	403
CL var for JRNLIB (10) (JRNLIB)	403
CL var for JRNIMG (1) (JRNIMG)	403
CL var for JRNOMTE (1) (JRNOMTE)	404
CL var for JRNSTRDATE (13) (JRNSTRDATE)	404
CL var for STRJRNRCV (10) (STRJRNRCV)	404
CL var for JRNRCVLIB (10) (JRNRCVLIB)	404
CL var for RCVLIBASP (10) (RCVLIBASP)	404
CL var for RCVLIBGRP (10) (RCVLIBGRP)	405
Examples	405
Error messages	406

Retrieve PDG Profile (RTVPDGRPF)	409
Parameters	409
User profile (USER)	409
CL var for RTNUSER (RTNUSER)	409
CL var for PDG (PDG)	410
CL var for PDGLIB (PDGLIB)	410
CL var for PRTD (PRTD)	410
Examples	410
Error messages	410

Retrieve Power Schedule Entry (RTVPWRSCDE)	413
Parameters	413
Day (DAY)	413
CL var for PWRONTIME (6) (PWRONTIME)	414
CL var for PWROFFTIME (6) (PWROFFTIME)	414
CL var for DAYDESC (38) (DAYDESC)	414
CL var for MSGITV (2 0) (MSGITV)	414
Examples	414
Error messages	415

Retrieve Query Management Form (RTVQMFORM)	417
Parameters	417
Query management report form (QMFORM)	417
Source file (SRCFILE)	418
Source member (SRCMBR)	418
Allow information from QRYDFN (ALWQRYDFN)	418
Examples	419
Error messages	419

Retrieve Query Mgmt Query (RTVQMQR)	421
Parameters	421
Query management query (QMQR)	421
Source file (SRCFILE)	422
Source member (SRCMBR)	422
Allow information from QRYDFN (ALWQRYDFN)	422
Examples	423
Error messages	423

Retrieve S/36 Environment Attr (RTVS36A)	425
Parameters	425
Environment name (ENV)	426
CL var for SLIB (8) (SLIB)	426
CL var for FLIB (10) (FLIB)	426
CL var for LIBL (4) (LIBL)	426
CL var for DATDIFF (4) (DATDIFF)	426
CL var for S36ESHARE (4) (S36ESHARE)	426
CL var for RCDBLK (4) (RCDBLK)	427
CL var for CACHEDLTF (4) (CACHEDLTF)	427
CL var for LPPAGE (3) (LPPAGE)	427
CL var for FORMTYPE (4) (FORMTYPE)	427
CL var for DFTMSGACN (9) (DFTMSGACN)	427
CL var for HALTOPT (4) (HALTOPT)	427
CL var for EVKJOBINIT (6) (EVKJOBINIT)	427
CL var for EVKJOBPOL (8) (EVKJOBPOL)	428
CL var for EVKJOBPTY (10) (EVKJOBPTY)	428
CL var for SRCRCLEN (3) (SRCRCLEN)	428
CL var for CHGACT (4) (CHGACT)	428
CL var for ADDS36ONLY (4) (ADDS36ONLY)	428
CL var for ICFSUBST (4) (ICFSUBST)	428
CL var for MRTUSRPRF (8) (MRTUSRPRF)	428
CL var for MRTAUT (8) (MRTAUT)	429
CL var for MRTDLY (5) (MRTDLY)	429
CL var for MRTJOBINIT (6) (MRTJOBINIT)	429
CL var for MRTJOBPOL (8) (MRTJOBPOL)	429
CL var for MRTJOBPTY (10) (MRTJOBPTY)	429
Examples	429
Error messages	430

Retrieve Service Agent (RTVSRVAGT)	431
Parameters	431
CL variable for day 1 (DAY1)	431
CL variable for day 2 (DAY2)	431
CL variable for day 3 (DAY3)	432
Examples	432
Error messages	432

Retrieve Stop Word List Source (RTVSWLSRC)	433
Parameters	433
Language ID (LANGID)	433
Source file (SRCFILE)	433
Source member (SRCMBR)	434
Stop word list type (TYPE)	434
Examples	434
Error messages	434

Retrieve System Information (RTVSYSINF) 435

Parameters	435
Library (LIB)	435
Examples	435
Error messages	435

Retrieve System Value (RTVSYSVAL) 437

Parameters	438
System value (SYSVAL)	438
CL variable for returned value (RTNVAR)	468
Examples	472
Error messages	472

Retrieve Table Source (RTVTBLSRC) 473

Parameters	473
Table (TBL)	473
Source file (SRCFILE)	473
Source member (SRCMBR)	474
Convert to UCS-2 (CVTTOUCS)	474
Text 'description' (TEXT)	474
Examples	474
Error messages	475

Retrieve User Profile (RTVUSRPRF) 477

Parameters	477
User profile (USRPRF)	478
CL var for RTNUSRPRF (10) (RTNUSRPRF)	479
CL var for SPCAUT (100) (SPCAUT)	479
CL var for MAXSTG (11 0) (MAXSTG)	479
CL var for STGUSED (15 0) (STGUSED)	479
CL var for PTYLMT (1) (PTYLMT)	479
CL var for INLPGM (10) (INLPGM)	480
CL var for INLPGMLIB (10) (INLPGMLIB)	480
CL var for JOBD (10) (JOBD)	480
CL var for JOBDLIB (10) (JOBDLIB)	480
CL var for GRPPRF (10) (GRPPRF)	480
CL var for OWNER (10) (OWNER)	480
CL var for GRPAUT (10) (GRPAUT)	481
CL var for ACGCDE (15) (ACGCDE)	481
CL var for MSGQ (10) (MSGQ)	481
CL var for MSGQLIB (10) (MSGQLIB)	481
CL var for OUTQ (10) (OUTQ)	481
CL var for OUTQLIB (10) (OUTQLIB)	481
CL var for TEXT (50) (TEXT)	481
CL var for PWDCHGDAT (6) (PWDCHGDAT)	482
CL var for USRCLS (10) (USRCLS)	482
CL var for ASTLVL (10) (ASTLVL)	482
CL var for SPCENV (10) (SPCENV)	482
CL var for CURLIB (10) (CURLIB)	482
CL var for INLMNU (10) (INLMNU)	482
CL var for INLMNULIB (10) (INLMNULIB)	483
CL var for LMTCPB (10) (LMTCPB)	483
CL var for DLVRY (10) (DLVRY)	483
CL var for SEV (2 0) (SEV)	483
CL var for PRTDEV (10) (PRTDEV)	483
CL var for ATNPGM (10) (ATNPGM)	483
CL var for ATNPGMLIB (10) (ATNPGMLIB)	483
CL var for USROPT (240) (USROPT)	484
CL var for DSPSGNINF (7) (DSPSGNINF)	484

CL var for PWDEXPITV (5 0) (PWDEXPITV)	484
CL var for PWDEXP (4) (PWDEXP)	484
CL var for STATUS (10) (STATUS)	484
CL var for PRVSIGN (13) (PRVSIGN)	484
CL var for NOTVLDSIGN (11 0) (NOTVLDSIGN)	485
CL var for LMTDEVSSN (7) (LMTDEVSSN)	485
CL var for KBDBUF (10) (KBDBUF)	485
CL var for LANGID (10) (LANGID)	485
CL var for CNTRYID (10) (CNTRYID)	485
CL var for CCSID (5 0) (CCSID)	485
CL var for SRTSEQ (10) (SRTSEQ)	486
CL var for SRTSEQLIB (10) (SRTSEQLIB)	486
CL var for OBJAUD (10) (OBJAUD)	486
CL var for AUDLVL (640) (AUDLVL)	486
CL var for GRPAUTTYP (10) (GRPAUTTYP)	487
CL var for SUPGRPPRF (150) (SUPGRPPRF)	487
CL var for UID (10 0) (UID)	488
CL var for GID (10 0) (GID)	488
CL var for SETJOBATR (160) (SETJOBATR)	488
Character identifier control (CHRIDCTL)	488
CL var for LCLPWDMGT (10) (LCLPWDMGT)	488
CL var for PWDCHGBLK (10) (PWDCHGBLK)	488
Examples	489
Error messages	489

Retrieve User Print Info (RTVUSRPTI) 491

Parameters	491
User (USER)	491
CL var for RTNTEXT (RTNTEXT)	491
Examples	491
Error messages	492

Retrieve WSCST source (RTVWSCST) 493

Parameters	493
Device type (DEVTYPE)	493
Keyboard language type (KBdtype)	494
Manufacturer type and model (MFRTYPMDL)	494
Source member (SRCMBR)	494
Keyboard attached (KBD)	494
Source file (SRCFILE)	495
Text 'description' (TEXT)	495
Examples	495
Error messages	496

Run Backup (RUNBCKUP) 497

Parameters	497
Backup options (BCKUPOPT)	497
Device (DEV)	497
Examples	498
Error messages	498

Run DNS Update (RUNDNSUPD) 499

Parameters	499
Batch input file (BCHFILE)	500
Show debug information (DEBUG)	500
Key name (KEYNAME)	501
Key file (KEYFILE)	501
Update timeout (TIMEOUT)	501
UDP timeout (UDPRTYITV)	502

UDP retry (UDPNBRRTY)	502
Network protocol (PROTOCOL)	502
Output file (TOSTMF)	502
Examples	502
Error messages	505

Run LPDA-2 (RUNLPDA) 507

Parameters	507
Line (LINE)	508
Option (OPTION)	508
Local DCE address (LCLDCEADR)	509
Remote DCE address (RMTDCEADR)	509
Telephone number (TELNBR)	509
Relay contact (CONTACT)	510
Transmit speed (SPEED)	510
DTE port (DTEPORT)	510
DCE retry (DCERTY)	511
Examples	511
Error messages	512

Run Query (RUNQRY) 515

Parameters	515
Query (QRY Parameter)	516
Query file (QRYFILE Parameter)	517
Report output type (OUTTYPE Parameter)	518
Output form (OUTFORM Parameter)	518
Record selection (RCDSLTP Parameter)	518
Print query definition (PRTFDFN Parameter)	519
Print device (PRTDEV Parameter)	519
Form size (FORMSIZE Parameter)	520
Form type (FORMTYPE Parameter)	520
Copies (COPIES Parameter)	520
Line spacing (LINESPACE Parameter)	521
Output file (OUTFILE Parameter)	521
Authority (AUT Parameter)	523
Examples	523
Error messages	524

Run Remote Command (RUNRMTCMD) 525

Parameters	525
Command (CMD)	525
Remote location (RMTLOCNAME)	526
Remote user ID (RMTUSER)	527
Remote password (RMTPWD)	527
Mode (MODE)	527
Coded character set ID (CCSID)	527
Wait time (in seconds) (WAITTIME)	528
Examples	528
Error messages	529

Run RNDC Command (RUNRNDCCMD) 531

Parameters	531
RNDC command (RNDCCMD)	532
Domain name server (DMNNSVR)	533
Domain name server port (PORT)	533
RNDC configuration file (RNDCCFGF)	533
Source address (SRCADR)	533
Key file (KEYFILE)	534

Key name (KEYNAME)	534
Show debug information (DEBUG)	534
Output file (TOSTMF)	534
Examples	535
Error messages	536

Run SQL Statements (RUNSQLSTM) 537

Parameters	537
Source file (SRCFILE)	538
Source member (SRCMBR)	538
Source stream file (SRCSTMF)	538
Commitment control (COMMIT)	539
Naming convention (NAMING)	539
Severity level (ERRLVL)	539
Date format (DATFMT)	540
Date separator character (DATSEP)	540
Time format (TIMFMT)	540
Time separator character (TIMSEP)	541
Source margins (MARGINS)	541
Default collection (DFTRDBCOL)	541
IBM SQL flagging (SAAFLAG)	542
ANS flagging (FLAGSTD)	542
Decimal Point (DECMPT)	542
Sort sequence (SRTSEQ)	542
Language id (LANGID)	543
Print file (PRTFILE)	543
Statement processing (PROCESS)	543
Second level text (SECLVLTX)	544
Allow copy of data (ALWCPYDTA)	544
Allow blocking (ALWBLK)	544
SQL rules (SQLCURRULE)	545
Decimal result options (DECRESULT)	545
Listing output (OUTPUT)	546
Target release (TGTRLS)	546
Debugging view (DBGVIEW)	546
Close SQL cursor (CLOSQLCSR)	547
Delay PREPARE (DLYPRP)	547
User profile (USRPRF)	548
Dynamic user profile (DYNUSRPRF)	548
Examples	548
Error messages	548

Revoke Access Code Authority (RVKACCAUT) 551

Parameters	551
Document access code (ACC)	551
User profile (USER)	551
Examples	552
Error messages	552

Revoke Object Authority (RVKOBJAUT) 553

Parameters	554
Object (OBJ)	554
Object type (OBJTYPE)	556
ASP device (ASPDEV)	556
Users (USER)	556
Authority (AUT)	557
Authorization list (AUTL)	558
Examples	558

Error messages 559

Revoke Public Authority (RVKPUBAUT) 561

Parameters 561
Library (LIB) 561
Examples 562
Error messages 562

Revoke User Permission (RVKUSRPMN) 563

Parameters 563
From user profile (FROMUSER) 563
For user profile (FORUSER) 563
Examples 564
Error messages 564

Revoke Workstation Object Aut (RVKWSOAUT) 565

Parameters 565
Workstation object type (WSOTYPE) 566
Users (USER) 567
Authority (AUT) 568
Authorization list (AUTL) 569
Examples 569
Error messages 569

Save Object (SAV) 571

Parameters 571
Device (DEV) 572
Objects (OBJ) 573
Name pattern (PATTERN) 574
Directory subtree (SUBTREE) 574
Save active (SAVACT) 575
Save active option (SAVACTOPT) 575
Output (OUTPUT) 575
Volume identifier (VOL) 576
Label (LABEL) 576
Optical file (OPTFILE) 577
Sequence number (SEQNBR) 577
File expiration date (EXPDATE) 577
End of media option (ENDOPT) 577
Use optimum block (USEOPTBLK) 578
Save active message queue (SAVACTMSGQ) 578
Synchronization ID (SYNCID) 579
Type of output information (INFTYPE) 579
System (SYSTEM) 579
Time period for last change (CHGPERIOD) 579
Object pre-check (PRECHK) 580
Target release (TGTRLS) 580
Update history (UPDHST) 581
Clear (CLEAR) 581
Data compression (DTACPR) 582
Data compaction (COMPACT) 583
ASP device (ASPDEV) 583
Scan objects (SCAN) 584
Private authorities (PVTAUT) 584
Examples 584
Error messages 587

Save APAR Data (SAVAPARDTA) 591

Parameters 591
Problem identifier (PRBID) 591
Examples 591
Error messages 592

Save Configuration (SAVCFG) 593

Parameters 593
Device (DEV) 594
Volume identifier (VOL) 594
Sequence number (SEQNBR) 595
File expiration date (EXPDATE) 595
End of media option (ENDOPT) 595
Use optimum block (USEOPTBLK) 596
Save file (SAVF) 596
Optical file (OPTFILE) 597
Clear (CLEAR) 597
Object pre-check (PRECHK) 598
Data compression (DTACPR) 598
Data compaction (COMPACT) 599
Output (OUTPUT) 599
File to receive output (OUTFILE) 599
Output member options (OUTMBR) 600
Examples 600
Error messages 601

Save Changed Objects (SAVCHGOBJ) 603

Parameters 604
Objects (OBJ) 606
Library (LIB) 606
Device (DEV) 607
Object types (OBJTYPE) 608
Journaled objects (OBJJRN) 608
Reference date (REFDATE) 608
Reference time (REFTIME) 609
Volume identifier (VOL) 609
Sequence number (SEQNBR) 609
Label (LABEL) 610
File expiration date (EXPDATE) 610
End of media option (ENDOPT) 610
Starting library (STRLIB) 611
Save file (SAVF) 611
Media definition (MEDDFN) 612
Optical file (OPTFILE) 612
Use optimum block (USEOPTBLK) 612
Target release (TGTRLS) 613
Update history (UPDHST) 614
Clear (CLEAR) 614
Object pre-check (PRECHK) 615
Save active (SAVACT) 615
Save active wait time (SAVACTWAIT) 616
Save active message queue (SAVACTMSGQ) 617
Synchronization ID (SYNCID) 618
Save access paths (ACCPATH) 618
Save file data (SAVFDTA) 618
Queue data (QDTA) 619
Private authorities (PVTAUT) 619
Data compression (DTACPR) 619
Data compaction (COMPACT) 620
Libraries to omit (OMITLIB) 620

Objects to omit (OMITOBJ)	621
ASP device (ASPDEV)	622
Output (OUTPUT)	622
File to receive output (OUTFILE)	622
Output member options (OUTMBR)	623
Type of information (INFATYPE)	623
Command user space (CMDUSRSPC)	624
Examples	624
Error messages	624

Save Document Library Object (SAVDLO) 629

Parameters	629
Document library object (DLO)	631
Folder (FLR)	632
Device (DEV)	633
Search type (SRCHTYPE)	633
Check for mark (CHKFORMRK)	633
Expiration date (CHKEXP)	634
Creation date (CRTDATE)	634
Document class (DOCCLS)	635
Owner profile (OWNER)	636
Last changed date (REFCHGDATE)	636
Last changed time (REFCHGTIME)	636
System object name (SYSOBJNAM)	637
Document list (DOCL)	637
Volume identifier (VOL)	637
Sequence number (SEQNBR)	638
File expiration date (EXPDATE)	638
End of media option (ENDOPT)	638
Label (LABEL)	639
Use optimum block (USEOPTBLK)	639
Optical file (OPTFILE)	639
Save file (SAVF)	640
Folders to omit (OMITFLR)	640
Output (OUTPUT)	640
File to receive output (OUTFILE)	641
Output member options (OUTMBR)	641
Clear (CLEAR)	642
Storage (STG)	642
Data compression (DTACPR)	643
Data compaction (COMPACT)	644
Command character identifier (CMDCHRID)	644
Target release (TGTRLS)	644
Save active (SAVACT)	645
Save active wait time (SAVACTWAIT)	645
ASP number (ASP)	645
Examples	646
Error messages	647

Save Library (SAVLIB) 651

Parameters	652
Library (LIB)	653
Device (DEV)	655
Volume identifier (VOL)	656
Sequence number (SEQNBR)	656
Label (LABEL)	656
File expiration date (EXPDATE)	657
End of media option (ENDOPT)	657
Starting library (STRLIB)	657

Save file (SAVF)	658
Media definition (MEDDFN)	658
Optical file (OPTFILE)	659
Use optimum block (USEOPTBLK)	659
Target release (TGTRLS)	659
Update history (UPDHST)	660
Clear (CLEAR)	661
Object pre-check (PRECHK)	661
Save active (SAVACT)	662
Save active wait time (SAVACTWAIT)	662
Save active message queue (SAVACTMSGQ)	663
Synchronization ID (SYNCID)	664
Save access paths (ACCPATH)	664
Save file data (SAVFDTA)	665
Spooled file data (SPLFDTA)	665
Queue data (QDTA)	665
Private authorities (PVTAUT)	666
Storage (STG)	666
Data compression (DTACPR)	666
Data compaction (COMPACT)	667
Libraries to omit (OMITLIB)	667
Objects to omit (OMITOBJ)	668
ASP device (ASPDEV)	669
Output (OUTPUT)	669
File to receive output (OUTFILE)	669
Output member options (OUTMBR)	670
Type of information (INFATYPE)	670
Command user space (CMDUSRSPC)	671
Examples	671
Error messages	673

Save Licensed Program (SAVLICPGM) 677

Parameters	677
Product (LICPGM)	678
Device (DEV)	678
Optional part to be saved (OPTION)	678
Release (RLS)	679
Language for licensed program (LNG)	679
Object type (OBJTYPE)	679
Check signature (CHKSIG)	679
Volume identifier (VOL)	680
Sequence number (SEQNBR)	680
File expiration date (EXPDATE)	680
End of media option (ENDOPT)	681
Save file (SAVF)	681
Target release (TGTRLS)	681
Clear (CLEAR)	682
Data compression (DTACPR)	683
License acceptance required (LICACPRQD)	683
Examples	684
Error messages	685

Save Object (SAVOBJ) 687

Parameters	688
Objects (OBJ)	689
Library (LIB)	690
Device (DEV)	690
Object types (OBJTYPE)	691
Volume identifier (VOL)	691
Sequence number (SEQNBR)	692

Label (LABEL)	692
File expiration date (EXPDATE)	692
End of media option (ENDOPT)	692
Save file (SAVF)	693
Media definition (MEDDFN)	693
Optical file (OPTFILE)	694
Use optimum block (USEOPTBLK)	694
Target release (TGTRLS)	694
Update history (UPDHST)	695
Clear (CLEAR)	696
Object pre-check (PRECHK)	696
Save active (SAVACT)	697
Save active wait time (SAVACTWAIT)	697
Save active message queue (SAVACTMSGQ)	699
Synchronization ID (SYNCID)	699
File member (FILEMBR)	700
Save access paths (ACCPH)	701
Save file data (SAVFDTA)	701
Spooled file data (SPLFDTA)	701
Queue data (QDTA)	702
Private authorities (PVTAUT)	702
Storage (STG)	702
Data compression (DTACPR)	702
Data compaction (COMPACT)	703
Libraries to omit (OMITLIB)	703
Objects to omit (OMITOBJ)	704
ASP device (ASPDEV)	705
Output (OUTPUT)	705
File to receive output (OUTFILE)	706
Output member options (OUTMBR)	706
Type of information (INFTYPE)	707
Command user space (CMDUSRSPC)	707
Examples	707
Error messages	708

Save Performance Collection (SAVPFRCOL) 713

Parameters	713
Collection (COL)	713
Library (LIB)	714
Collection type (COLTYPE)	714
Save file (SAVF)	715
Target release (TGTRLS)	715
Data compression (DTACPR)	716
Examples	716
Error messages	716

Save Restore (SAVRST) 719

Parameters	720
Remote location (RMTLOCNAME)	721
Objects (OBJ)	721
Name pattern (PATTERN)	722
Directory subtree (SUBTREE)	722
Rebuild mounted file system (RBDMFS)	723
Time period for last change (CHGPERIOD)	723
Target release (TGTRLS)	724
Object pre-check (PRECHK)	725
Save active (SAVACT)	725
Save active message queue (SAVACTMSGQ)	725
ASP device (ASPDEV)	726

Option (OPTION)	726
Allow object differences (ALWOBJDIF)	727
Force object conversion (FRCOBJCVN)	728
Scan objects (SCAN)	729
Private authorities (PVTAUT)	729
Create parent directories (CRTPRNDIR)	729
Parent directory owner (PRNDIROWN)	730
Examples	730
Error messages	730

Save Restore Configuration (SAVRSTCFG) 733

Parameters	733
Objects (OBJ)	734
Remote location (RMTLOCNAME)	735
Object types (OBJTYPE)	735
System Resource Management (SRM)	735
Object pre-check (PRECHK)	736
Allow object differences (ALWOBJDIF)	736
Examples	737
Error messages	737

Save Restore Changed Objects (SAVRSTCHG) 739

Parameters	740
Objects (OBJ)	741
Saved library (LIB)	742
Remote location (RMTLOCNAME)	742
Object types (OBJTYPE)	743
Journalized objects (OBJJRN)	743
Reference date (REFDATE)	743
Reference time (REFTIME)	744
Starting library (STRLIB)	744
Libraries to omit (OMITLIB)	744
Objects to omit (OMITOBJ)	745
ASP device (ASPDEV)	746
Target release (TGTRLS)	746
Object pre-check (PRECHK)	746
Save active (SAVACT)	747
Save active wait time (SAVACTWAIT)	747
Save active message queue (SAVACTMSGQ)	748
Save access paths (ACCPH)	749
Save file data (SAVFDTA)	750
Queue data (QDTA)	750
Private authorities (PVTAUT)	750
Restore to library (RSTLIB)	750
Restore to ASP device (RSTASPDEV)	750
Restore to ASP number (RSTASP)	751
Option (OPTION)	751
Data base member option (MBROPT)	752
Allow object differences (ALWOBJDIF)	752
Force object conversion (FRCOBJCVN)	753
Examples	754
Error messages	755

Save Restore Doc/Lib Object (SAVRSTDLO) 757

Parameters	758
Document library object (DLO)	759
Remote location (RMTLOCNAME)	759

Folder (FLR)	760
Folders to omit (OMITFLR)	761
Search type (SRCHTYPE)	761
Expiration date (CHKEXP)	761
Creation date (CRTDATE)	762
Document class (DOCCLS)	763
Owner profile (OWNER)	763
Last changed date (REFCHGDATE)	764
Last changed time (REFCHGTIME)	764
Document list (DOCL)	765
Target release (TGTRLS)	765
Storage (STG)	765
Command character identifier (CMDCHRID)	766
Save active (SAVACT)	766
Save active wait time (SAVACTWAIT)	767
ASP number (ASP)	767
Object name generation (NEWOBJ)	767
Allow object differences (ALWOBJDIF)	767
Restore to ASP number (RSTASP)	768
Examples	768
Error messages	769

Save Restore Library (SAVRSTLIB) 771

Parameters	772
Library (LIB)	773
Remote location (RMTLOCNAME)	774
Starting library (STRLIB)	775
Libraries to omit (OMITLIB)	775
Objects to omit (OMITOBJ)	775
ASP device (ASPDEV)	776
Target release (TGTRLS)	777
Object pre-check (PRECHK)	777
Save active (SAVACT)	777
Save active wait time (SAVACTWAIT)	778
Save active message queue (SAVACTMSGQ)	779
Save access paths (ACCPATH)	780
Save file data (SAVFDATA)	780
Spooled file data (SPLFDATA)	780
Queue data (QDTA)	781
Private authorities (PVTAUT)	781
Storage (STG)	781
Option (OPTION)	781
Data base member option (MBROPT)	782
Allow object differences (ALWOBJDIF)	782
Restore to library (RSTLIB)	784

Restore to ASP device (RSTASPDEV)	784
Restore to ASP number (RSTASP)	784
Force object conversion (FRCOBJCVN)	785
Examples	785
Error messages	786

Save Restore Object (SAVRSTOBJ) 787

Parameters	787
Objects (OBJ)	789
Saved library (LIB)	789
Remote location (RMTLOCNAME)	790
Object types (OBJTYPE)	790
Starting library (STRLIB)	791
Libraries to omit (OMITLIB)	791
Objects to omit (OMITOBJ)	792
ASP device (ASPDEV)	793
Target release (TGTRLS)	793
Object pre-check (PRECHK)	793
Save active (SAVACT)	794
Save active wait time (SAVACTWAIT)	794
Save active message queue (SAVACTMSGQ)	795
File member (FILEMBR)	796
Save access paths (ACCPATH)	797
Save file data (SAVFDATA)	798
Spooled file data (SPLFDATA)	798
Queue data (QDTA)	798
Private authorities (PVTAUT)	798
Storage (STG)	798
Option (OPTION)	799
Data base member option (MBROPT)	799
Allow object differences (ALWOBJDIF)	800
Restore to library (RSTLIB)	801
Restore to ASP device (RSTASPDEV)	801
Restore to ASP number (RSTASP)	801
Force object conversion (FRCOBJCVN)	802
Examples	803
Error messages	803

Appendix. Notices 805

Programming interface information	806
Trademarks	807
Terms and conditions	808
Code license and disclaimer information	808

Remove Trace Filter (RMVTRCFTR)

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

Parameters
Examples
Error messages

The Remove Trace Filter (RMVTRCFTR) command removes one or more trace filters from the system.

Restrictions:

- You must have service (*SERVICE) special authority, or be authorized to the Service trace function of i5/OS through System i Navigator's Application Administration support. The Change Function Usage Information (CHGFCNUSG) command can also be used to change the list of users that are allowed to perform trace operations.

Top

Parameters

Keyword	Description	Choices	Notes
FTR	Filter	Generic name, name	Required, Positional 1

Top

Filter (FTR)

Specifies the trace filter to be removed. A specific or generic filter name can be specified.

This is a required parameter.

generic-name

Specify the generic name of the trace filter to be removed. A generic name is a character string of one or more characters followed by an asterisk (*); or example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

name Specify the name of the trace filter to be removed.

Top

Examples

Example 1: Removing a Single Filter

```
RMVTRCFTR FTR(SAMPLE)
```

This command removes the trace filter named SAMPLE.

Example 2: Removing All Filters that Start with SAM

RMVTRCFTR FTR(SAM*)

This command removes all trace filters with names that start with SAM.

[Top](#)

Error messages

Unknown

[Top](#)

Remove Work Station Entry (RMVWSE)

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

Parameters
Examples
Error messages

The Remove Work Station Entry (RMVWSE) command removes a work station entry from the specified subsystem description.

Restrictions:

- To use this command, you must have:
 - object operational (*OBJOPR), object management (*OBJMGT), and read (*READ) authority to the specified subsystem description and execute (*EXECUTE) authority to the library containing the subsystem description.
- Work station entries in the subsystem description of an active subsystem may not be removed if there are active jobs associated with the entry.
- A *CON or CONS entry may not be removed from the controlling subsystem.

Top

Parameters

Keyword	Description	Choices	Notes
SBSD	Subsystem description	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Subsystem description	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
WRKSTN	Work station name	<i>Generic name, name</i>	Optional
WRKSTNTYPE	Work station type	*ALL, 3179, 3180, 3196, 3197, 3277, 3278, 3279, 3476, 3477, 3486, 3487, 5251, 5252, 5291, 5292, 5555, *ASCII, CONS, *CONS, *NONASCII	Optional

Top

Subsystem description (SBSD)

Specifies the name and library of the subsystem description containing the work station job entry that is removed.

This is a required parameter.

Qualifier 1: Subsystem description

name Specify the name of the subsystem description where the work station job entry is being removed.

Note: The following IBM-supplied objects are not valid on this parameter:

- QLPINSTALL
- QSYSSBSD

Qualifier 2: Library

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

***CURLIB**

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

name Specify the library where the subsystem description is located.

Top

Work station name (WRKSTN)

Specifies the name of the work station used by the subsystem. The device description name that was specified in the Create Device Desc (Display) (CRTDEV DSP) command associated with the work station is the name used.

Double-Byte Character Set Considerations: For double-byte character set (DBCS), a work station whose type is 5555 must be specified for either this parameter or the **Work station type (WRKSTNTYPE)** parameter, but not for both.

generic-name

Specify a generic name. Examples include: DSP*, RMT*,...

Note: Specifying a generic work station name does not result in multiple entries being added, changed, or removed.

name Specify the name of a specific work station. Examples include: DSP10, DSP11, RMT55,...

A value must be specified on either this parameter or the **Work station type (WRKSTNTYP)** parameter, but not for both.

Top

Work station type (WRKSTNTYPE)

Specifies the type of work station associated with the entry being added, changed, or removed. This entry applies to all work stations of this type that do not have specific entries for an individual work station.

***ALL** All work station devices. This includes devices with 5250, ASCII, and 327x device types.

3179 3179 work station.

3180 3180 work station.

3196 3196 work station.

3197 3197 work station.

3277 3277 work station.

3278 3278 work station.

3279 3279 work station.

3476 3476 work station.

3477 3477 work station.

3486 3486 work station.

3487 3487 work station.

5251 5251 work station.

5291 5291 work station.

5292 5292 color work station.

5555 5555 double-byte character set (DBCS) capable work station.

***ASCII**

All ASCII work station device types.

CONS System console display. This entry overrides a device type entry that specifies the same device type as the device being used as the console.

***CONS**

System console display. This entry overrides a device type entry that specifies the same device type as the device being used as the console.

***NONASCII**

All work station devices that use the 5250 data stream, as well as, 327x device types.

A value must be specified on either this parameter or the **Work station name (WRKSTN)** parameter, but not for both.

Top

Examples

```
RMVWSE  SBSDB(LIB2/CHARLES)  WRKSTN(B53)
```

This command removes the work station entry for work station B53 from the subsystem description named CHARLES in library LIB2.

Top

Error messages

***ESCAPE Messages**

CPF1619

Subsystem description &1 in library &2 damaged.

CPF1691

Active subsystem description may or may not have changed.

CPF1697

Subsystem description &1 not changed.

Top

Run RNDC Command (RNDC)

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

Parameters
Examples
Error messages

The Run RNDC Command (RUNRNDCCMD) command, or its alias RNDC, starts the Remote Name Daemon Control utility. This command allows you to control a Domain Name System (DNS) running on your local system.

Restrictions:

- You must have execute (*X) authority to the directories in the path `/QIBM/UserData/OS400/DNS/_DYN`
- You must have read (*R) authority to the `/QIBM/UserData/OS400/DNS/_DYN` directory files.
- You must have execute (*X) authority to the directories in the path of the RNDC configuration file.
- You must have read (*R) authority to the RNDC configuration file.
- You must have execute (*X) authority to the directories in the path of the RNDC key file.
- You must have read (*R) authority to the key file.
- You must have execute (*X) authority to the directories in the path of the output file.
- You must have write (*W) authority to the output file if it already exists.
- You must have read, write and execute (*RWX) authority to the output file's parent directory if the output file does not already exist.

Top

Parameters

Keyword	Description	Choices	Notes
RNDCCMD	RNDC command	<i>Character value</i>	Required, Positional 1
DMNNAMSVR	Domain name server	<i>Character value, *DFT</i>	Optional, Positional 2
PORT	Domain name server port	1-65535, <u>953</u>	Optional, Positional 3
RNDCCFGF	RNDC configuration file	<i>Path name, *DFT</i>	Optional
SRCADR	Source address	<i>Character value, *DFT, *ANY4, *LOOPBACK4, *ANY6, *LOOPBACK6</i>	Optional
KEYFILE	Key file	<i>Path name, *DFT</i>	Optional
KEYNAME	Key name	<i>Character value, *DFT</i>	Optional
DEBUG	Show debug information	<u>*NO</u> , *YES	Optional
TOSTMF	Output file	<i>Path name, *STDOUT</i>	Optional

Top

RNDC command (RNDCCMD)

Specifies the control command to be sent to the DNS server.

This is a required parameter.

character-value

Specify the control command to run. The command must be one of the following:

```
reload
  Reload configuration file and zones
reload zone [class [view]]
  Reload a single zone
refresh zone [class [view]]
  Schedule immediate maintenance for a zone
retransfer zone [class [view]]
  Re-transfer a single zone without checking
  serial number
freeze zone [class [view]]
  Suspend updates to a dynamic zone
thaw zone [class [view]]
  Enable updates to a frozen dynamic zone and
  reload it
reconfig
  Reload configuration file and new zones only
stats
  Write server statistics to the statistics file
querylog
  Toggle query logging
dumpdb [-all|-cache|-zones] [view ...]
  Dump cache(s) to the dump file (named_dump.db)
stop
  Save pending updates to master files and stop
  the server
stop -p
  Save pending updates to master files and stop
  the server reporting process id
halt
  Stop the server without saving pending updates
halt -p
  Stop the server without saving pending updates
  reporting process id
trace
  Increment debugging level by one
trace level
  Change the debugging level
notrace
  Set debugging level to 0
flush
  Flushes all of the servers caches
flush [view]
  Flushes the servers cache for a view
flushname name [view]
  Flush the given name from the servers cache(s)
status
  Display status of the server
recurring
  Dump the queries that are currently recurring
  (named.recurring)
```

Top

Domain name server (DMNNAMSVR)

Specifies the name or the IP address of the DNS server that RNDC will use as its current server for the query session. You can specify any DNS server to which your TCP/IP network has access.

Note: RNDC sends information to DNS servers and it needs an active DNS server to send its commands. If you do not specify a DNS server with DMNNAMSVR when you start the tool, it will attempt to use the default server, port and key defined in the `rndc.conf` file. This is usually the loopback interface 127.0.0.1 and port 953.

***DFT** Use the default DNS server defined in the `/QIBM/UserData/OS400/DNS/_DYN/rndc.conf` file.

server-domain-name

Specify the name of a DNS server. This is a domain name like 'myserver.i5os.ibm.com'.

server-internet-address

Specify the IP address of a DNS server.

RNDC-server-name

Specify the name of a DNS server in the `rndc.conf` file. This can be the name like 'NS'.

Top

Domain name server port (PORT)

Specifies the default server port to use.

953 Use control channel port 953.

1-65535

Specify a valid port number.

Top

RNDC configuration file (RNDCCFGF)

Specifies the RNDC configuration file to be used for this session. This file contains both access key and option statements that define the default server and the access key for that server. It is possible to use RNDC to control DNS servers located on other systems if this configuration file contains the access key and server statements that apply to the remote servers.

***DFT** Use `/QIBM/UserData/OS400/DNS/_DYN/rndc.conf` as the default configuration file. The default server in this file is 'localhost'.

path-name

Specify the path name for a stream file containing RNDC configuration information. For example, `'/home/myprofile/my-rndc-conf-file'`.

Top

Source address (SRCADR)

Specifies the source address for the connection to the server.

***DFT** Uses the default supplied by the stack.

***ANY4**

Use the IPv4 wildcard address ('0.0.0.0').

***LOOPBACK4**

Use the IPv4 loopback address ('127.0.0.1').

***ANY6**

Use the IPv6 wildcard address ('::').

***LOOPBACK6**

Use the IPv6 loopback address ('::1').

character-value

Specify a valid IPv4 or IPv6 internet address.

Top

Key file (KEYFILE)

Specifies the access key file to use for this session. The key in this file will be used to authenticate commands sent to the server.

***DFT** Use /QIBM/UserData/OS400/DNS/_DYN/rndc.key as the default key file.

path-name

Specify the path name for a stream file containing key information. For example, '/home/myprofile/my-rndc-key-file'.

Top

Key name (KEYNAME)

Specifies the access key name to be used for this session. This key name must be known by the server that is being used for this session. If no key name is specified, RNDC will use the defaults from the /QIBM/UserData/OS400/DNS/_DYN/rndc.conf file.

***DFT** Use key name *rndc-key*.

character-value

Specify the RNDC key name to use.

The DNS server being queried needs to include this key and algorithm in its named.conf configuration file in order to allow RNDC clients with this key.

Top

Show debug information (DEBUG)

Specifies whether or not to turn debugging mode on. More information is displayed about the packet sent to the server and the resulting answer when debugging mode is on.

***NO** Turn off debugging messages.

***YES** Turn on debugging messages.

Top

Output file (TOSTMF)

Specifies the name of a stream file where all command output is written.

***STDOUT**

All command output goes to the standard output device (normally the display).

path-name

Specify the path for a stream file where output should be written.

Top

Examples

Example 1: Reload All DNS Server Configuration and Static Zones

```
RUNRNDC CMD RNDCCMD('reload')
```

This command illustrates a simple reload of any changes to a DNS server configuration and any static zones.

The output from this type of query might look like this:

```
server reload successful
```

Similarly, if your RNDCC key from the rndc.conf file is not valid, the output from this type of query might look like this:

```
rndc: connection to remote host closed
This may indicate that
* the remote server is using an older version of the
  command protocol,
* this host is not authorized to connect,
* the clocks are not synchronized, or
* the key is invalid.
```

Example 2: Reload a Single Zone

```
RUNRNDC CMD RNDCCMD('reload i5os.ibm.com')
```

This command illustrates a simple reload of a single static zone called 'i5os.ibm.com'.

The output from this type of query might look like this:

```
zone reload up-to-date
```

If the zone is not a static zone, the output from this type of query might look like this:

```
rndc: 'reload i5os.ibm.com' failed: dynamic zone
```

Example 3: Dumping the Cache

```
RUNRNDC CMD RNDCCMD('dumpdb -cache')
```

This command illustrates how to dump the active cache on the server. The output from this type of query will be empty, because it goes to file named_dump.db in the server directory, i.e., for server NS the file would be /QIBM/UserData/OS400/DNS/NS/named_dump.db.

Top

Error messages

*ESCAPE Messages

DNS0013

Error processing command parameters.

DNS0065

Option 33 of i5/OS is required, but is not installed.

TCP7124

Program &1 in library &2 type *PGM ended abnormally.

Top

Rename Object (RNM)

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

Parameters
Examples
Error messages

The Rename Object (RNM) command changes the name of an object in a directory.

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

- REN

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

Restrictions:

- This command works on only one object. If a pattern is specified on the **Object (OBJ)** parameter and more than one object matches the pattern, the user can select the object from a list in an interactive job. If this is a batch job, the command fails with error message CPFA08E, "More than one name matches pattern."
- When renaming an object in the "root" (/), QOpenSys or user-defined file systems, the user must have object management (*OBJMGT) authority to the object to be renamed, and write and execute (*WX) authority to the directory that contains the object. If the object to be renamed is a directory, the user must also have write (*W) authority to the directory.
- The user must have execute (*X) authority to each directory in the path.
- A user cannot rename an object within a "root" (/), QOpenSys, or user-defined file system directory that has the "restricted rename and unlink" attribute set on (this attribute is equivalent to the S_ISVTX mode bit) unless one or more of the following are true:
 - The user is the owner of the object.
 - The user is the owner of the directory.
 - The user has all object (*ALLOBJ) special authority.
- The authority requirements and restrictions from the existing Rename Object (RNMOBJ) command and Rename Document Library Object (RNMDLO) command apply to objects in the QSYS.LIB, independent ASP QSYS.LIB, and QDLS file systems.
- In the QSYS.LIB and independent ASP QSYS.LIB file systems, the new name must contain the same object type suffix.
- Some objects cannot be renamed. An error is returned if an attempt is made to rename these objects.
- The file cannot be renamed if the file is a DataLink column in an SQL table and where a row in that SQL table references this file.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Path name</i>	Required, Positional 1
NEWOBJ	New object	<i>Character value</i>	Required, Positional 2

Object (OBJ)

Specifies the path name of the object to be renamed.

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

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

Top

New object (NEWOBJ)

Specifies the new name of the object to be renamed. This name cannot contain any directory qualifiers and is in the same directory containing the existing object.

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

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

Top

Examples

Example 1: Renaming an Object Link

```
RNM  OBJ('DECEMBER-1994-MONTHLY-PAYROLL-FILE')
      NEWOBJ('JANUARY-1995-MONTHLY-PAYROLL-FILE')
```

This command renames a file named DECEMBER-1994-MONTHLY-PAYROLL-FILE to a file named JANUARY-1995-MONTHLY-PAYROLL-FILE.

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.
- CPFA09D**
Error occurred in program &1.
- CPFA0A1**
An input or output error occurred.
- CPFA0A3**
Path name resolution causes looping.
- CPFA0A6**
Number of links exceeds maximum allowed for the file system.
- CPFA0A7**
Path name too long.
- CPFA0A9**
Object not found. Object is &1.
- CPFA0AA**
Error occurred while attempting to obtain space.
- CPFA0AB**
Operation failed for object. Object is &1.
- CPFA0B1**
Requested operation not allowed. Access problem.
- CPFA0B2**
No objects satisfy request.
- CPFA0B4**
NEWOBJ parameter cannot start with a slash.
- CPFA0B5**
The NEWOBJ parameter cannot contain path.
- CPFA0C4**
Object not a file. Object is &1.

Top

Rename Directory Entry (RNMDIRE)

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

Parameters
Examples
Error messages

The Rename Directory Entry (RNMDIRE) command renames a local or remote user identifier (ID) and user address to a new user ID and user address. A rename operation is not allowed for generic (*ANY) user IDs or default directory entries (QSYS, QDFTOWN, QLPAUTO, QLPINSTL). This command renames all occurrences of the specified user ID and address in all IBM-supplied files.

It is recommended that this job be scheduled during low-use periods using the Submit Job (SBMJOB) command.

Only one rename operation can be run on the system at one time. If the rename is submitted to batch, the job waits for an active rename to complete.

Restriction: You must have administrator (*SECADM) or all object (*ALLOBJ) authority to rename the user ID and user address.

Top

Parameters

Keyword	Description	Choices	Notes
OLDUSRID	Old user identifier	<i>Element list</i>	Required, Key, Positional 1
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
NEWUSRID	New user identifier	Single values: *BACKOUT Other values: <i>Element list</i>	Required, Positional 2
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
FWDFRM	Forward from user identifier	Single values: *NONE, *OLDUSRID Other values: <i>Element list</i>	Optional
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
NETUSRID	Change network user identifier	<u>*SAME</u> , *NEWUSRID	Optional

Top

Old user identifier (OLDUSRID)

Specifies the user ID and address of the directory entry being renamed. Both elements must be specified. If lowercase characters are specified, the system stores them as uppercase characters. More information about specifying the user ID and address is in the SNA Distribution Services book, SC41-5410.

This is a required parameter.

Element 1: User ID

character-value

Specify the current user ID for the directory entry. A maximum of 8 characters can be specified. If this value is specified, an address must be specified on Element 2.

Element 2: Address

character-value

Specify the current address for the directory entry. A maximum of 8 characters can be specified. If this value is specified, a user ID must be specified on Element 1.

Top

New user identifier (NEWUSRID)

Specifies the user ID and address to which the old user ID and address is being renamed. Both elements must be specified but only one element needs to be different from the user ID and address specified on the OLDUSRID parameter.

The new user ID and address specified cannot be an existing user ID and address or exist as a forward-from value in the directory.

If the entry being renamed is in error from a previous rename request, you can continue with the rename operation or back out the changes and reset the files to the old user ID and address. To back out the changes, specify *BACKOUT on this parameter. To continue with the rename operation, do not change the value of this parameter (if the value is changed to another user ID and address, this is an error).

If lowercase characters are specified, the system stores them as uppercase characters.

Single values

***BACKOUT**

Back out of the rename directory entry operation. This value is only allowed on a directory entry that is in error as the result of a previous rename. This value sets the user ID and address in all IBM-supplied files changed by a previous rename request to the values specified on the OLDUSRID parameter.

Element 1: User ID

character-value

Specify the new user ID for the directory entry. A maximum of 8 characters can be specified. If this value is specified, an address must be specified on Element 2.

Element 2: Address

character-value

Specify the new address for the directory entry. A maximum of 8 characters can be specified. If this value is specified, a user ID must be specified on Element 1.

Note: Changing the address element does not change the system name of the directory entry. If you want distributions for the user forwarded to a system other than what is specified by the directory entry, you must change the system name for the directory entry using the Change Directory Entry (CHGDIRE) command.

Top

Forward from user identifier (FWDFRM)

Specifies whether distributions are automatically forwarded from the old user ID and address or a specified user ID and address. This value is valid only for local users.

Single values

*NONE

Distributions are not forwarded.

*OLDUSRID

All distributions are forwarded from the old user ID and address.

Element 1: User ID

character-value

Specify the user ID from which distributions are to be forwarded. A maximum of 8 characters can be specified. If this value is specified, an address must be specified on Element 2.

Element 2: Address

character-value

Specify the address from which distributions are to be forwarded. A maximum of 8 characters can be specified. If this value is specified, a user ID must be specified on Element 1.

Top

Change network user identifier (NETUSRID)

Specifies whether the current network user ID and address are renamed to the new user ID and address. The network user ID is used in shadowing to uniquely identify a user in the network. The default is the user ID and address. If you are using directory shadowing with the user ID and address as the unique value in the network, you can also change this value to the new user ID and address specified on the NEWUSRID parameter.

*SAME

The value does not change.

*NEWUSRID

The network user ID and address are changed to the new user ID and address.

Top

Examples

Example 1: Renaming a User ID

```
RNMDIRE  OLDUSRID(HURST PAYROLL)  NEWUSRID(HURST NEWYORK)
          FWDFRM(*OLDUSRID)
```

This command renames the current user ID HURST PAYROLL to the new user ID HURST NEWYORK. Distributions sent to the old user ID and address are forwarded.

Example 2: Renaming a User ID and Network User ID

```
RNMDIRE  OLDUSRID(HURST PAYROLL)  NEWUSRID(HURST NEWYORK)
          FWDFRM(*OLDUSRID)  NETUSRID(*NEWUSRID)
```

This command renames the current user ID HURST PAYROLL and the current network user ID to the new user ID HURST NEWYORK. Distributions sent to the old user ID and address are forwarded.

[Top](#)

Error messages

***ESCAPE Messages**

CPF897F

Rename failed for user ID and address &1 &2.

CPI90F1

Rename completed with errors. See previously listed messages.

[Top](#)

Rename Document Library Object (RNMDLO)

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

Parameters
Examples
Error messages

The Rename Document Library Object (RNMDLO) command changes the name of a document or folder. You will not be able to change the name if the document or folder is in use.

Restrictions:

- You must have all (*ALL) authority to the document or folder being renamed and must have change (*CHANGE) authority to the folder containing it.
- While using this command, you may encounter an error message indicating that internal objects are locked. Another user is using the document library functions which cannot run at the same time as the RNMDLO command; therefore, retry this command in a few minutes.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value	Required, Positional 1
NEWDLO	New document library object	Character value	Required, Positional 2
FLR	In folder	Character value, <u>*NONE</u>	Optional, Positional 3

Top

Document library object (DLO)

Specifies the name of the document or folder that is being renamed.

This is a required parameter.

Top

New document library object (NEWDLO)

Specifies the new name for the document or folder.

This is a required parameter.

Top

In folder (FLR)

Specifies the name of the folder that contains the document or folder being renamed.

*NONE

The folder being renamed is a first-level folder. *NONE cannot be specified if the object being renamed is a document.

name Specify the name of the folder that contains the document or folder being renamed.

Top

Examples

RNMDLO DLO(A) NEWDLO(B) FLR(FLR1)

This command changes the name of document or folder A located in folder FLR1, to B.

Top

Error messages

*ESCAPE Messages

CPF8A14

&2 of type &4 not renamed to &3 in folder &1.

Top

Rename Distribution List (RNMDSTL)

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

Parameters
Examples
Error messages

The Rename Distribution List (RNMDSTL) command renames the list identifier (ID) of an existing distribution list.

A **distribution list** is a list of entries from the distribution directory. The distribution list can include entries for local, remote, indirect, and independent work station users. It can also include remote distribution lists, but it cannot include local distribution lists. More information about distribution lists is in the SNA Distribution Services book, SC41-5410.

When renaming the distribution list, the new list ID must be unique to all local user IDs and other list IDs in the directory. If a new list ID is not unique, the list is not renamed and an error message is returned.

Restriction: You must have security administrator (*SECADM) authority to rename a distribution list that you do not own. No special authority is needed for you to rename a distribution list that you own.

Top

Parameters

Keyword	Description	Choices	Notes
LSTID	List identifier	<i>Element list</i>	Required, Positional 1
	Element 1: List ID	<i>Character value</i>	
	Element 2: List ID qualifier	<i>Character value</i>	
NEWLSTID	New list identifier	<i>Element list</i>	Required, Positional 2
	Element 1: List ID	<i>Character value</i>	
	Element 2: List ID qualifier	<i>Character value</i>	

Top

List identifier (LSTID)

Specifies the two-part list identifier of the distribution list to be renamed.

The possible list identifier value is:

list-ID

Specify the list identifier (ID) of the distribution list.

The possible list qualifier value is:

list-ID-qualifier

Specify the list ID qualifier of the distribution list.

Note: The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services book, SC41-5410.

This is a required parameter.

Top

New list identifier (NEWLSTID)

Specifies the new two-part list identifier of the distribution list.

The possible list identifier value is:

list-ID

Specify the list identifier (ID) of the distribution list.

The possible list qualifier value is:

list-ID-qualifier

Specify the list ID qualifier of the distribution list.

Note: The distribution list identifier has two parts, the ID and the qualifier, separated by at least one space. If lowercase characters are specified, the system changes them to uppercase.

The naming rules for the two-part list ID are identical to the rules for the user ID and address. A complete description of these rules is in the SNA Distribution Services book, SC41-5410.

This is a required parameter.

Top

Examples

```
RNMDSTL  LSTID(DEPTABC DLIST)  NEWLSTID(DEPTXYZ DLIST)
```

This command renames a distribution list that contains the members of Department ABC. The list ID is being changed to correspond to a new department name, XYZ. If the new list ID is unique, the distribution list is changed.

Top

Error messages

*ESCAPE Messages

CPF9A8B

Distribution list &1 &2 not renamed.

CPF9024

System cannot get correct record to finish operation.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

Top

Rename LAN Adapter (RNMLANADPI)

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

Parameters
Examples
Error messages

This command is used to change the name associated with the LAN adapter information in the adapter file.

Restriction: Adapter names must be unique.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
ADPTNAME	Adapter	<i>Name</i>	Required, Positional 1
NEWNAME	New name	<i>Name</i>	Required, Positional 2

[Top](#)

Adapter (ADPTNAME)

Specifies the name that is currently associated with the adapter.

This is a required parameter.

[Top](#)

New name (NEWNAME)

Specifies the new name associated with the adapter information. The new name has an object of type *NAME.

This is a required parameter.

[Top](#)

Examples

```
RNMLANADPI ADPTNAME(ACCTG3A) NEWNAME(BILLING2)
```

This command changes the name from ACCTG3A to BILLING2.

[Top](#)

Error messages

*ESCAPE Messages

CPF8B48

Adapter name - &29 or address - &30 already in the network adapter file

CPF8B68

Line description &23 not found.

CPF8B74

Request to display active adapters failed.

CPF8B75

No adapter entries in network adapter file.

CPF8B76

No functional addresses for adapter.

CPF8B83

Rename request for adapter &29 failed. Adapter name not found.

Top

Rename Member (RNMM)

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

Parameters
Examples
Error messages

The Rename Member (RNMM) command changes the name of a specified file member. The member cannot be renamed while it is in use; other users can read and change records of other members in the file that contains the member being renamed. A member that is opened in the same job cannot be renamed.

Restrictions:

- You must have object management (*OBJMGT) authority for the file that contains the member that is to be renamed, and have update (*UPDATE) and execute (*EXECUTE) authorities for the library in which the file is located.
- 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	Data base file	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Data base file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	<i>Name</i>	Required, Positional 2
NEWMBR	New member	<i>Name</i>	Required, Positional 3

Top

Data base file (FILE)

Specifies the database file (physical or logical) that contains the member to be renamed.

This is a required parameter.

Qualifier 1: Data base file

name Specify the name of the database 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 physical or logical file member that is to be renamed.

This is a required parameter.

name Specify the name of the file member.

New member (NEWMBR)

Specifies the new name to be given to the file member. The member remains in the same file. The new name must not be the same as the name specified on the **Member (MBR)** parameter. The new name must be unique in the file.

This is a required parameter.

name Specify the new name of the file member.

Examples

```
RNMM FILE(ELEMENT) MBR(LEAD) NEWMBR(GOLD)
```

This command renames member LEAD in file ELEMENT as GOLD. The library list (*LIBL) is used to find the file.

Error messages

*ESCAPE Messages

CPF3178

Member &3 not renamed to &4.

Rename Nickname (RNMNCK)

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

Parameters
Examples
Error messages

The Rename Nickname (RNMNCK) command is used to rename an existing nickname in the system distribution directory. The new nickname must be unique if it is a public nickname. The new nickname must be unique only for the owner if it is a private nickname.

A **nickname** is a short version of either a directory entry or a distribution list name. More information about nicknames is in the SNA Distribution Services book, SC41-5410.

Restrictions:

1. You must have security administrator (*SECADM) authority to rename a public nickname that you do not own. No special authority is needed for you to rename a public nickname that you own.
2. Only the owner can rename a private nickname. No special authority is needed.

Top

Parameters

Keyword	Description	Choices	Notes
NCK	Nickname	<i>Element list</i>	Required, Positional 1
	Element 1: Nickname	<i>Character value</i>	
	Element 2: Access	*PRIVATE , *PUBLIC	
NEWNCK	New nickname	<i>Character value</i>	Required, Positional 2

Top

Nickname (NCK)

Specifies the existing nickname to be renamed and the access of that nickname.

The possible nickname value is:

nickname

Specify the nickname you are renaming.

The possible nickname access values are:

***PRIVATE**

The private nickname that you own is being renamed.

***PUBLIC**

The public nickname is being renamed. Public nicknames can be renamed only by a user with security administrator (*SECADM) authority or by the owner.

This is a required parameter.

New nickname (NEWNCK)

Specifies the new nickname.

This is a required parameter.

Examples

```
RNMNCK NCK(SEC44A *PUBLIC) NEWNCK(SEC44C)
```

This command renames the public nickname SEC44A to SEC44C. If the new nickname is unique, the nickname is renamed.

Error messages

*ESCAPE Messages

CPF8AA1

Library QUSRSYS not completely installed.

CPF8360

Not enough storage for commitment control operation.

CPF9A80

Public nickname &1 in use.

CPF9A80

Public nickname &1 in use.

CPF9024

System cannot get correct record to finish operation.

CPF905C

Error occurred trying to find a translation table.

CPF9838

User profile storage limit exceeded.

Rename Object (RNMOBJ)

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

Parameters
Examples
Error messages

The Rename Object (RNMOBJ) command changes the name of an object in a library. The new name specified for the object must be unique in the library for the object type. If the object to be renamed is in use when this command is entered, the object is not renamed. If a library is on an active user's library list when the library is renamed, a Display Library List (DSPLIBL) command reflects the new name. Renaming a library can cause programming errors. Therefore, it is not recommended.

If the object being renamed is currently journaled, an entry is written to the journal recording the change. Use the Display Object Description (DSPOBJD) command to display journal information for the object.

Note: For additional information regarding journaling, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Restrictions:

1. You must have object management (*OBJMGT) authority for the object that is to be renamed and have update (*UPDATE) and execute (*EXECUTE) authorities for the library in which the object is located.
2. When a library contains one of the following object types, it cannot be renamed: *CRG, *DTADCT, *JRN, *JRNRCV, *SQLPKG, and *SQLUDT.
3. A PL/I program cannot be renamed after it has been created.
4. Configuration objects including controller descriptions, line descriptions, device descriptions, and network interface descriptions must be varied off in order to be renamed.
5. The following objects cannot be renamed:
 - The job's temporary library (QTEMP)
 - The following system libraries (where xxxxx is a 5-digit number):
 - QSYS or QSYSxxxxx
 - QSYS2 or QSYS2xxxxx
 - QRCL or QRCLxxxxx
 - QRECOVERY or QRCYxxxxx
 - SYSIBM or SYSIBxxxxx
 - SYSIBMADM
 - SYSPROC
 - SYSTOOLS
 - QQALIB
 - QSYSCGI
 - The system operator message queue (QSYSOPR)
 - All work station user message queues
 - The system log (QHST)
 - The configuration objects (QCTL and QCONSOLE)
 - The configuration lists (QAPPNRMT, QAPPNLCL, QASYNCLC, QRTLPASTR)
 - The Electronic Customer Support configuration objects (QESLINE, QESPAP, QESCTL, QTILINE, QTICTL, QTIDA, QTIDA2, QIADSP, QIAPRT, QQAHOST)
6. A user library cannot be renamed to any of the following (where xxxxx is a 5-digit number):

- QSYSxxxxx
 - QSYS or QSYS2xxxxx
 - QRCL or QRCLxxxxx
 - QRECOVERY or QRCYxxxxx
 - SYSIBM or SYSIBxxxxx
 - SYSIBMADM
 - SYSPROC
 - SYSTOOLS
 - QTEMP
7. When renaming objects of type *CSI, *GSS, *FNTRSC, *FORMDF, *OVL, *PAGDFN, and *PAGSEG, the new name for the object cannot exceed 8 characters in length.
 8. This command is conditionally threadsafe. The following restrictions apply:
 - In multithreaded jobs, this command is not threadsafe for distributed files and fails for distributed files that use relational databases of type *SNA. This command is also not threadsafe and fails for Distributed Data Management (DDM) files of type *SNA, when SYSTEM(*RMT) or SYSTEM(*FILETYPE) is specified.
 - Renaming objects of type *CFGL, *CNL, *CTLD, *DEVD, *LIND or *NWID will fail in multithreaded jobs.

NOTES:

1. References made to the following items may need to be updated by the user after a rename of a configuration object:
 - Connection lists
 - Work station entries
 - Communication entries
 - Display files
 - Printer files
 - Tape files
 - Diskette files
 - ICF files
 - User profiles
 - Job descriptions
 - CL programs
 - QPRTDEV system value
 - Display descriptions referencing it as an auxiliary printer
 - Communication side information (CSI) objects
 - Distributed data management files (APPC device name)
 - Integrated services digital network (ISDN) controller descriptions that refer to a renamed connection list (CNL)
 - ISDN line descriptions that refer to a renamed CNL
 - Other configuration objects. For example, lines, controllers, and other devices that refer to the renamed configuration objects
2. References made to the renamed object by the following items are automatically changed by the system after a rename operation. The reference changes reflect the changes made to the renamed configuration objects.
 - QCONSOLE system values
 - message queues associated with display devices

- System/36 environment device tables
- output queues associated with the old printer device
- local work station controllers associated with a twinaxial data link control (TDLC) line
- TDLC lines associated with the local or remote work station controller

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, *AUTL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *LIB, *LIND, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *NTBD, *NWID, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDL0D, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SRVPGM, *SSND, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
NEWOBJ	New object	<i>Name</i>	Required, Positional 3
ASPDEV	ASP device	<i>Name, *, *CURASPGRP, *SYSBAS</i>	Optional
SYSTEM	System	<i>*LCL, *RMT, *FILETYPE</i>	Optional, Positional 4

Top

Object (OBJ)

Specifies the object to be renamed. If the library (qualifier 2) is not specified, *LIBL is used. A library name can be specified for qualifier 2 to ensure that the correct object is renamed.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object to be renamed.

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.

Object type (OBJTYPE)

Specifies the object type of the object to be renamed.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

object-type

Specify the object type of the object to be renamed.

Top

New object (NEWOBJ)

Specifies the new name of the object to be renamed. The object remains in the same library.

This is a required parameter.

name Specify the new name of the object.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library containing the object to be renamed. If the library is in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct object is renamed. If a library to be renamed is in an auxiliary storage pool (ASP) device that is not currently part of the thread's library name space, specify the library as OBJ(QSYS/name), the object type as OBJTYPE(*LIB), and the ASP device as ASPDEV(name).

Note: ASPDEV(*) is the only valid value when *LIBL or *CURLIB is specified for the library (qualifier 2) of the **Object (OBJ)** parameter.

*
- The ASPs that are currently part of the thread's library name space will be searched to find the library. This includes the system ASP (ASP 1), all defined basic user ASPs (ASPs 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

*CURASGRP

If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will not be searched. If no ASP group is associated with the thread an error will be issued.

*SYSBAS

The system ASP (ASP 1) and all defined basic user ASPs (ASPs 2-32) will be searched to find the library. No primary or secondary ASPs will be searched, even if the thread has an ASP group.

name Specify the name of the primary or secondary ASP device to be searched to find the library. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Active' or 'Available'. The system ASP (1) and defined basic user 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

System (SYSTEM)

Specifies if the rename is to be done on the local system or on a remote system, if the file is a Distributed Data Management (DDM) file.

***LCL** The file on the local system is renamed.

***RMT** The remote file referred to by the source DDM file is renamed.

Note: If you want to rename a remote file, two DDM files must be used. The existing DDM file is specified for the **Object (OBJ)** parameter, and the new DDM file is specified for the **New object (NEWOBJ)** parameter. The new DDM file must be in the same library as the existing DDM file. When the remote rename occurs, it is the remote file name in the existing DDM file that is renamed to the new DDM file name.

***FILETYPE**

If the name in the OBJ parameter is a DDM file, the rename will be a remote rename. If the name in the OBJ parameter is not a DDM file, the rename will be a local rename.

Top

Examples

```
RNMOBJ OBJ(PAYROLL/FILEX) OBJTYPE(*FILE) NEWOBJ(MSTR)
```

The library named PAYROLL is searched for the file named FILEX. If the file is found, and you have object operational (*OBJOPR) authority for FILEX and update (*UPD) authority for the PAYROLL library, FILEX is renamed MSTR.

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.

CPF180B
Function &1 not allowed.

CPF2105
Object &1 in &2 type *&3 not found.

CPF2110
Library &1 not found.

CPF2111
Library &1 already exists.

CPF2112
Object &1 in &2 type *&3 already exists.

CPF2113
Cannot allocate library &1.

CPF2114
Cannot allocate object &1 in &2 type *&3.

CPF2132
Object &1 already exists in library &2.

CPF2136
Renaming library &1 failed.

CPF2139
Rename of library &1 failed.

CPF2140
Rename of library &1 previously failed.

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF2160
Object type *&1 not eligible for requested function.

CPF2164
Rename of library &2 not complete.

CPF2166
Library name &1 not valid.

CPF2173
Value for ASPDEV not valid with special value for library.

CPF2176
Library &1 damaged.

CPF218C
&1 not a primary or secondary ASP.

CPF2182
Not authorized to library &1.

CPF2189
Not authorized to object &1 in &2 type *&3.

CPF2190
Not able to do remote delete or rename request.

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.

CPF2691
Rename of &2 type *&5 did not complete.

CPF2692
Object &2 type *&5 must be varied off.

CPF2693
&2 type *&5 cannot be used for rename.

CPF2694
Object &2 type *&5 cannot be renamed.

CPF2696
Object &2 type *&5 not renamed.

CPF320B
Operation was not valid for database file &1.

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.

CPF323C
QRECOVERY library could not be allocated.

CPF323D
User does not have correct authority.

CPF324C
Concurrent authority holder operation prevents move, rename or restore.

CPF3245
Damage to file &1 member &6 prevents operation on file &3.

CPF326A
Operation not successful for file &1 in library &2.

CPF327F
Operation not successful for file &1 in library &2.

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.

CPF3375
Job queue &1 in &2 not renamed. Job queue in use.

CPF3376
Output queue &1 in &2 not renamed. Output queue in use.

CPF3467
Output queue &1 deleted and then created again.

CPF3469
Operation not allowed for output queue.

CPF5702
File either not DDM file or not found.

CPF7003
Entry not journaled to journal &1. Reason code &3.

CPF8D05
Library &1 already exists.

CPF88C4
Value &1 for new object is more than 8 characters.

CPF9801
Object &2 in library &3 not found.

CPF9803
Cannot allocate object &2 in library &3.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9809
Library &1 cannot be accessed.

CPF9814
Device &1 not found.

CPF9825
Not authorized to device &1.

CPF9830
Cannot assign library &1.

CPF9833
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPF9876
Protected library &2 cannot be modified.

Rename TCP/IP Host Table Entry (RNMTCPHTE)

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

Parameters
Examples
Error messages

The Rename TCP/IP Host Table Entry (RNMTCPHTE) command is used to rename a host table entry in the local host table. This is done by assigning a new internet address to an existing host table entry. This has the effect of associating the host names for the old internet address with the new internet address. Host names and the text description field are not altered with the RNMTCPHTE command.

The local host table is defined to allow 1 internet address, 65 host names and 1 text description field per entry.

The RNMTCPHTE command performs the rename of the host table entry by removing the existing host table entry and then adding a new host table entry at the end of the table that contains the new internet address.

See also the following host table commands:

- Add TCP/IP Host Table Entry (ADDTCPHTE) command adds a new entry in the local host table.
- Change TCP/IP Host Table Entry (CHGTCPHTE) command changes one or more host names or the text description field.
- Copy TCP/IP Host Table (CPYTCPHT) command copies the contents of the local host table to a physical file member.
- Merge TCP/IP Host Table (MRGTCPHT) command merges host names, internet addresses, and text comment entries from a physical file member into the local host table. A replace option is also provided that allows the entire local host table to be replaced by the physical file member.
- Remove TCP/IP Host Table Entry (RMVTCPHTE) command removes an entire entry from the local host table.

Restrictions:

- You must have input/output system configuration (*IOSYSCFG) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
INTNETADR	Internet address	<i>Character value</i>	Required, Positional 1
NEWINTNETA	New internet address	<i>Character value</i>	Required, Positional 2

Top

Internet address (INTNETADR)

Specifies the internet address associated with the host table entry that you want to rename. An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

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

An IPv4-mapped IPv6 address may be specified. An IPv4-mapped IPv6 address is specified in the form *::FFFF:nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255.

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

This is a required parameter.

character-value

Specify the current internet address.

Top

New internet address (NEWINTNETA)

Specifies the new internet address for the host table entry. An IPv4 internet address is specified in the form *nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255. An IPv4 internet address is not valid if it has a value of all binary ones or all binary zeros for the network identifier (ID) portion or the host ID portion of the address.

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

An IPv4-mapped IPv6 address may be specified. An IPv4-mapped IPv6 address is specified in the form *::FFFF:nnn.nnn.nnn.nnn*, where *nnn* is a decimal number ranging from 0 through 255.

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

This is a required parameter.

character-value

Specify the new internet address.

Top

Examples

Example 1: Renaming a Host Table Entry Containing an IPv4 Address

```
RNMTCPHTE INTNETADR('132.28.71.5') NEWINTNETA('142.48.81.6')
```

This command replaces the host table entry internet address of 132.28.71.5 with the internet address of 142.48.81.6. All host names and the text description field associated with the entry remain the same.

Example 2: Renaming a Host Table Entry Containing an IPv6 Address

```
RNMTCPHTE  INTNETADR('1234::5678')  NEWINTNETA('5678::1234')
```

This command replaces the host table entry internet address of 1234::5678 with the internet address of 5678::1234. All host names and the text description field associated with the entry remain the same.

Example 3: Renaming an Host Table Entry Containing an IPv4 Address

```
RNMTCPHTE  INTNETADR('132.28.71.5')  NEWINTNETA('5678::1234')
```

This command replaces the host table entry internet address of 132.28.71.5 with the internet address of 5678::1234. All host names and the text description field associated with the entry remain the same. It is also possible to rename a host table entry that contains an IPv6 address and specify an IPv4 address as the new internet address.

Top

Error messages

*ESCAPE Messages

TCP1901

Internet address &2 not valid.

TCP1902

Internet address &1 not valid.

TCP1904

Duplicate internet address &2 found in host table.

TCP1907

Internet address entry &1 does not exist.

TCP1908

Internet address &1 not valid.

TCP1910

LOOPBACK internet address &2 not valid.

TCP1929

Host table not available.

TCP8050

*IOSYSCFG authority required to use &1.

Top

Rollback (ROLLBACK)

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

Parameters
Examples
Error messages

The Rollback (ROLLBACK) command is used to restart the current transaction and reestablish the last commitment boundary as the current commitment boundary for the commitment definition associated with the program issuing the command.

When the ROLLBACK command is issued:

- Changes made to database files and other commitment resources under commitment control for the commitment definition since the last commitment boundary was established are rolled back. Updates, additions, or deletions made to the database file's data since that commitment boundary are rolled back or removed, and the original entries are put back in the files. Records that were added to the files remain as deleted records. The files are repositioned to the last commitment boundary. Changes made to other commitment resources are rolled back as well.
- All record locks held for files opened under commitment control for the commitment definition are released.
- Locks on object level commitment control resources, acquired when the resources were created or changed during the transaction, are released.

More information on the ROLLBACK command is in the Commitment control topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

ROLLBACK

This command reestablishes the last commitment boundary (the point at which a Commit (COMMIT) command or Rollback (ROLLBACK) command was last issued) for the commitment definition associated with the program issuing the command.

Top

Error messages

*ESCAPE Messages

CPF83D0

Commitment operation not allowed.

CPF835F

Commit or rollback operation failed.

CPF8350

Commitment definition not found.

CPF8359

Rollback operation failed.

CPF8367

Cannot perform commitment control operation.

STATUS Messages*CPF83E6**

Commitment control operation completed with resynchronization in progress.

Top

Start RPC Binder Daemon (RPCBIND)

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

Parameters
Examples
Error messages

The Start RPC Binder Daemon (RPCBIND) command starts 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:

- STRNFSSVR SERVER(*RPC)

If the user attempts to start this daemon and it is already running, it will not cause the command to fail. The command will issue diagnostic message CPDA1BA if the daemon is already running.

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

Keyword	Description	Choices	Notes
RTVRPCREG	Retrieve RPC registration	*NO, *YES	Optional, Positional 1

Top

Retrieve RPC registration (RTVRPCREG)

Specifies whether to retrieve previously recorded registration information from an internal file when the RPCBind daemon is started. If registration information is retrieved from the file, any services already registered with the RPCBind daemon do not have to re-register with the RPCBind daemon.

***NO** Do not retrieve registration information.

***YES** Retrieve registration information from an internal file.

Top

Examples

Example 1: Start RPC Binder Daemon

```
RPCBIND RTVRPCREG(*YES)
```

This command starts the RPC binder daemon job, and retrieves previously recorded registration information.

[Top](#)

Error messages

*ESCAPE Messages

CPFA1B8

*IOSYSCFG authority required to use &1.

[Top](#)

Convert RPC Source (RPCGEN)

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

Parameters
Examples
Error messages

The Convert RPC Source (RPCGEN) command generates C code from an input file written in the Remote Procedure Call (RPC) Language. The generated C code can be used to implement an RPC protocol.

This command is equivalent to running the **rpcgen** utility on a UNIX system.

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

- CVTRPCSRC

Restrictions:

- The user must have execute (*X) authority to each directory in the path for both the input and output files.
- The user must have read (*R) authority to the input file.
- The user must have write and execute (*WX) authority to the output file directory.

Top

Parameters

Keyword	Description	Choices	Notes
FROMFILE	From file	<i>Path name</i>	Required, Positional 1
OPTION	Option	*ALL, *XDR, *HDR, *CLTSTUB, *SVRSTUB, *CLTSAMP, *SVRSAMP, *NOSAMP	Optional
PROTOCOL	Protocol	Values (up to 2 repetitions): *NONE, *TCP, *UDP	Optional
TOFILE	To file	<i>Path name</i>	Optional

Top

From file (FROMFILE)

Specifies the path name of the input source file written in the Remote Procedure Call (RPC) Language. The input source file must be a file in the "root" (/) or QOpenSys file system.

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

Top

Option (OPTION)

Specifies the compile options.

***NOSAMP**

All file types except the sample files (*CLTSAMP and *SVRSAMP) are generated.

***ALL** All file types are generated.

***XDR** The input file is compiled into XDR (eXternal Data Representation) routines.

***HDR** The input file is compiled into C data-definitions (a header file).

***CLTSTUB**

The input file is compiled into client-side stub procedures.

***SVRSTUB**

The input file is compiled into server-side stub procedures. However, no "main" routine is generated.

***CLTSAMP**

Sample client code that uses remote procedure calls is generated. The file can be customized for the application.

***SVRSAMP**

Sample server code that uses remote procedure calls is generated. The file can be customized for the application.

Top

Protocol (PROTOCOL)

Compiles into server-side stub procedures for the transport that is specified. The specified value must be present in the /etc/netconfig file at the time the server application is run. This parameter is only valid when OPTION(*SVRSTUB) is specified. One or more of the following options may be specified:

***NONE**

Compile server-side stub procedures for all transports that are in the /etc/netconfig file.

***TCP** Compile server-side stub procedures for the TCP transport.

***UDP** Compile server-side stubs for the UDP transport.

Top

To file (TOFILE)

Specifies the path name of the output file. This option is only allowed if OPTION(*ALL) or OPTION(*NOSAMP) is not specified. When OPTION(*ALL) or OPTION(*NOSAMP) is specified, or if the TOFILE parameter is not specified when using another option, the **From file (FROMFILE)** parameter is used to generate the TOFILE name as follows, where filename is the name of the input file name from the FROMFILE parameter.

- filename.h for a header file
- filename_xdr.c for an XDR file
- filename_clnt.c for client-side stubs
- filename_svc.c for server-side stubs
- filename_client.c for client-side sample files
- filename_server.c for server-side sample files

The output file or files for sample code must not exist; if any of the sample output files exist, the command will fail. Other output files will be overwritten if they exist.

'to-file-path name'

Specify a path name to be used to generate the TOFILE name or names.

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

Top

Examples

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

Example 1: Convert RPC Source to Default Files

```
CVTRPCSRC FROMFILE('/myrpc') OPTION(*ALL)
```

This converts the RPC language file *'/myrpc'* into all four file types, *XDR, *HDR, *CLTSTUB and *SVRSTUB. The default PROTOCOL(*TCP) is used to generate the server-side stub programs. The files are placed into the following file names:

- myrpc.h for a header file
- myrpc_xdr.c for an XDR file
- myrpc_clnt.c for client-side stubs
- myrpc_svc.c for server-side stubs

Example 2: Convert RPC Source to Client Stubs Only

```
CVTRPCSRC FROMFILE('/myrpc2') OPTION(*CLTSTUB)
          TOFILE('/myclnt.c')
```

This converts the RPC language file *'/myrpc2'* into client-side stub procedures. The results are placed into the file *'/myclnt.c'* as specified.

Top

Error messages

*ESCAPE Messages

CPFB416

CVTRPCSRC or RPCGEN command failed.

Top

Replace Document (RPLDOC)

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

Parameters
 Examples
 Error messages

The Replace Document (RPLDOC) command is used to replace the document content and the interchange document profile (IDP) of a document that exists in the document library. This command places the new version of the document back into the document library. The document must be checked out before the data can be replaced.

Restrictions:

- The document must be checked out with the Retrieve Document (RTVDOC) command by the user specified on the USRID parameter before using this command.
- The user must have at least change (*CHANGE) authority for the document, all object (*ALLOBJ) special authority, or be working on behalf of a user who is authorized for the document.
- Authority to work on behalf of another user is granted with the Grant User Permission (GRTUSRPMN) command.

Top

Parameters

Keyword	Description	Choices	Notes
TODOC	Document	Character value, *DOCID	Required, Positional 1
TOFLR	In folder	Character value, *NONE	Optional
DOCFILE	Document file	Qualified object name	Required, Positional 2
	Qualifier 1: Document file	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
DOCMBR	Document member	Name, *FIRST	Optional
DOCPART	Document part	*DOC, *IDP, *BOTH	Optional
USRID	User identifier	Single values: *CURRENT Other values: Element list	Optional
	Element 1: User ID	Character value	
	Element 2: Address	Character value	
DOCID	Document identifier	Character value, *NONE	Optional
DOCTYPE	Document type	2-65535, *DFT, *FFT, *RFT	Optional
DOCCHRID	Document character identifier	Single values: *SYSVAL, *DEVVD Other values: Element list	Optional
	Element 1: Graphic character set	Integer	
	Element 2: Code page	Integer	
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEVVD Other values: Element list	Optional
	Element 1: Graphic character set	Integer	
	Element 2: Code page	Integer	

Document (TODOC)

Specifies the name of the document where the data is being placed, or that the **Document identifier (DOCID)** parameter is used to specify the library-assigned document name where the data is being placed.

name Specify the user-assigned name of the document to be replaced.

*DOCID

The document being replaced is identified by the library-assigned document name specified on the DOCID parameter.

In folder (TOFLR)

Specifies the name of the folder that contains the document being replaced.

*NONE

No folder is specified when the document is identified on the **Document identifier (DOCID)** parameter.

name Specify the name of the folder that contains the document being replaced. A folder name can consist of a series of folder names if the document being replaced is located in a folder that is contained in another folder. A maximum of 63 characters can be specified.

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.

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

Document part (DOCPART)

Specifies the part of the document to be replaced.

Note: If you specify *BOTH and the replacement of one fails, neither is replaced and the document remains checked out.

***DOC** Replace the document content only. If no document content records exist in the specified file, the document is replaced but a message is returned to alert the user that no document content was replaced.

***IDP** Replace only the interchange document profile (IDP) of the document. If this value is specified, IDP records (record code 500) must exist in the specified file or the command fails.

***BOTH** Replace both the document content and the IDP of the document. If this value is specified, IDP records (record code 500) must exist in the specified file or the command fails.

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 identifier (DOCID)

Specifies the library-assigned name of the document. This is the name assigned to the document by the system when it was created. Documents filed outside the local system have only library-assigned document names. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format:

YYYYMMDDHHMNSSHSSNSNSNSN

where:

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of a second
SNSNSNSN = system name

***NONE**

No library-assigned document name is required when the document is identified on the **Document (DOC)** parameter.

name Specify the library-assigned name of the document being sent.

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.

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.

Single values

***SYSVAL**

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEV D**

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

The value specified on the **User identifier (USRID)** parameter is translated to character set and code page '930 500'.

Single values

*SYSVAL

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

*DEV D

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Examples

```
RPLDOC  TODOC(*DOCID)  DOCFILE(*LIBL/MYFILE)  DOCPART(*BOTH)
        DOCID('1987060710102053SYSTEM1')  DOCTYPE(*FFT)
```

This command replaces the document data and IDP with data in the file MYFILE. The data is placed in the document identified by the document identifier '1987060710102053SYSTEM1'. The document type is changed to Final Form Text.

Top

Error messages

*ESCAPE Messages

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF906A

Document not replaced in library.

CPF9096

Cannot use CMDCHRID(*DEVVD), DOCCHRID(*DEVVD) in batch job.

CPF9847

Error occurred while closing file &1 in library &2.

Top

Request Order Assistance (RQSORDAST)

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

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

The Request Order Assistance (RQSORDAST) command sends a request to IBM for order assistance. You can request assistance in ordering services and products including:

- Software upgrades
- Hardware upgrades
- System i information
- Service offerings
- General help (for example, network planning)

When the RQSORDAST command is successfully processed, a file containing the order information is created and sent along with the order assistance request. This file contains:

- hardware configuration information (vital product data (VPD) and topology data)
- software configuration information (installed IBM program products)

The order information file is sent with all requests.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. You must have *ALLOBJ authority or be signed on as QSYSOPR or QSRV to use the command.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

RQSORDAST

This command displays the Request Order Assistance entry panel.

[Top](#)

Error messages

None

[Top](#)

Reroute Job (RRTJOB)

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

Parameters
Examples
Error messages

The Reroute Job (RRTJOB) command starts a new routing step for a job in the current subsystem. The rerouting allows changes in the routing data for the job, and calls a different program used with the new routing step. A job can use rerouting in order to run under a different class or in a different storage pool. When this command is used, any objects allocated in the previous routing step are deallocated and any open files are closed. If the objects or files are needed in the new routing step, they must be allocated or opened again.

Notes:

1. Running of this command in a batch job causes loss of spooled inline files because they cannot be accessed in the new routing step.
2. If the RRTJOB command is run while the system is ending (by running of a End Subsystem (ENDSBS) command, End System (ENDSYS) command, or the Power Down System (PWRDWN SYS) command), a new routing step is not started and the job is ended.
3. If the RRTJOB command is issued in a CL program, all subsequent commands in the CL program are bypassed.

Restrictions:

1. The job must not be a group job, prestart job, batch immediate job, or a batch job in restricted state.

Top

Parameters

Keyword	Description	Choices	Notes
RTGDTA	Routing data	Character value, <u>QCMDI</u> , *RQSDTA	Optional, Positional 1
RQSDTA	Request data or command	Character value, * <u>NONE</u> , *RTGDTA	Optional, Positional 2

Top

Routing data (RTGDTA)

Specifies the routing data used to start the next routing step in the job. The routing data is used to determine the routing entry that identifies the program that the job runs.

QCMDI

This routing data matches a routing entry in the IBM-supplied subsystem description, QINTER, which indicates a routing step that is processed by the IBM-supplied control language processor, QCMD, in the QSYS library.

*RQSDTA

The first 80 characters of the request data specified in the **Request data or command (RQSDTA)** parameter of this command is also used as the routing data for the next routing step.

character-value

Specify the character string used as the routing data for starting the next routing step. A maximum of 80 characters can be entered, enclosed in apostrophes if necessary.

Top

Request data or command (RQSDTA)

Specifies the request data that is placed as the last entry in this job's message queue. The request data can be a CL command to be run or a string of characters used by another program.

*NONE

No request data is placed in the job's message queue.

*RTGDTA

The routing data specified in the **Routing data (RTGDTA)** parameter is also placed at the end of the job's message queue.

character-value

Specify the character string placed at the end of the job's message queue for use by the new routing step or some subsequent routing step in the job. A maximum of 256 characters can be entered, enclosed in apostrophes if necessary.

Top

Examples

```
RRTJOB RTGDTA(INQUIRY)
```

This command reroutes the job in which the command is issued by starting a new routing step with the routing data INQUIRY. The job remains in the same subsystem.

Top

Error messages

*ESCAPE Messages

CPF1315

Command &1 not allowed in this environment.

Top

Resume Breakpoint (RSMBKP)

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

Parameters
Examples
Error messages

The Resume Breakpoint (RSMBKP) command causes a program to continue processing after it has been stopped at a breakpoint. The program that continues is the one that most recently stopped at a breakpoint. When more than one program in the job is stopped at a breakpoint, the End Request (ENDRQS) command can be used to return to the command entry display for a previous program call that is also stopped at a breakpoint.

If you are servicing another job and it has not ended, this command resumes that job from the breakpoint. This command also returns the servicing job to the point immediately before the breakpoint display was shown.

Restrictions:

- This command is valid only in debug mode and only when the program is stopped at a user-defined breakpoint.
- This command is not valid at a breakpoint caused by an unmonitored message.

There are no parameters for this command.

[Top](#)

Parameters

None

[Top](#)

Examples

RSMBKP

Assuming that the program having control is stopped at a breakpoint, this command causes the program to continue processing, starting from the breakpoint location.

[Top](#)

Error messages

*ESCAPE Messages

CPF1999

Errors occurred on command.

[Top](#)

Resume Controller Recovery (RSMCTRLRCY)

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

Parameters
Examples
Error messages

The Resume Controller Recovery (RSMCTRLRCY) command resumes error recovery procedures for a specific controller. Error recovery procedures can be ended by using the End Controller Recovery (ENDCTRLRCY) command or by responding to a failure-related inquiry message with a cancel option.

The Resume Controller Recovery (RSMCTRLRCY) command allows you to resume automatic error recovery procedures after they have been stopped, and to reactivate a controller (and jobs using that controller) after it has been canceled. (if the C response was entered to the inquiry message associated with a controller failure). When the controller is canceled with the C response, all jobs are ended; once the controller is repaired and the RSMCTRLRCY command is entered, jobs are allowed to start using the controller again.

Restriction: To use this command, the user must have object operational authority for the controller.

Top

Parameters

Keyword	Description	Choices	Notes
CTL	Controller	<i>Name</i>	Required, Positional 1

Top

Controller (CTL)

Specifies the controller for which error recovery procedures are to resume. Valid types of controllers are:

CTL Value

Controller

5251 Display station

***PU2** Physical unit (type 2); SDLCs for basic BSC and RJE

***BSC** BSC device (basic BSC and RJE)

***BSCT**

BSC device (Multipoint tributary and 3270 Device Emulation)

***APPC**

Advanced program-to-program communications

***WSC** Local work station

***WSCE**

Local work station (extended)

This is a required parameter.

Examples

RSMCTLR CY CTL(TROLL3)

This command resumes error recovery procedures for the controller TROLL3.

Error messages

*ESCAPE Messages

CPF2703

Controller description &1 not found.

CPF5924

Controller &1 does not allow automatic error recovery.

CPF5928

Controller &1 not varied on.

CPF5929

Controller &1 assigned to another job.

CPF5930

Recovery not resumed for controller &1.

CPF5931

Recovery not resumed for controller &1.

CPF5935

Error occurred during command processing.

CPF5936

Not authorized to controller &1.

Resume Device Recovery (RSMDEVRCY)

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

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

The Resume Device Recovery (RSMDEVRCY) command resumes error recovery procedures for a specific device. Error recovery procedures are ended by the End Device Recovery (ENDDEVRCY) command or by responding to a failure-related inquiry message with a cancel option.

The RSMDEVRCY command allows you to resume automatic error recovery procedures after they have been stopped, and to reactivate a device after it has been canceled (if you entered the C response to the inquiry message associated with a device failure). When the device is canceled with the C response, all jobs are ended; once the device is repaired and the RSMDEVRCY command is entered, jobs are allowed to start using the device again.

Restriction: To use this command, you must have object operational authority for the device.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
DEV	Device	<i>Name</i>	Required, Positional 1

[Top](#)

Device (DEV)

Specifies the device for which error recovery procedures are to resume. Valid types of devices are:

DEV Type

Device

5219 Printer (work station)

5224 Printer (work station)

5225 Printer (work station)

5251 Display station

5252 Dual display station

5256 Printer (work station)

5291 Display station

5292 Display station

*PLU1 Physical unit (type 1)

*BSC BSC device

*BSCT

BSC multipoint tributary

*APPC

Advanced program-to-program communications

This is a required parameter.

Top

Examples

```
RSMDEVRCY  DEV(WSPR03)
```

This command resumes error recovery procedures for the device WSPR03 to resume.

Top

Error messages

*ESCAPE Messages

CPF5923

Device &1 does not allow automatic error recovery.

CPF5925

Device &1 not varied on.

CPF5926

Recovery not resumed for device &1.

CPF5927

Recovery not resumed for device &1.

CPF5935

Error occurred during command processing.

CPF9814

Device &1 not found.

CPF9825

Not authorized to device &1.

Top

Resume Line Recovery (RSMLINRCY)

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

Parameters
Examples
Error messages

The Resume Line Recovery (RSMLINRCY) command resumes error recovery procedures for a specific line. Error recovery procedures are ended by the End Line Recovery (ENDLINRCY) command or by responding to a failure-related inquiry message with a cancel option.

The Resume Line Recovery (RSMLINRCY) command allows you to resume automatic error recovery procedures after they stop, and to reactivate a line (and jobs using that line) after it is canceled (if the C response was entered to the inquiry message associated with a line failure). When the line is canceled with the C response, all jobs are ended; once the line is repaired and the RSMLINRCY command is entered, jobs are allowed to start using the line again.

Restriction: To use this command, the user must have object operational authority 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 started again.

This is a required parameter.

Top

Examples

```
RSMLINRCY LINE(NYC2)
```

This command resumes error recovery procedures for the line NYC2 to resume.

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.

CPF5934

Recovery not resumed for line &1.

CPF5935

Error occurred during command processing.

[Top](#)

Resume NWI Recovery (RSMNWIRCY)

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

Parameters
Examples
Error messages

The Resume Network Interface Recovery (RSMNWIRCY) command resumes error recovery procedures for a network interface description. Error recovery procedures are ended by the End Network Interface Recovery (ENDNWIRCY) command or by responding to a failure-related inquiry message with a cancel option. The Resume Network Interface Recovery (RSMNWIRCY) command allows you to resume automatic error recovery procedures after they stop, and to reactivate a network interface description (and jobs using that description) after it is canceled.

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 automatic error recovery is restarted.

This is a required parameter.

Top

Examples

```
RSMNWIRCY  NWID(ISDNNET)
```

This command resumes error recovery procedures for the network interface description 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.

CPF593D

Recovery not resumed for network interface &1.

[Top](#)

Restore Object (RST)

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

Parameters
 Examples
 Error messages

The Restore (RST) command restores a copy of one or more objects that can be used in the integrated file system.

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

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- For detailed restrictions on using this command to restore objects by using name patterns in the root directory, to restore objects in libraries, or to restore document library objects, see the Recovering your system book, SC41-5304.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Values (up to 4 repetitions): <i>Path name</i>	Required, Positional 1
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: Name	<i>Path name, _</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
	Element 3: New object name	<i>Path name, *SAME</i>	
PATTERN	Name pattern	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value, _</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
SUBTREE	Directory subtree	<u>*ALL</u> , *DIR, *NONE, *OBJ, *STG	Optional
OUTPUT	Output	<i>Path name, *NONE, *PRINT</i>	Optional
CRTPRNDIR	Create parent directories	<u>*NO</u> , *YES	Optional
PRNDIROWN	Parent directory owner	<i>Simple name, *PARENT</i>	Optional
RBDMFS	Rebuild mounted file system	<u>*NONE</u> , *UDFS	Optional
VOL	Volume identifier	Single values: <u>*MOUNTED</u> Other values (up to 75 repetitions): <i>Character value</i>	Optional
LABEL	Label	<i>Character value, *SEARCH</i>	Optional
SEQNBR	Sequence number	1-16777215, <u>*SEARCH</u>	Optional
ENDOPT	End of media option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
OPTFILE	Optical file	<i>Path name, _</i>	Optional
INFYTYPE	Type of output information	<u>*ALL</u> , *ERR, *SUMMARY	Optional
SYSTEM	System	*ALL, <u>*LCL</u> , *RMT	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional

Keyword	Description	Choices	Notes
OPTION	Option	<u>*ALL</u> , *NEW, *OLD	Optional
ALWBJDIF	Allow object differences	Single values: <u>*NONE</u> , *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: <u>*SYSVAL</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	
OBJID	Object ID	<u>*SAVED</u> , *SYS	Optional
PVTAUT	Private authorities	<u>*NO</u> , *YES	Optional

Top

Device (DEV)

Specifies the device from which the objects are restored.

For more information on specifying device path names, refer to "Specifying the device name" in the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

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

'save-file-path-name'

Specify the path name of the save file used to restore the objects.

'optical-device-path-name'

Specify the path name of the optical device used to restore the objects.

'tape-media-library-device-path-name'

Specify the path name of the tape media library used to restore the objects.

'tape-device-path-name'

Specify the path name of the tape device used to restore the objects. A maximum of four tape devices can be specified. If a virtual tape device is used, it must be the only device specified.

'media-definition-path-name'

Specify the path name of the media definition (*MEDDFN) object that identifies the devices and media used to restore the data.

For information about creating a media definition, see the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

For information about using a media definition, see the Recovering your system book, SC41-5304 and the Back up your server topic in the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Objects (OBJ)

Specifies the path name of the object to restore. You can specify a pattern for this path name. A maximum of 300 path names can be specified.

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

Additional information about object name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 1: Name

Specifies the objects saved on the media. Directory abbreviations (for example, the current directory) are expanded with their current values, not the values they had at the time of the save operation.

'*' The objects in the current directory are restored.

path-name

Specify an object path name or a pattern that can match many names. If *SAME is specified for the third element of this parameter, each component in the path name must exist with the exception of the last component. The object name in the last component is restored as new if it doesn't exist.

Element 2: Include or omit

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 determines whether the subtrees are included or omitted.

*INCLUDE

The objects that match the object name pattern are to be restored, unless overridden by an *OMIT specification.

*OMIT

The objects that match the object name pattern are not to be restored. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Element 3: New object name

Specifies the new path name of the object.

*SAME

The objects are to be restored with the same names they had when they were saved.

path-name

Specify the path name with which to restore the object. If a pattern is specified in the first element, the new path name must be the directory into which to restore any objects that match the pattern. If an object name is specified in element 1, each component in the new path name must exist with the exception of the last component. If the object described in the last component doesn't exist, it will be restored as new.

Name pattern (PATTERN)

Specifies a pattern to be used to include or omit objects. A maximum of 300 patterns can be specified.

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

Element 1: Pattern

'* All objects which qualify for the operation are included or omitted.

character-value

Specify an object name or a pattern that can match many names.

Element 2: Include or omit

Specifies whether names that match the pattern should be included or omitted from the operation.

Note: The SUBTREE parameter determines whether the subtrees are included or omitted.

*INCLUDE

Only objects which are included by the OBJ parameter and match the PATTERN parameter are included in the restore, unless overridden by an *OMIT specification.

*OMIT

All objects which are included by the OBJ parameter are included in the restore except those objects which match the PATTERN parameter. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the restore operation.

*ALL The entire subtree of each directory that matches the object name pattern is processed. The subtree includes all subdirectories and the objects within those subdirectories.

*DIR The objects in the first level of each directory that matches the object name pattern are processed. The subdirectories of each matching directory are included, but the objects in the subdirectories are not included.

*NONE

No subtrees are included in the restore operation. If a directory matches the object name pattern specified, the objects in the directory are included. If the directory has subdirectories, neither the subdirectories nor the objects in the subdirectories are included.

*OBJ Only the objects that match the object name pattern will be processed. If the object name pattern specifies a directory, objects in the directory are not included.

*STG The objects that match the object name pattern are processed along with the storage for related objects. Objects can only be restored using this value if they were saved with SUBTREE(*STG).

Output (OUTPUT)

Specifies whether a list of information about the restored objects is created. The information can be directed to a spooled file, a stream file, or a user space.

A stream file or user space is specified as a path name.

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

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

*NONE

No output is created.

*PRINT

Output information about the restore will be printed.

'stream-file-path-name'

Specify the path name of the existing stream file to which the output of the command is directed.

'user-space-path-name'

Specify the path name of the existing user space to which the output of the command is directed.

Top

Create parent directories (CRTPRNDIR)

Specifies whether parent directories of objects being restored should be created if they do not exist. For example, if object '/a/b/c/file1' is being restored then directories '/a', '/a/b' and '/a/b/c' must exist. This parameter only applies to "root" (/), QOpenSys and user-defined file systems, and will be ignored for all other file systems.

*NO Parent directories will not be created if they do not exist. Diagnostic message CPD375B will be sent and the object will not be restored.

*YES The restore will create parent directories if they do not exist. The directories created by the restore will have *EXCLUDE public authority and will be owned by the user profile specified for the **Parent directory owner (PRNDIROWN)** parameter. The other parent directory attributes will be set using the shipped default values for the **Create Directory (CRTDIR)** command parameters.

Top

Parent directory owner (PRNDIROWN)

Specifies the name of an existing user profile that will own parent directories created by the restore. This parameter only applies to "root" (/), QOpenSys and user-defined file systems, and will be ignored for all other file systems.

Note: If a value is specified for this parameter, *YES must be specified for the **Create parent directories (CRTPRNDIR)** parameter.

*PARENT

The owner of a parent directory being created by the restore will be the same as the owner of the directory it is being created into. For example, if object '/a/b/c/file1' is being restored and

directory `/'a'` exists but the `/'b'` and `/'b/c'` directories do not exist, the `/'b'` and `/'b/c'` directories are created with the same owner as the `/'a'` directory.

name Specify the name of a user profile to be the owner of any parent directories that are created by the restore.

Top

Rebuild mounted file system (RBDMFS)

Specifies which mounted file systems should be rebuilt during the restore.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify a value other than *NONE for this parameter.

*NONE

Mounted file systems will not be rebuilt during the restore. Objects that were saved from a mounted file system will be restored to the file system that contains the directory being restored into.

*UDFS

Mounted user-defined file systems will be rebuilt during the restore. A user-defined file system will be created if it does not exist and then will be mounted over the same directory it was mounted over during the save.

If a user-defined file system is created during the restore, the attributes will be set based on the saved user-defined file system including any user-defined file system specific attributes such as 'Case sensitivity' or 'Default file format'. If the user-defined file system exists before the restore, no attributes will be changed.

If there is an error creating or mounting a user-defined file system, none of the objects that were saved from the mounted user-defined file system will be restored.

Note: If it does not exist before the restore, the directory that is being mounted over will be created with attributes and authorities copied from the directory being restored into. This could cause problems when the user-defined file system is unmounted and then remounted, if the authorities are not sufficient to allow the mount to occur.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Note: The version of the object that is restored is the first version found in the specified location, unless a specific version is identified by the values on the SAVDATE and SAVTIME parameters.

Single values

*MOUNTED

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Label (LABEL)

Specifies the file identifier of the media to be used for the restore operation.

*SEARCH

The file label for which to search is determined by the system.

character-value

Specify the identifier (up to 17 characters) of the tape file to be used for the restore operation.

Top

Sequence number (SEQNBR)

Specifies the tape file sequence number to be used.

*SEARCH

The tape volume is searched for the next file that contains any of the specified objects.

1-16777215

Specify the sequence number of the file.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

*
_ The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

*'optical-directory-path-name'**

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Type of output information (INFTYPE)

Specifies the type of information which is directed to the spooled file, stream file, or user space.

***ALL** The file will contain information about the command, an entry for each directory, an entry for each object that was successfully restored, and an entry for each object that was not successfully restored.

***ERR** The file will contain information about the command, an entry for each directory, and an entry for each object that was not successfully restored.

***SUMMARY**

The file will contain information about the command, and an entry for each directory.

Top

System (SYSTEM)

Specifies whether to process objects that exist on the local system or remote systems.

***LCL** Only local objects are processed.

***RMT** Only remote objects are processed.

***ALL** Both local and remote objects are processed.

Top

Date when saved (SAVDATE)

Specifies the date the objects were saved. If the most recently saved version is not the one being restored, or if multiple saved versions reside on the media, specify the date that identifies which version of the objects to restore.

The date must be specified in the job date format. If the separators that are specified by the system value QDATSEP are used, the value must be enclosed in apostrophes.

Note: This parameter is valid only if a volume identifier or the value *MOUNTED is specified for the VOL parameter, or if *SAVF is specified for the DEV parameter. If this parameter is valid and is not specified, the restored version of the objects is the first version found.

date Specify the date the objects to be restored were saved.

Top

Time when saved (SAVTIME)

Specifies the time the objects were saved.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

If a volume identifier or the value *MOUNTED is specified for the VOL parameter, and the SAVTIME parameter is not specified, the version of the objects to be restored is the first version found on the volume.

Notes:

1. This parameter is valid only if the SAVDATE parameter is specified.
2. This parameter is ignored when the SEQNBR parameter is specified.

time Specify the time the objects to be restored were saved.

Top

Option (OPTION)

Specifies whether to restore objects that already exist on the system or objects that do not already exist on the system.

***ALL** All of the specified objects are restored, whether they already exist on the system or not.

***NEW** Objects are restored only if they do not already exist on the system.

***OLD** Objects are restored only if they already exist on the system.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.

Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

***NONE**

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

Other values (up to 3 repetitions)

***AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

- ***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Object ID (OBJID)

This parameter has been disabled and is no longer valid.

Top

Private authorities (PVTAUT)

Specifies whether to restore private authorities with the objects that are restored.

***NO** No private authorities are restored.

***YES** Private authorities are restored with the objects. Objects will be restored only from save operations that specified that private authorities should be saved with the objects.

Note: You must have all object (*ALLOBJ) special authority to specify this value.

Top

Examples

Example 1: Restoring All Data Not in Libraries or Document Library Objects

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT)
```

This command restores all objects that are not in libraries and are not document library objects.

Example 2: Restoring a Library

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')  OBJ('/QSYS.LIB/A.LIB')
```

This command restores the library A from the tape device named TAP01.

Example 3: Restoring All Files in MYLIB

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/QSYS.LIB/MYLIB.LIB/*.FILE')
```

This command restores all files in the library MYLIB from the tape device named TAP01.

Example 4: Restoring All Objects in the Current Directory

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
```

This command uses the default value on the OBJ parameter to restore all objects in the current directory and its subdirectories.

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')  OBJ('*')  SUBTREE(*NONE)
```

This command restores all objects in the current directory but not in subdirectories.

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')  OBJ('.')  SUBTREE(*DIR)
```

This command restores the current directory and all of the objects in the current directory. It does not restore objects in the subdirectories of the current directory.

Example 5: Omitting Objects

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('*') ('**.*BACKUP' *OMIT) ('**.*TEMP' *OMIT)
```

This command restores all objects in the current directory except those with extensions of .BACKUP and .TEMP (the entire subtrees of directories with these extensions are omitted).

Example 6: Renaming or Moving Objects

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
     OBJ(('MYDIR/X.PGM' *INCLUDE 'YOURDIR/Y.PGM'))
```

This command restores the program X from the directory MYDIR as the program Y in the directory YOURDIR.

```
RST  DEV('/QSYS.LIB/TAP01.DEVD')
     OBJ(('MYDIR/*.PGM' *INCLUDE 'YOURDIR')) SUBTREE(*OBJ)
```

This command restores all programs in the directory MYDIR to the directory YOURDIR.

Example 7: Restoring From a Save File

```
RST  DEV('/QSYS.LIB/MYLIB.LIB/MYSAVF.FILE') OBJ(MYDIR)
```

This command restores the directory MYDIR from a save file named MYSAVF in a library named MYLIB.

Example 8: Using Symbolic Links

Assume the current directory contains the following symbolic links.

- DevLink = /QSYS.LIB/TAP01.DEVD
- DirLink = /SomeDirectory
- FileLink = /SomeDirectory/SomeFile

Symbolic links can be used to specify the device and output file. When symbolic links are restored, only the names of the associated objects are restored, not the content of the associated objects. A symbolic link to a directory can be used to restore objects in the directory. Additional information about symbolic link is in the **Integrated file system** topic in the **File systems and management** category of the Information Center. To restore the names associated with DirLink and FileLink from device TAP01:

```
RST  DEV('DevLink') OBJ(('DirLink') ('FileLink'))
```

To restore the objects in SomeDirectory from device TAP01:

```
RST  DEV('DevLink') OBJ(('DirLink/*'))
```

Top

Error messages

*ESCAPE Messages

CPFA0DB

Object not a QSYS.LIB object. Object is &1.

CPFA0DC
Object not a QDLS object. Object is &1.

CPF370C
Not authorized to ALWOBJDIF parameter.

CPF3707
Save file &1 in &2 contains no data.

CPF3727
Duplicate device &1 specified on device name list.

CPF3738
Device &1 used for save or restore is damaged.

CPF3743
File cannot be restored, displayed, or listed.

CPF3768
Device &1 not valid for command.

CPF3782
File &1 in &2 not a save file.

CPF3794
Save or restore operation ended unsuccessfully.

CPF380D
Save or restore of entire system completed unsuccessfully.

CPF3805
Objects from save file &1 in &2 not restored.

CPF381E
Not authorized to ALWOBJDIF parameter.

CPF3812
Save file &1 in &2 in use.

CPF382A
Specified parameter not valid for QDLS file system.

CPF382B
Parameters not valid with multiple file systems.

CPF382C
OBJ parameter value not valid for QSYS file system.

CPF382D
Specified parameter not valid for QSYS file system.

CPF382F
OBJ parameter value not valid for QDLS file system.

CPF3823
No objects saved or restored.

CPF3826
*INCLUDE object required on OBJ parameter.

CPF3828
Error occurred while attempting to use &1.

CPF383A
Save or restore ended unsuccessfully.

CPF383B
End of file &1.

CPF383C
Storage limit exceeded for user profile &1.

CPF383D
Cannot use &1.

CPF383E
&1 objects restored. &2 objects not restored.

CPF3833
Specified value on DEV parameter not valid.

CPF3834
Too many values specified on the DEV parameter.

CPF3835
Tape devices do not support same densities.

CPF3839
&1 objects restored. &2 not restored.

CPF384A
Volume identifier &1 not valid.

CPF384B
Optical file specified not valid.

CPF384C
Error occurred during CCSID conversion.

CPF384F
&2 &1 not restored to library &3.

CPF3840
Specified file for restore operation not found.

CPF38A5
Error on the PATTERN parameter.

CPF5729
Not able to allocate object &1.

CPF9802
Not authorized to object &2 in &3.

CPF9825
Not authorized to device &1.

OPT1498
Volume name list exhausted on device &1.

OPT1502
Attempted to process past the end of a multi-volume set.

OPT1605
Media or device error occurred.

Top

Restore Authority (RSTAUT)

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

Parameters
Examples
Error messages

The Restore Authority (RSTAUT) command restores the private authorities to user profiles. This command restores the same object authority to specified objects in the user profile that each user profile had when all the profiles were saved by the Save System (SAVSYS) or the Save Security Data (SAVSECDTA) command. It allows existing authorities, given after the save, to remain. Authority cannot be restored to the user profiles until the profiles are first restored to the system by the Restore User Profile (RSTUSRPRF) command and all the objects (for which authority is being given) are restored to the same libraries where they were saved. The objects can be restored by the Restore Library (RSTLIB) or Restore Object (RSTOBJ) command. Documents and folders can be restored using the Restore Document Library Object (RSTDLO) command. Device configuration objects can be restored using the Restore Configuration (RSTCFG) command. Integrated file system objects can be restored by the Restore Object (RST) command.

If the whole system is being restored, the following sequence must be followed. Using the RSTAUT command must be the last step in the sequence.

1. Restore the operating system. This is an alternative method to load the program. This restores the QSYS library and ensures that the IBM-supplied user profiles are there.
2. Restore all the saved user profiles to the system (*ALL is the default for the USRPRF parameter) by using the RSTUSRPRF command.
3. Restore all the configuration and system resource management (SRM) objects to the system by using the RSTCFG command.
4. Restore all the user libraries by using the RSTLIB command.
5. Restore all document library objects to the system by using the RSTDLO command.
6. Restore all objects in directories using the RST command.
7. Restore the object authority to user profiles by using the RSTAUT command.

Note: Steps 2 through 7 can be done more than once. For example, after the user profiles are restored (step 2), the user can restore only critical application libraries (step 3), followed by a restore of object authority (step 7). This supplies an operational system limited to using only the critical libraries. Later, the remaining user profiles can be restored, followed by the operations to restore the libraries and object authority.

If authorities for a user profile are restored using the RSTAUT command while all subsystems are ended, the user profile must be restored again before other authorities for it can be restored.

If one user profile is being restored, the following sequence must be followed. Using the RSTAUT command must be the last step.

1. Restore the specified user profile to the system by using the RSTUSRPRF command.
2. Restore all the device configuration and SRM objects to the system by using the RSTCFG command.
3. Restore the specified user libraries to the system by using the RSTLIB command or the RSTOBJ command. If the user profile is being restored because the current profile on the system is damaged, then the needed libraries already exist on the system and restoring of the libraries is not necessary.
4. Restore all document library objects to the system using the RSTDLO command.
5. Restore all objects in directories using the RST command.

6. Restore the object authority to the user profile by using the RSTAUT command. The specified profile may have been restored using the RSTUSRPRF command.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must have save system (*SAVSYS) special authority to run this command.
- Only one RSTAUT command can be run on a system at one time.

Top

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	Single values: <u>*ALL</u> Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional, Positional 1
SAVASPDEV	Saved from ASP device	<i>Name, <u>*ALLAVL</u>, *, *SYSBAS, *CURASPGRP</i>	Optional
RSTASPDEV	Restore to ASP device	<i>Name, <u>*SAVASPDEV</u>, *SYSBAS</i>	Optional

Top

User profile (USRPRF)

Specifies the names of one or more user profiles to have their private authorities restored. The specified user profiles must first be restored using the Restore User Profile (RSTUSRPRF) command.

Single values

*ALL Specifies all of the user profiles that are restored but do not have their private authorities restored. This includes user profiles that were restored using multiple previous Restore User Profile (RSTUSRPRF) commands.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of sets of user profiles. A generic name is a character string that contains one or more characters followed by an asterisk (*). (If an * is not specified with the name, the system assumes that the name is a complete user profile name.)

name

Specify one or more names of specific user profiles. Both generic names and specific names can be specified in the same command. A maximum of 300 user profile names can be specified.

Top

Saved from ASP device (SAVASPDEV)

Specifies the auxiliary storage pool (ASP) device from which private authorities were saved.

*ALLAVL

The private authorities saved from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs are restored.

*

The private authorities saved from the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group are restored.

*SYSBAS

The private authorities saved from the system ASP and all basic user ASPs are restored.

*CURASPGRP

If the current thread has an ASP group, the private authorities saved from all independent ASPs in the ASP group are restored.

name Specify the name of the ASP device from which private authorities were saved.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device for which to restore the private authorities.

*SAVASPDEV

The private authorities are restored to the same ASPs from which they were saved.

*SYSBAS

The private authorities are restored to the system ASP (ASP number 1) or to the basic user ASPs (ASP numbers 2-32).

name Specify the name of the independent ASP device for which to restore the private authorities.

Top

Examples

Example 1: Restore All Authorities

```
RSTAUT
```

This command restores to each user profile the authority to use each object that the profile had at the time when the system was saved. The user profiles and the libraries and their objects must be restored before the RSTAUT command is sent.

Example 2: Restore Authorities for Specific Users

```
RSTUSRPRF  USRPRF(USER1 USER2 USER3 USER4)
RSTLIB    SAVLIB(USERLIB)
RSTAUT    USRPRF(USER1 USER2 USER3)
```

To each specified user profile that was successfully restored, this command restores the authority to use each object that the profile had at the time the system was saved. The user profiles and the libraries and their objects must be restored before the RSTAUT command is sent. Because USER4 was not specified in the RSTAUT command, its authorities are still available and may be restored at a later date.

Example 3: Restore Authorities for Specific Libraries

```
RSTUSRPRF  USRPRF(*ALL)
RSTLIB    SAVLIB(USERLIBA)
RSTLIB    SAVLIB(USERLIBB)
RSTLIB    SAVLIB(USERLIBC)
RSTAUT    USRPRF(*ALL)
```

This command restores private authorities for all restored user profiles on the system. This includes authorities for all user profiles restored by the RSTUSRPRF command. Other user profiles on the system that did not have their authorities restored before these commands were specified are also restored by the RSTAUT(*ALL) command.

Example 4: Restore Authorities After Multiple RSTUSRPRF Commands

```
RSTUSRPRF  USRPRF(USER1 USER2)
RSTLIB    SAVLIB(USERLIBA)
RSTUSRPRF  USRPRF(USER1 USER3)
RSTLIB    SAVLIB(USERLIBB)
RSTAUT    USRPRF(*ALL)
```

This command restores private authorities for USER2 and USER3 and for the most recent version of USER1. Because the user profiles have the same name, the second RSTUSRPRF command overlays the first version of USER1.

Example 5: Restore Libraries and Authorities to an Independent ASP

```
RSTUSRPRF  USRPRF(*ALL) SAVASPDEV(*SYSBAS)
RSTLIB    SAVLIB(USERLIBA) RSTASPDEV(DIVISION1)
RSTLIB    SAVLIB(USERLIBB) RSTASPDEV(DIVISION1)
RSTLIB    SAVLIB(USERLIBC) RSTASPDEV(DIVISION1)
RSTAUT    USRPRF(*ALL) SAVASPDEV(*SYSBAS)
          RSTASPDEV(DIVISION1)
```

This example shows a way to move data and authorities to an independent auxiliary storage pool (ASP). The RSTUSRPRF command restores all user profiles and the private authority information saved from the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2-32). The RSTLIB commands restore libraries USERLIBA, USERLIBB, and USERLIBC to the independent ASP named DIVISION1. The RSTAUT command restores authorities saved from the system ASP and basic user ASPs for all user profiles to objects that now exist on the DIVISION1 ASP.

Top

Error messages

*ESCAPE Messages

CPF2206

User needs authority to do requested function on object.

CPF222E

&1 special authority is required.

CPF3776

Not all user profiles had all authorities restored.

CPF3785

Not all subsystems ended.

CPF3855

RSTAUT not allowed at this time.

CPF386D

Prestart job failed.

CPF9814

Device &1 not found.

CPF9833

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

CPF8ED

Device description &1 not correct for operation.

Top

Restore Configuration (RSTCFG)

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

Parameters
Examples
Error messages

The Restore Configuration (RSTCFG) command restores to the system a configuration object that was saved by the Save System (SAVSYS) or Save Configuration (SAVCFG) command. The types of objects that can be restored by this command are listed on the **Object types (OBJTYPE)** parameter. The QDFTOWN user profile becomes the default owner of any objects restored on the system whose owner is not known to the system. If an object already exists in the library to which that object is restored, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be given again.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- You must have save system (*SAVSYS) special authority, or have object existence (*OBJEXIST) authority for (or be the owner of) each object specified if the object already exists on the system.
- This command is shipped with no public authority (*EXCLUDE).
- The device configuration object must be varied off when it is being restored. To vary off a device configuration object, use the Vary Configuration (VRYCFG) command.
- System resource management (SRM) objects are not restored if the RSTCFG command is run using media that was created prior to V2R2M0.
- If the RSTCFG command and the SAVSYS or SAVCFG commands are not run on the same system, the configuration objects may not match the physical hardware on the target system.
- If you restore system resource management objects on a system other than the one on which the SAVSYS or SAVCFG command was saved, the system then treats the target system hardware as new and creates all new resource names, making the existing configuration descriptions useless. If this occurs, you need to restore the correct system resource management objects from the most current SAVSYS or SAVCFG command for that command. If neither of these is available, you must change existing configuration descriptions to reflect the new resource names.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL, *SRM Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
OBJTYPE	Object types	Single values: *ALL Other values (up to 10 repetitions): *CFGL, *CNNL, *COSD, *CTLD, *DEVD, *LIND, *MODD, *NTBD, *NWID, *NWS	Optional, Positional 3
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 4
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional
ENDOPT	End of media option	*REWIND , *LEAVE, *UNLOAD	Optional

Keyword	Description	Choices	Notes
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRM	System Resource Management	<i>*ALL, *NONE, *HDW, *TRA</i>	Optional
OPTFILE	Optical file	<i>Path name, _</i>	Optional
ALWOBJDIF	Allow object differences	Single values: <i>*NONE, *ALL</i> Other values (up to 3 repetitions): <i>*AUTL, *OWNER, *PGP</i>	Optional
OUTPUT	Output	<i>*NONE, *PRINT, *OUTFILE</i>	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualifier list</i>	
	Qualifier 1: Object	<i>Generic name, name, *NONE, *ALL</i>	
	Element 2: Object type	<i>*ALL, *CFGL, *CNL, *COSD, *CTLD, *DEVD, *LIND, *MODD, *NTBD, *NWID, *NWS</i>	

Top

Objects (OBJ)

Specifies the objects to be restored. Specify the name of each object, or the generic name of each group of objects to restore. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk (*) is not specified with the name, the system assumes that the name is a complete object name.

If the **Object types (OBJTYPE)** parameter has a value of **ALL*, all the object types listed in the description of the OBJTYPE parameter are restored, if they have the specified names.

System resource management (SRM) objects cannot be restored individually or by specifying a generic name. To restore only SRM objects, specify **SRM* for this parameter and a value for the **System Resource Management (SRM)** parameter.

This is a required parameter.

Single values

***ALL** All the device configuration objects are restored, depending on the values specified for the OBJTYPE parameter.

***SRM** The device configuration objects are not restored, but system resource management (SRM) objects are restored based on the SRM parameter value.

ATTENTION You must specify SRM(*NONE) on the RSTCFG, unless the system you are restoring to is the exact same hardware configuration that the original configuration was saved on, to prevent the restore of the SRM information. If the SRM information is restored, the configuration objects may become unusable.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of objects in the specified library to restore.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify one or more names of specific objects to restore. Both generic names and specific names can be specified in the same command. A maximum of 300 object names can be specified.

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume.

Top

Object types (OBJTYPE)

Specifies the types of system objects that are to be restored.

Single values

***ALL** All of the device configuration object types that are specified by name are restored. If *ALL is also specified for the **Objects (OBJ)** parameter, all of the saved device configuration objects are restored.

Other values (up to 10 repetitions)

character-value

Specify the value for each of the types of objects that are to be restored.

The object types shown are the ones that are saved in the device configuration file by the Save System (SAVSYS) command or the Save Configuration (SAVCFG) command.

***CFGL**

Configuration lists

***CNL**

Connection list

***COSD**

Class-of-service description

***CTLD**

Controller description

***DEVD**

Device description

***IPXD** Internet Packet Exchange description

***LIND**

Line description

***MODD**

Mode description

***NTBD**

NetBIOS description

***NWID**

Network interface description

***NWSD**

Network server description

Note: *SRMSPC can be saved but not restored as an object type. To restore *SRM data, specify *SRM for the OBJ parameter.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Note: The first version of the objects found in the specified location is restored, unless a specific version is identified by the SAVDATE parameter and SAVTIME parameter, or for tape, the SEQNBR parameter.

Single values

***MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

Specifies the sequence number that is used for the restore operation.

*SEARCH

The volume placed in a device is searched for a data file containing the saved device configuration objects. When a match is found, the configuration objects are restored.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

System Resource Management (SRM)

Specifies the type of system resource management (SRM) information to be restored. This parameter is valid only when *ALL or *SRM is specified on the **Objects (OBJ)** parameter.

ATTENTION Unless the system you are restoring to has exactly the same hardware configuration as the system that the original configuration was saved on, you must specify SRM(*NONE) on this command to prevent the restore of the SRM information. If the SRM information is restored on a system with a different hardware configuration, the configuration objects may become unusable.

***ALL** All system resource management information is restored.

***NONE**

No system resource management information is restored.

***HDW**

All hardware information is restored.

***TRA** All token-ring adapter information is restored.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

***** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'*optical-directory-path-name*'

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.
Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

***NONE**

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

Other values (up to 3 repetitions)

***AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

- ***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

***NONE**

No output is created.

***PRINT**

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

***OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Element 2: Object type

***ALL** All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Examples

Example 1: Restoring All Objects

```
RSTCFG OBJ(*ALL) DEV(TAP01) OBJTYPE(*ALL)
```

This command restores all of the device configuration and SRM objects from the tape on the TAP01 drive.

Example 2: Restoring a Device Description

```
RSTCFG OBJ(PRT01) DEV(TAP01) OBJTYPE(*DEV) VOL(ABCD)
```

The device description for PRT01 that was saved on tape volume ABCD is restored to the system. If device description PRT01 already exists on the system, it must be varied off before it can be restored.

Example 3: Restoring a Network Server Description

```
RSTCFG OBJ(SERVER1) DEV(TAP01) OBJTYPE(*NWS) VOL(SAV1)
```

The network server description SERVER1 that was saved on tape volume SAV1 is restored to the system. If the network server description SERVER1 already exists on the system, it must be varied off before it can be restored.

Top

Error messages

*ESCAPE Messages

CPF26C1

Network server &1 not varied off.

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3707

Save file &1 in &2 contains no data.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3733

&2 &1 in &3 previously damaged.

CPF3738

Device &1 used for save or restore is damaged.

CPF3743

File cannot be restored, displayed, or listed.

CPF3748
Object information for library &1 damaged.

CPF376B
File &1 not found.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3769
Data encrypted or not save data.

CPF377B
No objects restored. &3 objects excluded.

CPF377C
&1 objects restored; &2 not restored; &6 excluded.

CPF3780
Specified file for library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF379B
Objects from save file &1 in &2 not restored.

CPF3791
While processing &2 &1 in &3, encountered end of file &4.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

CPF3805
Objects from save file &1 in &2 not restored.

CPF3807
Data decompression error for save file &1 in &2.

CPF3812
Save file &1 in &2 in use.

CPF3883
S/38 device configuration objects not restored.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

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.

[Top](#)

Restore Deferred Objects (RSTDFROBJ)

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

Parameters
Examples
Error messages

The Restore Deferred Objects (RSTDFROBJ) command completes the restore of deferred objects, if the objects they depend on are now available.

Deferred objects can be logical files or SQL materialized query tables (MQTs). A deferred logical file is not created until the restore is complete. A deferred MQT is created, but until the restore is complete, any functions performed on the MQT that require access to the based-on files will fail.

Restrictions:

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

Top

Parameters

Keyword	Description	Choices	Notes
DFRID	Defer ID	<i>Name</i>	Required, Positional 1
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	*REPLACE, *ADD	
INFTYPE	Type of output information	*OBJ, *MBR	Optional

Top

Defer ID (DFRID)

Specifies the name that was used for the **Defer ID (DFRID)** parameter on the restore operations.

This is a required parameter.

name Specify the name that was used for the **Defer ID (DFRID)** parameter on the restore operations.

Top

Output (OUTPUT)

Specifies whether a list with information about the restored objects is created. The information can be printed with the job's spooled output or directed to a database file.

*NONE

No output is created.

*PRINT

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

*LIBL The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

*CURLIB

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Type of output information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

***OBJ** The list contains an entry for each object requested to be restored.

***MBR** The list contains an entry for each object, database file member, and spooled file requested to be restored.

Top

Examples

```
(Restore libraries before LFLIB.)
RSTLIB SAVLIB(LFLIB) DEV(TAP01) ENDOPT(*LEAVE) DFRID(ABC)
(Restore libraries between LFLIB and PFLIB.)
RSTLIB SAVLIB(PFLIB) DEV(TAP01) ENDOPT(*LEAVE) DFRID(ABC)
(Restore libraries after PFLIB.)
RSTDFROBJ DFRID(ABC)
```

This example shows libraries that were saved in alphabetical order being restored. Library LFLIB contains files that have based-on files in library PFLIB. The files in LFLIB cannot be restored until after the files in PFLIB are restored. A DFRID is specified, so the files in LFLIB are deferred. The restore of these files is completed by using the RSTDFROBJ command, which does not use the tape. If a DFRID had not been specified, the tape would need to be rewound or reloaded to restore the files in LFLIB.

Top

Error messages

*ESCAPE Messages

CPF222E

&1 special authority is required.

CPF32C7

Deferred file not restored.

CPF37C1

&2 objects completed. &3 objects deferred.

Top

Restore Document Lib Object (RSTDLO)

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

Parameters
Examples
Error messages

The Restore Document Library Object (RSTDLO) command restores documents, folders, and distribution objects (mail).

This command can be used to restore the documents and folders if the document was or was not freed by the Save Document Library Object (SAVDLO) command, or to restore documents and folders that were deleted by the Delete Document Library Object (DLTDLO) command.

Restoring a document either replaces the existing document content and control information if the document exists on the system, or it adds new document content and control information if the document does not exist.

For a filed document (electronic mail or a document stored in the document library), the document and folder name of the document object on the media must be the same as the document name and folder name of the document on the system, unless the document is renamed and put in a different folder during the restore operation.

Note: Folder names must match exactly for restored folders. All objects that are not in use are restored from the folder on the media or in the save file to the existing folder. Restoring a folder creates a new folder object if the folder does not exist and adds to this new folder all objects saved with the folder on the media or in the save file. If the folder exists, any document or folder objects that do not exist within it are created. The existing documents are replaced with the version from the media.

For a filed document restored on the system whose owner is not known to the system or is not enrolled in the system distribution directory, the user profile of the default owner (QDFTOWN) becomes the owner of the document or folder.

The creation date of a document does not change if the document exists. If the document does not exist, the creation date is set to the date on which the document is created.

The security does not change if a document or folder exists on the system where it is to be restored. If the document or folder does not exist, public authority, authorization list, and personal status are restored; however, all other private document and folder authorities are not restored. These authorities must be established again by the owner.

If a document is restored that had a mail log entry when it was saved, the mail log entry is restored if the distribution tracking object exists on the system. If the distribution tracking object does not exist on the system, a message is sent saying that the document was restored without a mail log entry.

If this command ends abnormally, objects are left on the system in an unknown state and cannot be found in a library. This can happen if a power failure occurs when this command is run. The Reclaim Storage (RCLSTG) command can be used to clean up the auxiliary storage and delete most of those objects from the system; however, unknown mail objects are not cleaned up with the RCLSTG command.

When a set of documents and folders are restored, all documents and folders in the set must exist in the same tape, optical volume, or save file.

If a document exists in more than one tape file, the user can control which document is restored by specifying the media file using the sequence number or label parameter. If more than one version of the document exists, the SAVDATE and SAVTIME parameters can also be used to select the correct document.

When text search services are on the system and the user restores a document library object, the text search index for the object is restored.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority or be enrolled in the system distribution directory to run this command.
- This command cannot be run when RCLDLO DLO(*ALL) is running because RCLDLO requires exclusive use of internal objects.
- When saving or restoring to an existing database file using the OUTFILE parameter, you must have execute (*EXECUTE) authority to the library where the output database file is located.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SYSOBJNAM, *MAIL Other values (up to 300 repetitions): <i>Character value</i>	Required, Positional 1
SAVFLR	Saved folder	Single values: *ANY, *NONE Other values (up to 300 repetitions): <i>Character value</i>	Optional
RENAME	New object name	Single values: *SAME Other values (up to 300 repetitions): <i>Character value</i>	Optional
RSTFLR	Restore into folder	<i>Character value</i> , *SAME	Optional
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
SYSOBJNAM	System object name	Values (up to 300 repetitions): <i>Name</i> , *NONE	Optional
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	Single values: *SEARCH Other values: <i>Element list</i>	Optional
	Element 1: Beginning sequence number	1-16777215	
	Element 2: Ending sequence number	1-16777215, *ONLY	
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
LABEL	Label	<i>Character value</i> , *GEN	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	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
NEWOBJ	Object name generation	*SAME, *NEW	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional
ALWOBJDIF	Allow object differences	*NONE, *ALL	Optional
SAVASP	Saved from ASP number	1-32, *ANY	Optional

Keyword	Description	Choices	Notes
RSTASP	Restore to ASP number	1-32, <u>*SAVASP</u>	Optional
OUTPUT	Output	<u>*NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	

Top

Document library object (DLO)

Specifies the document library objects to be restored.

This is a required parameter.

Single values

***ALL** All documents, folders, and distribution objects (mail) that are saved on media and meet the criteria specified for the **Saved folder (SAVFLR)** parameter are restored.

***MAIL**
All distribution objects and documents that were referred to by a mail log are restored.

***SYSOBJNAM**
All documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are restored.

Other values (up to 300 repetitions)

document-name

Specify the user-assigned names of the documents to be restored. All documents named must be in the folder specified for the SAVFLR parameter.

Top

Saved folder (SAVFLR)

Specifies the name of the folder on the media from which the documents and folders are restored.

Single values

***ANY** All document library objects that meet the values specified for the **Document library object (DLO)** parameter are restored, regardless of the folders (if any) from which they were saved. This value is valid only if *ALL, *MAIL, or *SYSOBJNAM is specified for the DLO parameter.

***NONE**
All documents that were saved as documents not in any folder are restored. *NONE is valid for documents only and is valid only when *ALL is specified for the DLO parameter.

Other values (up to 300 repetitions)

folder-name

Specify the name of the saved folder from which documents are to be restored, or the name of the folders to be restored if *ALL was specified for the DLO parameter. A maximum of 63 characters can be specified for the folder name. When *ALL is specified for the DLO parameter, a maximum of 300 folder names can be specified. The name of a saved folder must be specified when DLO(document-name) is specified.

Top

New object name (RENAME)

Specifies the new user-assigned name for the restored document.

Single values

*SAME

The documents are restored with the names they had when they were saved.

Other values (up to 300 repetitions)

document-name

Specify the new user-assigned names that the documents have after being restored. When document names are specified for the **Document library object (DLO)** parameter, a maximum of 300 user-assigned names can be specified for documents being restored.

Top

Restore into folder (RSTFLR)

Specifies the name of the folder in which the restored folders and documents to be restored will be placed. The folder must exist on the system or when *ALL is specified for the **Document library object (DLO)** parameter, the saved folder must exist on the media.

*SAME

The folders and documents to be restored are placed into the same folder from which they were saved.

folder-name

Specify the name of the folder where the restored documents and folders are to be placed.

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume.

Top

System object name (SYSOBJNAM)

Specifies the system object names of the documents to be restored if *SYSOBJNAM is specified for the Document library object (DLO) parameter.

Single values

*NONE

A system object name is not specified.

Other values (up to 300 repetitions)

system-object-name

Specify the system object names of the documents to be restored.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Single values

*MOUNTED

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

Specifies the sequence numbers of the tape files used for the restore operation.

Note:When restoring document library objects (DLOs) from more than one auxiliary storage pool (ASP), a beginning and ending sequence number must be specified. The beginning and ending sequence numbers were shown in the completion message for the SAVDLO command used to create the tape media files. Or use the Display Tape (DSPTAP) command to find the sequence numbers of the QDOC and QDOCnnnn files, where nnnn is the ASP number.

Single values

*SEARCH

The tape is searched for the first data file with an identifier matching the **Label (LABEL)** parameter value and with contents of a minimum of one of the specified document library objects. If the last operation on the device specified ENDOPT(*LEAVE) (that is, the tape is positioned at the location at which the last operation ended), the file search begins with the first data file beyond the current tape position. If ENDOPT(*LEAVE) was not specified on the last operation (or if the tape has been rewound since the ENDOPT(*LEAVE) operation), the search begins with the first data file on the volume.

Element 1: Beginning sequence number

1-16777215

Specify the sequence number of the first file used for the restore operation.

Element 2: Ending sequence number

*ONLY

The ending sequence number is the same as the starting sequence number.

1-16777215

Specify the sequence number of the last file used for the restore operation.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Label (LABEL)

Specifies the file label used to find the file that was written onto the media during the save operation.

***GEN** The system generates the default name of the file label for which to search.

character-value

Specify the file label of the file that contains the document library objects to be restored. A maximum of 17 alphanumeric characters can be used.

Top

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

***** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'optical-directory-path-name*'

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Object name generation (NEWOBJ)

Specifies whether a new library-assigned name and system object name are generated for the folders and documents that are restored.

***SAME**

The library-assigned name and system object name do not change.

***NEW** A new library-assigned name and system object name are generated for each document or folder that is restored.

Top

Date when saved (SAVDATE)

Specifies the date on which the document library objects were saved. If more than one version of the document library objects exist on the media, use this parameter to identify which version of the document library objects to restore. The date must be specified in the job date format. If separators are used, the value must be enclosed in apostrophes. If the SAVDATE parameter is not specified, the version of the documents and folders to be restored will be the first version found on the volume or the version found with the specified file label.

Top

Time when saved (SAVTIME)

Specifies the time when the document library objects were saved. If more than one version of the document library objects exist on the media with the same value for the date saved, use this parameter to identify which version of the document library objects to restore. Specify the time as a 6-digit value, in the format hours, minutes, seconds (hhmmss). If separators are used, the value must be enclosed in apostrophes ('hh:mm:ss'). If a volume identifier is specified, but the **Time when saved (SAVTIME)** parameter is not specified, the version of the document library objects to be restored will be the first version found on the volume or the first version found with the specified file label.

This parameter is valid only if the **Date when saved (SAVDATE)** parameter is specified also.

Top

Allow object differences (ALWOBJDIF)

Specifies whether the following differences encountered during a restore operation are allowed.

- Ownership—the owner of the object on the system is different than the owner of the object from the save operation.
- System object name—the system object name on the system does not match the system object name on the media.
- Authorization list linking—the object is being restored to a system different from the one on which it was saved.

The ALWOBJDIF parameter can be used to allow an object to be restored whose owner or object name on the system is different than on the media used for the restore operation. By specifying the ***ALL** special value, an object with a different name is restored to the name on the media, while an object with a different owner keeps the owner name from the system instead of the media.

Note: To use this parameter, you need all object (***ALLOBJ**) authority.

***NONE**

None of the differences described above are allowed on the restore operation. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to ***EXCLUDE**. For other cases, a diagnostic message is sent for the object, and the object is not restored.

***ALL** All of the differences described above are allowed for the restore operation. An informational message is sent, and the object is restored.

Notes:

- If the owners of the object do not match, the object is restored, but it will keep the ownership and authorities of the object on the system before the restore operation.
- If *ALL is specified on this parameter, *NEW cannot be specified for the **Object name generation (NEWOBJ)** parameter.
- If you are restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

Top

Saved from ASP number (SAVASP)

Specifies the number of the auxiliary storage pool (ASP) on media from which saved documents and folders are to be restored.

*ANY The documents and folders saved in any ASP are restored.

Note:When restoring document library objects (DLOs) from more than one ASP, the sequence number (SEQNBR) parameter must be specified.

1-32 Specify the number of the ASP from which documents and folders are restored.

Top

Restore to ASP number (RSTASP)

Specifies the number of the auxiliary storage pool (ASP) on media in which restored documents and folders are to be placed.

*SAVASP

The documents and folders are placed in the same ASP from which they were saved.

1-32 Specify the number of the ASP in which restored documents and folders are placed.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

*NONE

No output is created.

*PRINT

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

File to receive output (OUTFILE)

Specifies the qualified name of the database file to which the information about the object is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJRSTO in QSYS with the format name QOJRST as a model.

Qualifier 1: File to receive output

name Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

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.

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Examples

Example 1: Restoring Documents with System Object Names

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01) SYSOBJNAM(HZ83B55219)
```

This command restores the document named HZ83B55219 from the tape unit TAP01.

Example 2: Restoring Documents from a Save Folder

```
RSTDLO DLO(A) DEV(TAP01) SAVFLR(X)
```

This command restores the document named A from folder X.

Example 3: Restoring All Documents

```
RSTDLO DLO(*ALL) DEV(TAP01)
```

This command restores all documents and folders that are on the first tape file on tape unit TAP01.

Example 4: Restoring a Folder Saved from the System ASP to a User ASP

```
RSTDLO DLO(*ALL) FLR(Y) SAVASP(1) RSTASP(2)
```

This command restores folder Y, which was saved from ASP 1, to user ASP 2. Folder Y must be deleted from ASP 1 before it can be restored to ASP 2.

Example 5: Creating New Library-Assigned Name

```
RSTDLO DLO(*SYSOBJNAM) DEV(TAP01) SYSOBJNAM(HZ83B55219)  
        NEWOBJ(*NEW)
```

This command restores document HZ83B55219 from tape unit TAP01 and gives it a new library-assigned name and a new system object name.

Example 6: Renaming Documents

```
RSTDLO DLO(A B) DEV(TAP01) SAVFLR(C)  
        RENAME(Y Z) RSTFLR(X)
```

This command restores documents A and B from within folder C. Document A is renamed to Y and document B is renamed to Z. The command then puts them in folder X.

Example 7: Specifying Sequence Numbers

```
RSTDLO DLO(*ALL) DEV(tape-device-name) SAVFLR(A)  
        SEQNBR(1 3) LABEL(*GEN)
```

This command restores all of folder A from tape files with the sequence numbers 1, 2, and 3, and the label QDOC or QDOCxxxx.

Example 8: Specifying Allowed Differences

```
RSTDLO DLO(A) DEV(TAP01) SAVFLR(X) ALWOBJDIF(*ALL)
```

This command restores document A from folder X. If document A in folder X exists on the system and the owner of the document on the system does not match the owner of the document being restored, the document is restored and the owner of the document on the system remains unchanged.

Example 9: Reporting Information about Objects Restored and Not Restored

```
RSTDLO DLO(*ALL) DEV(TAP01) OUTPUT(*OUTFILE)
        OUTFILE(INFO92) OUTMBR(FOURQT *ADD)
```

This command restores all documents and folders from the tape device TAP01. A list reporting information about objects restored and objects not restored is directed to the output file INFO92. The output is received in the member FOURQT as an addition to existing information in the member.

Top

Error messages

*ESCAPE Messages

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3718

Restore command not valid for file &1.

CPF3728

Device &1 specified with other devices.

CPF3767

Device &1 not found.

CPF3780

Specified file for library &1 not found.

CPF3782

File &1 in &2 not a save file.

CPF381B

No DLOs restored to ASP &1.

CPF3812

Save file &1 in &2 in use.

CPF384D

Save or restore operation not allowed on ASP &1.

CPF8AB5

ASP &5 is not configured.

CPF8A47

Internal system objects in use.

CPF90AF
RSTFLR value not allowed.

CPF90A4
RENAME value not allowed.

CPF90B4
&1 folders restored to system, &2 not restored.

CPF90CD
Not authorized to restore distributions.

CPF90CF
Search index data base is damaged.

CPF90E0
Not enough authority for ALWOBJDIF(*ALL).

CPF90E7
Document library objects not restored.

CPF9003
&1 document library objects restored. &10 not restored.

CPF905C
Error occurred trying to find a translation table.

CPF9050
Ending sequence number not valid.

CPF9069
User not permitted to restore into folder &1.

CPF908A
Requester &1 not enrolled.

CPF909B
&1 document library objects restored. &10 not restored.

CPF9412
List of folder names not allowed with DLO parameter.

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.

CPF9831
Cannot assign device &1.

CPF9845
Error occurred while opening file &1.

- CPF9846**
Error while processing file &1 in library &2.
- CPF9850**
Override of printer file &1 not allowed.
- CPF9851**
Overflow value for file &1 in &2 too small.
- CPF9860**
Error occurred during output file processing.
- CPF9899**
Error occurred during processing of command.
- OPT1498**
Volume name list exhausted on device &1.
- OPT1502**
Attempted to process past the end of a multi-volume set.
- OPT1605**
Media or device error occurred.

[Top](#)

Restore Library (RSTLIB)

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

Parameters
Examples
Error messages

The Restore Library (RSTLIB) command restores to the system one library or a group of libraries that was saved by the Save Library (SAVLIB) command. The Restore Library (RSTLIB) command restores the whole library, which includes the library description, object descriptions, and contents of the objects in the library.

For job queues, message queues, user queues, and logical files, only the object descriptions are restored, because only the definitions are saved. Also, logical file access paths may be restored if they were saved. More information on restoring access paths is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. The contents of spooled files on output queues can be restored, if they were saved, by specifying *NEW for the **Spooled file data (SPLFDTA)** parameter.

This command can be used to restore libraries whose objects had their storage freed by the corresponding SAVLIB command of the restore operation, or libraries deleted by the Delete Library (DLTLIB) command. If the data portions of the objects in the saved libraries were not freed, each library is copied into the same area of storage that it previously occupied. If the storage was freed, the system finds the needed storage to store the library contents (the object description and data portion of every file, module, program, service program, Structured Query Language (SQL) package, and journal receiver in the library). If the library does not exist on the system because it has been deleted or is being restored on a different system, the system must find the storage for everything that is in the library, including the library description.

If an object is being restored over an existing object on the system, the owner, primary group, public and private authorities, authorization list, and object auditing value of the existing object are retained. The **Allow object differences (ALWOBJDIF)** parameter may affect whether the object is restored, and the **Private authorities (PVTAUT)** parameter may add saved private authorities to the existing private authorities. Existing objects that are in use, such as output queues that are actively spooling, are not restored. Existing data queues are not restored.

If an object is being restored as new to the system, the saved owner name, primary group name, public authority, and object auditing value are used. The ALWOBJDIF parameter may affect whether the saved authorization list name is used, and the PVTAUT parameter may restore saved private authorities. If the owner user profile does not exist, the system default owner (QDFTOWN) becomes the owner of the object. If the primary group does not exist, *NONE is used.

If a library is being restored over an existing library on the system, the default public authority and auditing values for objects created in the existing library are retained. If the library is being restored as new; these values are restored from the media.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must have save system (*SAVSYS) special authority, or have all of the following object authorities:
 1. Read (*READ) and add (*ADD) authorities for, or be the owner of, each library specified.
 2. Object existence (*OBJEXIST) authority for, or be the owner of, each object in the library if the object already exists in the library on the system. *OBJEXIST and use (*USE) authorities are

required for message queue objects. If the object does not exist, add (*ADD) authority for the user profiles that own the objects to be restored. For spooled file data, *OBJEXIST authority is required for the output queue to which it is restored.

If you do not have the correct authority for all the libraries and objects specified, only those for which you have authority are restored.

3. If VOL(*SAVVOL) is specified, *USE authority to the saved-from library.
 4. *USE authority for the save file is required when restoring libraries from a save file. *USE authority for the device description and the device file are required when restoring libraries from a tape or optical device.
- You must have use (*USE) authority for the Create Save File (CRTSAVF) command when restoring a save file that does not currently exist on the system.
 - When using a media definition, you must have *USE authority to the media definition and execute (*EXECUTE) authority to the media definition library.
 - When saving or restoring to an existing database file using the OUTFILE parameter, you must have *EXECUTE authority to the library where the output file is located.
 - The current versions of programs on the system should not be run while the library that contains those programs is being restored. If any OPM program is running while it is being restored, the running program may fail or behave unpredictably.

Top

Parameters

Keyword	Description	Choices	Notes
SAVLIB	Saved library	Single values: *NONSYS, *ALLUSR, *IBM, *ANY Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
DEV	Device	Single values: *SAVE, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
VOL	Volume identifier	Single values: *MOUNTED, *SAVVOL Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional, Positional 4
LABEL	Label	<i>Character value</i> , *SAVLIB	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	<i>Name</i> , *FIRST	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	
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
OMITLIB	Libraries to omit	Single values: *NONE Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OPTION	Option	*ALL, *NEW, *OLD, *FREE	Optional
MBROPT	Data base member option	*MATCH, *ALL, *NEW, *OLD	Optional
DFRID	Defer ID	<i>Name</i> , *DFT, *NONE	Optional
SPLFDTA	Spooled file data	*NEW, *NONE	Optional
PVTAUT	Private authorities	*NO, *YES	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional

Keyword	Description	Choices	Notes
SAVTIME	Time when saved	<i>Time</i>	Optional
ALWOBJDIF	Allow object differences	Single values: <u>*NONE</u> , *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: *SYSVAL, *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	
RSTLIB	Restore to library	<i>Name</i> , <u>*SAVLIB</u>	Optional
RSTASPDEV	Restore to ASP device	<i>Name</i> , <u>*SAVASPDEV</u>	Optional
RSTASP	Restore to ASP number	1-32, <u>*SAVASP</u>	Optional
OUTPUT	Output	<u>*NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , <u>*FIRST</u>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
INFTYPE	Type of output information	<u>*OBJ</u> , *MBR	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name</i> , <u>*NONE</u> , *ALL	
	Qualifier 2: Library	<i>Generic name, name</i> , <u>*ALL</u>	
	Element 2: Object type	<i>Character value</i> , <u>*ALL</u>	

Top

Saved library (SAVLIB)

Specifies the libraries to be restored to the system.

Notes:

- If you specify more than one library, or a special value, or a generic value:
 - The **Device (DEV)** parameter must not specify *SAVF.
 - The **Restore to library (RSTLIB)** parameter must specify *SAVLIB.
 - The **Label (LABEL)** parameter must specify *SAVLIB.
 - The **Optical file (OPTFILE)** parameter must specify * or end with /*.
- For values *NONSYS and *IBM, all other operations on the system must be ended before this option is specified. This requires ending all subsystems through the End Subsystem (ENDSBS) command or the End System (ENDSYS) command.
- The tape or optical file that you select to restore may contain a library that does not match the value that you specify for this parameter. This parameter is used to determine the default LABEL value for a tape file and the default OPTFILE value for an optical file, but the restore operation does not verify that this parameter matches the saved library.

This is a required parameter.

Single values

*NONSYS

Libraries saved by the Save Library (SAVLIB) command with LIB(*NONSYS) specified are restored.

You can do a RSTLIB SAVLIB(*IBM) and a RSTLIB SAVLIB(*ALLUSR) from a SAVLIB LIB(*NONSYS).

*ALLUSR

All user libraries are restored. All libraries with names that do not begin with the letter Q are restored 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 restored:

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

*IBM Restores all IBM-supplied libraries except for the following:

QDOC	QRCYxxxxx	QTEMP	QUSRPYMSVR
QDOCxxxx	QRECOVERY	QUSER38	QUSRDRARS
QDSNX	QRPLOBJ	QUSRADSM	QUSRSYS
QGPL	QRPLxxxxx	QUSRBRM	QUSRVI
QGPL38	QSPL	QUSRDIRCF	QUSRVxRxMx
QMGTC	QSPLxxxx	QUSRDIRCL	
QMGTC2	QSRV	QUSRDIRDB	
QMPGDATA	QSRVAGT	QUSRIJS	
QMOMDATA	QSYS	QUSRINFSKR	
QMOMPROC	QSYSxxxxx	QUSRNOTES	
QPFRDATA	QSYS2	QUSROND	
QRCL	QSYS2xxxxx	QUSRPOSGS	
QRCLxxxxx	QS36F	QUSRPOSSA	

Note: 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.

The following libraries with names that do not begin with the letter Q are also restored:

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

*ANY Restores the first version of all saved libraries found on the tape beginning with the sequence

number specified for the **Sequence number (SEQNBR)** parameter, or restores all saved libraries found on the optical media in the directory specified for the **Optical file (OPTFILE)** parameter.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the library to be restored.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library to be restored. The name of the library being restored must be the same as the name that was used when the library was saved.

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

***MEDDFN**

The restore operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Note: The first version of the saved library that is found on the media is restored, unless a specific version is identified by the **Date when saved (SAVDATE)** and **Time when saved (SAVTIME)** parameters or, for tape, the **Sequence number (SEQNBR)** parameter.

Single values

*MOUNTED

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

*SAVVOL

The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the library. If the device type of the device specified for the **Device (DEV)** parameter does not match the device type of the most recently saved version of the library, an error message is sent to the user, and the function is ended. If *SAVVOL is specified, the SAVDATE and SAVTIME parameters cannot be specified. If *SAVVOL is specified, *SEARCH must be specified for the SEQNBR parameter.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

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

*SEARCH

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified *LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If *LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

If *NONSYS, *ALLUSR, or *IBM is specified for the **Saved library (SAVLIB)** parameter, the sequence number specifies the location of the QFILE. The QFILE file is at the beginning of the *NONSYS, *ALLUSR, or *IBM save operation. The QFILE file contains the list of libraries saved.

Top

Label (LABEL)

Specifies the name that identifies the data file to be used for the restore operation. This label must have been specified on the save command.

*SAVLIB

The file label is the name specified for the **Saved library (SAVLIB)** parameter.

character-value

Specify the data file identifier of the data file used for the restore operation. A maximum of 17 characters can be used. This option is valid only for a single-library restore.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

***UNLOAD**

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

Top

Starting library (STRLIB)

Specifies the name of the starting library for a *NONSYS, *IBM, or *ALLUSR restore operation.

If an unrecoverable media error occurs during the restore operation, this parameter can be used to restart the operation.

Note: In the recovery steps that follow, *NONSYS is specified for the **Saved library (SAVLIB)** parameter of this command. If you are restoring IBM-supplied libraries or all user-created libraries and IBM-supplied libraries, specify *IBM or *ALLUSR instead.

The basic recovery steps for a restore operation are:

1. Look at the job log to determine the library where the previous restore library (RSTLIB SAVLIB(*NONSYS)) command failed. Find the last library restored, which is indicated by a successful restore completion message.
2. Load the first tape of the SAVLIB LIB(*NONSYS) media.
3. Type the following command, and then press the Enter key:
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) ENDOPT(*LEAVE)
STRLIB(library-name) OMITLIB(library-name)

where the *library-name* for the STRLIB and OMITLIB parameters is the library where the last RSTLIB failed. This starts the restore operation on the library after the library where the restore operation failed.

4. When you are prompted, load the volume containing the starting library.
5. After the restore operation is complete, restore the library where the restore operation failed using the media from a previous save operation.

Note: Consider removing the tape with the media error from the next save rotation cycle to avoid another tape error.

Note: This parameter is valid only if *NONSYS, *IBM, or *ALLUSR is specified for the SAVLIB parameter.

***FIRST**

The restore operation begins with the first library saved.

name Specify the name of the library with which to start the restore operation.

Top

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Recovering your system book, SC41-5304, and the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

Qualifier 1: Media definition

name Specify the name of the media definition to be used.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

***** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'optical-directory-path-name*'

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Libraries to omit (OMITLIB)

Specifies a list of libraries to be excluded from the restore operation.

Single values

***NONE**

No libraries are excluded from the restore operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

library-name

Specify the name of the library to be excluded from the restore operation.

Top

Option (OPTION)

Specifies how to handle restoring each object.

***ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.

***NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.

- *OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- *FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

- *ALL** All members in the saved file are restored.
- *NEW** Only new members (members not known to the system) are restored.
- *OLD** Only members already known to the system are restored.

Top

Defer ID (DFRID)

Specifies the identifier to be used if you want to defer the restore of objects that depend on other objects that are not yet available. To complete the restore of deferred objects, restore the objects that they depend on, and specify the same Defer ID. If any objects remain deferred when the other objects are available, use the Restore Deferred Objects (RSTDFROBJ) command, and specify the same Defer ID. This parameter allows you to restore all objects in a set of libraries when the libraries with dependent objects are restored before the libraries with the objects they depend on.

Deferred objects can be logical files or SQL materialized query tables (MQTs). A deferred logical file is not created until the restore is complete. A deferred MQT is created, but until the restore is complete, any functions performed on the MQT that require access to the based-on files will fail.

Notes:

1. If the following conditions are true, the restore of a deferred object may be completed automatically when the objects it depends on are restored:
 - The deferred object is restored to the same library from which it was saved.
 - The same Defer ID is specified for the restore operations for both the deferred object and the objects it depends on.
- *DFT** If *NONSYS, *ALLUSR, or *IBM is specified for the **Saved library (SAVLIB)** parameter, and you have save system (*SAVSYS) special authority, identifier QRSTLIB is used. Objects will be deferred if they depend on other objects that are not available. In most cases, the deferred objects will be completed as this restore operation proceeds to restore the other objects.

If any other value is specified for the SAVLIB parameter, or you do not have *SAVSYS special authority, no identifier is used. Objects will not be restored or deferred if they depend on other objects that are not available.

***NONE**

Objects will not be restored or deferred if they depend on other objects that are not available.

name Specify an identifier to defer the restore of objects that depend on other objects that are not yet available. You must have save system (*SAVSYS) special authority to specify a name.

Top

Spooled file data (SPLFDTA)

Specifies whether to restore spooled file data and attributes.

***NEW** For each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system.

***NONE**

No spooled file data is restored.

Top

Private authorities (PVTAUT)

Specifies whether to restore private authorities with the objects that are restored.

***NO** No private authorities are restored.

***YES** Private authorities are restored with the objects. Objects will be restored only from save operations that specified that private authorities should be saved with the objects.

Note: You must have all object (*ALLOBJ) special authority to specify this value.

Top

Date when saved (SAVDATE)

Specifies the date when the library was saved. If the most recently saved version is not restored, or if more than one saved version is on the volumes, type the date that specifies which version of the library is restored.

If a volume identifier or *MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not specified, the first version of the library found on the volume is restored. This parameter is not valid with VOL(*SAVVOL).

date Specify the date that the library to be restored was saved. The date must be entered in the job date format.

Top

Time when saved (SAVTIME)

Specifies the time when the library was saved, if the current version is not restored.

If a volume identifier or *MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not, the first version of the library found on the volume is restored. This parameter is valid only if the **Date when saved (SAVDATE)** parameter is also specified.

time Specify the time that the library to be restored was saved. The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.
Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **File level id:** The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- **Member level id:** The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will

need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

*FILELVL

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

*OWNER

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Restore to library (RSTLIB)

Specifies whether the library contents are restored to the same library in which they were saved, or to a different library.

*SAVLIB

The library contents are restored to the same library or libraries in which they were saved.

name Specify the name of the library where the saved library contents are restored. If *NONSYS, *IBM, or *ALLUSR is specified for the **Saved library (SAVLIB)** parameter, a library name cannot be specified for this parameter.

Note: If a Structured Query Language (SQL) database is restored to a library other than the one in which it was saved, the journals are not restored.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

*SAVASPDEV

The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Top

Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Recovering your system book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

*SAVASP

The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

*NONE

No output is created.

*PRINT

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Type of output information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

***OBJ** The list contains an entry for each object requested to be restored.

***MBR** The list contains an entry for each object, database file member, and spooled file requested to be restored.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

*ALL All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Examples

Example 1: Restoring New Objects

```
RSTLIB SAVLIB(JOE) DEV(TAP01) OPTION(*NEW)
```

This command restores the saved version of library JOE from tape device TAP01. The only objects that are restored in the library are new objects (ones that were in the library when they were saved and later deleted).

Example 2: Printing Output

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) OUTPUT(*PRINT)
DFRID(*DFT)
```

This command restores all the saved nonsystem libraries to the system from tape. The contents of the libraries are restored exactly as they were saved. New objects (on tape) are added to the system; old objects in the system are overlaid by the version of the old objects on tape. Because OUTPUT(*PRINT) is specified, a printout of all objects (restored and not restored) for each library, is sent to the printer with the job's spooled output. Each library after the first library starts on a new page. After each library, a completion message states how many objects were restored and how many were not restored. At the end of the printout, a list of any objects that were deferred is printed, and a final completion message states how many libraries were restored and how many were not restored.

Example 3: Specifying Where the Restore Operation Begins

```
RSTLIB SAVLIB(*NONSYS) DEV(TAP01) STRLIB(MIKESLIB)
```

This command restores the saved nonsystem libraries beginning with library MIKESLIB from the tape device, TAP01. The *first* tape of the nonsystem save must be loaded. An inquiry message instructs the user to load the tape containing MIKESLIB. If necessary, the same message is sent until the tape containing MIKESLIB is found.

Example 4: Restoring a Version From a Specific Date and Time

```
RSTLIB SAVLIB(PAYROLL) DEV(TAP01) SAVDATE(060193)
SAVTIME(103214) RSTLIB(OLDPAY) VOL(PAY)
```

This command restores the version of the PAYROLL library from the device TAP01, whose volume identifier is PAY. The version to be restored was saved at 10:32:14 on the date 06/01/93. All of the objects in the saved PAYROLL library are restored to the library OLDPAY. All new files are restored. Old files are restored only if the member lists of the files on the tape match the member lists of the files on the system.

Example 5: Restoring From Multiple Tape Volumes

```
RSTLIB SAVLIB(QGPL) DEV(TAP01) VOL(QGPL QGPL)
```

This command restores the QGPL library from two tape volumes both named QGPL. Even though the volume identifiers are the same, they must both be specified.

Example 6: Restoring From Multiple Tape Devices Serially

```
RSTLIB SAVLIB(USRLIB) DEV(TAP01 TAP02 TAP03)
VOL(USRA USRB USRC USRD) ENDOPT(*UNLOAD)
```

This command restores library USRLIB from four volumes on three tape devices. Volume USRA is put on tape device TAP01, volume USRB on TAP02, volume USRC on TAP03, and volume USRD on TAP01. The operator removes volume USRA from TAP01, so that TAP01 can be used by volume USRD. If the tape volumes are put in the wrong order, an error message is sent to the system operator message queue.

Example 7: Restoring From Multiple Devices in Parallel

```
RSTLIB SAVLIB(USRLIB) DEV(*MEDDFN) MEDDFN(LIBA/MEDDFNA)
```

This command restores library USRLIB using the devices and volumes specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the **Backup, Recovery, and Availability** topic in the Information Center.

Example 8: Restoring a Specific Version

```
RSTLIB SAVLIB(LIB1) DEV(TAP01) MBROPT(*ALL)
SAVDATE(082392) SAVTIME(123251)
RSTLIB(LIB2) OUTPUT(*PRINT)
```

This command restores the version of library LIB1 from the device TAP01. The version to be restored was saved at 12:32:51 on the date 08/23/92. All of the objects in the saved library LIB1 are restored to library LIB2. A list of restored objects and those not restored is given. All files and file members are restored.

Example 9: Restoring a Library From a Save File

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
```

This command restores library LIB1 from the save file SAVF1.

Example 10: Restoring to a Basic User ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1) RSTASP(2)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to basic user auxiliary storage pool (ASP) 2 unless:

- The library already exists in a different ASP.
- ASP 2 contains a journal, journal receiver, or SAVF which is part of a library in the system ASP.
- ASP 2 does not exist on the system.
- There are object types in the library which cannot be restored to user ASPs. These objects will not be restored.

Example 11: Restoring to an Independent ASP

```
RSTLIB SAVLIB(LIB1) DEV(*SAVF) SAVF(SAVF1)
RSTASPDEV(DIVISION1)
```

This command restores the library named LIB1 from the save file named SAVF1. The library and all objects in the saved version of LIB1 are restored to the independent ASP named DIVISION1 unless:

- The library already exists in the system ASP, a basic user ASP, or a different independent ASP in the same linked ASP group as DIVISION1.
- ASP DIVISION1 is not available on the system.
- There are object types in the library which cannot be restored to independent ASPs. These objects will not be restored.

Example 12: Restoring More Than One Library From Tape

```
RSTLIB SAVLIB(*ANY) DEV(TAP01) SEQNBR(75)
```

This command restores the first saved version of each library from tape device TAP01 starting at tape file sequence number 75.

Example 13: Restoring More Than One Library From Optical

```
RSTLIB SAVLIB(A* B*) DEV(OPT01) OPTFILE('/BACKUP/*')
```

This command restores all libraries whose names start with A or B that are found in directory BACKUP on the optical volume in device OPT01.

Example 14: Restoring Spooled Files

```
RSTLIB SAVLIB(JOE) DEV(TAP01) SPLFDTA(*NEW)
```

This command restores the saved version of library JOE from tape device TAP01. Any spooled files that do not currently exist on the system and that were saved with output queues in this library are restored.

[Top](#)

Error messages

*ESCAPE Messages

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3705

&2 &1 in &3 not journaled.

CPF3706

&2 &1 not restored to library &3.

CPF3707

Save file &1 in &2 contains no data.

CPF3709

Tape devices do not support same densities.

CPF372C

Library &1 not restored to ASP &2.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF373E
Library &1 not restored to ASP &2.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3732
&2 &1 status error during restore operation.

CPF3733
&2 &1 in &3 previously damaged.

CPF3738
Device &1 used for save or restore is damaged.

CPF3739
Database file &1 member in &3 damaged.

CPF3740
Object &1 type &2 in &3 not found.

CPF3743
File cannot be restored, displayed, or listed.

CPF3748
Object information for library &1 damaged.

CPF375F
File not selected. Cannot restore from save type file &1.

CPF3752
No record of save operation exists for library &1.

CPF3757
&2 &1 not restored to &3.

CPF3758
&2 &1 not restored to &3.

CPF376B
File &1 not found.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3769
Data encrypted or not save data.

CPF3770
No objects saved or restored for library &1.

CPF3773
&1 objects restored. &2 not restored to &4.

CPF3779
&1 libraries restored; &4 partially restored; &2 not restored.

CPF378B
Library &1 not created.

CPF3780
Specified file for library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3783
Cannot determine VOL(*SAVVOL) location. No objects restored.

CPF3784
Restore device specified in the DEV parameter does not match VOL(*SAVVOL) device.

CPF3785
Not all subsystems ended.

CPF3791
While processing &2 &1 in &3, encountered end of file &4.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

CPF37A5
RSTASP &1 is not online.

CPF37B8
Not authorized to restore private authorities.

CPF37C2
Not authorized to use DFRID.

CPF380E
User profile &1 not restored. Not able to create UID or GID.

CPF3805
Objects from save file &1 in &2 not restored.

CPF3807
Data decompression error for save file &1 in &2.

CPF3812
Save file &1 in &2 in use.

CPF3818
Starting library &1 not found.

CPF384F
&2 &1 not restored to library &3.

CPF3894
Cancel reply received for message &1.

CPF38A2
ASP device &1 not correct.

CPF38A6

Only one library allowed with specified parameters.

CPF5729

Not able to allocate object &1.

CPF9809

Library &1 cannot be accessed.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9829

Auxiliary storage pool &1 not found.

CPFB8ED

Device description &1 not correct for operation.

OPT1498

Volume name list exhausted on device &1.

OPT1502

Attempted to process past the end of a multi-volume set.

OPT1605

Media or device error occurred.

STATUS Messages*CPF3770**

No objects saved or restored for library &1.

CPF3773

&1 objects restored. &2 not restored to &4.

Top

Restore Licensed Program (RSTLICPGM)

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

Parameters
 Examples
 Error messages

The Restore Licensed Program (RSTLICPGM) command loads or restores a licensed program for initial installation, for new-release installation, or for recovery.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. To use this command, the user must have *SECADM and *ALLOBJ authority.
3. If this command is used to restore a program in the licensed program, the copy of the program currently in the system should not be running while the program is being restored. If this occurs, the processing program is ended abnormally.
4. If other objects of the licensed program are in use, they are not restored.
5. With the exception of overrides for the restore operation printing OUTPUT(*PRINT), this command ignores all file overrides currently in effect for the job.
6. Some licensed programs are restored only if the user is enrolled in the system distribution directory. See the publications for each licensed program for a description of this restriction.
7. This command does not restore code and language objects for the base operating system.
8. This command does not support the use of user ASPs (auxiliary storage pools). All objects supplied by a licensed program must remain in the system ASP.
9. At most one optical device, one save file, one virtual tape device or one tape media library device can be specified.

Top

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	Character value	Required, Positional 1
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): Name	Required, Positional 2
OPTION	Optional part to be restored	* <u>BASE</u> , 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 01, 02, 03, 04, 05, 06, 07, 08, 09	Optional
RSTOBJ	Type of object to be restored	* <u>ALL</u> , *PGM, *LNG	Optional
LNG	Language for licensed program	Character value, * <u>PRIMARY</u> , *SAVVOL	Optional
OUTPUT	Output	* <u>NONE</u> , *PRINT	Optional
RLS	Release	Character value, * <u>FIRST</u>	Optional
REPLACERLS	Replace release	Character value, * <u>ONLY</u> , *NO	Optional
VOL	Volume identifier	Single values: * <u>MOUNTED</u> Other values (up to 75 repetitions): Character value	Optional

Keyword	Description	Choices	Notes
SEQNBR	Sequence number	1-16777215, <u>*SEARCH</u>	Optional
ENDOPT	End of media option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
LIB	Library	<i>Element list</i>	Optional
	Element 1: Library 1	<i>Name</i> , <u>*SAME</u>	
	Element 2: Library 2	<i>Name</i> , <u>*SAME</u>	
	Element 3: Library 3	<i>Name</i> , <u>*SAME</u>	
	Element 4: Library 4	<i>Name</i> , <u>*SAME</u>	
	Element 5: Library 5	<i>Name</i> , <u>*SAME</u>	
	Element 6: Library 6	<i>Name</i> , <u>*SAME</u>	
	Element 7: Library 7	<i>Name</i> , <u>*SAME</u>	
	Element 8: Library 8	<i>Name</i> , <u>*SAME</u>	
	Element 9: Library 9	<i>Name</i> , <u>*SAME</u>	
	Element 10: Library 10	<i>Name</i> , <u>*SAME</u>	
	Element 11: Library 11	<i>Name</i> , <u>*SAME</u>	
LNLIB	Language library	<i>Name</i> , <u>*SAME</u>	Optional
FLR	Folder	<i>Name</i> , <u>*SAME</u>	Optional
CODHOMEDIR	Code home directory	Single values: *PROMPT Other values (up to 300 repetitions): <i>Path name</i> , *SAME	Optional
LNGHOMEDIR	Language home directory	Single values: *PROMPT Other values (up to 300 repetitions): <i>Path name</i> , *SAME	Optional
FRCOBJCVN	Force object conversion	Single values: <u>*SYSVAL</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	

Top

Product (LICPGM)

Specifies the seven-character identifier of the licensed program that is restored.

This is a required parameter.

Top

Device (DEV)

Specifies the name of the device used for the restore of the licensed program. The device name must already be known on the system by a device description. Use the Work with Device Descriptions (WRKDEV) command to display the names of the devices available on this system.

This is a required parameter.

The possible values are:

***SAVF** The restore operation is done using the save file name specified on the save file (SAVF) parameter.

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the name of one or more tape devices to use to restore the licensed program. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Top

Optional part to be restored (OPTION)

Specifies which one of the optional parts of the licensed program given in the **Product** prompt (LICPGM parameter) is to be restored.

The possible values are:

***BASE**

Only the base part of the licensed program is to be restored.

number-of-licensed-program-option

Specify the number of the optional part of the listed licensed program that is to be restored.

Top

Type of object to be restored (RSTOBJ)

Specifies the type of licensed program objects to be restored.

The possible values are:

***ALL** All of the objects for the licensed program are restored. This includes both program objects and the language objects specified on the **Language for licensed program** prompt (LNG parameter).

If a tape device is specified on the DEV parameter, then the RSTOBJ(*ALL) value is used when the saving of the licensed program has been done with the SAVLICPGM command such that the language objects immediately follow the program objects on the tape media. If the language objects (*LNG) and programming objects (*PGM) are not in consecutive order on the distribution tape, *ALL cannot be used in most cases. Instead, the program and language objects must be restored separately. The DSPTAP command can be used to determine the order of the objects on the tape. An example of how to restore language and program objects separately is in the "Examples" section at the end of this command.

If *SAVF is specified on the DEV parameter, then the RSTOBJ(*ALL) value can be used when the saving of the licensed program has been done with the SAVLICPGM command using OBJTYPE(*ALL).

***PGM** Only the program objects for the licensed program are restored. *PGM should be used when restoring program objects from a distribution media where the program objects and selected language objects are not on the same distribution media or are not in consecutive order.

*LNG Only the language objects for a licensed program are restored. The **Language for licensed program** prompt (LNG parameter) determines if the objects are associated with the licensed program library or with a multilingual library.

Top

Language for licensed program (LNG)

Specifies the national language version (NLV) to be used for restoring the licensed program. If the language feature of the licensed program on the save media matches the system language feature, the language objects are restored to the licensed program's libraries. If the language features do not match, the language objects are restored into the multilingual library for that language feature.

The possible values are:

*PRIMARY

The language feature of the operating system is restored for the specified licensed program.

Note: Use GO LICPGM with option 20 to display the primary language of the operating system.

*SAVVOL

The language file on the mounted volume is to be restored for the licensed program. This option is not valid with DEV(*SAVF).

feature-codes

Specify the NLV identifier for the language file that is to be restored for the licensed program. More information on feature identifications and a list of IBM-supplied feature codes is in the Installing, upgrading, or deleting i5/OS and related software book, SC41-5120.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created. The listing shows the parameter values that were specified and shows all objects, restored and not restored. Information about each object's security is listed for the restored objects.

The possible values are:

*NONE

No output is created.

*PRINT

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

Top

Release (RLS)

Specifies the version, release, and modification level of the licensed program being restored.

The possible values are:

*FIRST

The first version, release, and modification level found on the distribution media is restored.

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.

Top

Replace release (REPLACERLS)

Specifies the version, release, and modification level of the licensed program being replaced.

The possible values are:

*ONLY

Replace only the version, release, and modification level of the licensed program currently installed.

***NO** The licensed program currently installed on the system is not replaced. The licensed program being restored must be a different release than the one currently installed. If the licensed program being restored exists in the same libraries as the installed program, an override parameter must be specified indicating to which libraries the licensed program is restored.

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.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

The possible values are:

*MOUNTED

The licensed program is restored from the volumes that are currently on the device specified on the **Device** prompt (DEV 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.

Note: This value cannot be specified when using an optical media library device.

volume-identifier

Specify the identifiers of one or more volumes in the order they are put on the device and used to restore the licensed program. A maximum of 75 volume identifiers can be specified.

Top

Sequence number (SEQNBR)

Specifies which sequence number to use for the restore process. This option is only valid for tape devices.

The possible values are:

*SEARCH

The volume that is placed in the device is searched for a data file with an identifier that matches the label for the specified licensed program or licensed program option. When a match is found,

the objects are restored. If the last operation on the device specified ***LEAVE** on the **End of media option** prompt (ENDOPT parameter), indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If ***LEAVE** was not used for the **End of media option** prompt (ENDOPT parameter) of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

file-sequence-number

Specify the sequence number of the file that is used for the restore process. Valid values range from 1 through 16777215.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

***UNLOAD**

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

Top

Save file (SAVF)

Specifies the qualified name of the save file containing the product.

The name of the save file can be qualified by one of the following library values:

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

***CURLIB**

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

library-name

Specify the name of the library where the save file exists.

The possible values are:

save-file-name

Specify the name of the save file.

Top

Library (LIB)

Specifies the libraries into which the licensed program is being restored. This function is not supported by all licensed programs.

The possible values are:

***SAME**

The licensed program is restored into the specified library.

library-name

Specify the name of the library into which the licensed program is being restored. A maximum of 11 libraries can be specified.

Top

Language library (LNLIB)

Specifies the secondary language library into which the secondary language objects are restored. This function is not supported by all licensed programs.

The possible values are:

***SAME**

The licensed program is restored into the specified secondary language library.

library-name

Specify the name of the secondary language library into which the licensed program is restored.

Top

Folder (FLR)

Specifies the name of the root folder into which the licensed program is being restored. This function is not supported by all licensed program.

The possible values are:

***SAME**

Use the specified root folder.

folder-name

Specify the name of the root folder. The root folder is the folder on the system containing all of the other folders.

Top

Code home directory (CODHOMEDIR)

Specifies the directories into which the code part of the product is being restored. This function is not supported by all products.

Note: This parameter is mutually exclusive with the FLR parameter.

The possible values are:

***SAME**

The code part of the product is restored into the directories specified when packaged or already

installed. *SAME may be specified as the only parameter value or within a list of directories. If used within a list, *SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

***PROMPT**

The code directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

path-name

Specify the home path directory name into which the code part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, *SAME can be specified for the path name.

Top

Language home directory (LNGHOMEDIR)

Specifies the directories into which the language part of the product is being restored. This function is not supported by all products.

Note: This parameter is mutually exclusive with the FLR parameter.

The possible values are:

***SAME**

The language part of the product is restored into the directories specified when packaged or already installed. *SAME may be specified as the only parameter value or within a list of directories. If used within a list, *SAME specifies that a particular directory is unchanged, though other directories may be different than when the product was packaged or previously installed.

***PROMPT**

The language directories to be used are displayed. If the product is not currently installed, the directory names can be changed.

path-name

Specify the home path directory name into which the language part of the product is being restored. Up to 300 directories may be specified. For directory name entries which are unchanged, *SAME can be specified for the path name.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Examples

Example 1: Restoring Program Using Defaults

```
RSTLICPGM LICPGM(5761WDS) DEV(TAP01)
```

This command restores the WebSphere Development Studio (5761WDS) licensed program to the system. The tape containing the licensed program objects must be put on the TAP01 tape drive. Because no other parameters are specified, the defaults are used for the command.

Example 2: Restoring a Third Version of a Product

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE)
DEV(TAP01) RLS(V5R2M0)
REPLACERLS(*NO) LIB(A B C)
```

This command restores the base part of the V5R2M0 1MYPROD product to the system if the base of the V5R2M0 1MYPROD product is not currently installed on the system.

Example 3: Restoring One Version of a Product Over Another Version

```
RSTLICPGM LICPGM(2MYPROD) OPTION(*BASE)
DEV(TAP01) RLS(*FIRST)
REPLACERLS(*ONLY)
```

This command restores the first version release modification level of the base part of the 2MYPROD product that is found on the tape in the TAP01 drive. It does not matter if the version release modification level of the base of the product on the tape matches the version release modification level of the base of the product on the system.

Example 4: Restoring Product From Save File

```
RSTLICPGM LICPGM(5761WDS) DEV(*SAVF) SAVF(MYLIB/MYSAVF)
```

This command restores the WebSphere Development Studio (5761WDS) licensed program to the system from the save file MYSAVF in MYLIB. Because no other parameters are specified, the defaults are used for the command.

Example 5: Restoring a Third Version of a Product From a Save File

```
RSTLICPGM LICPGM(1MYPROD) OPTION(*BASE)
          DEV(*SAVF) RLS(V5R2M0)
          REPLACERLS(*NO) LIB(A B C) SAVF(MYLIB/MYSAVF)
```

This command restores the base part of the V5R2M0 1MYPROD product to the system from save file MYSAVF in MYLIB if the base of the V5R2M0 1MYPROD product is not currently installed on the system.

Top

Error messages

*ESCAPE Messages

CPF3D96

Objects for product &1 option &2 release &4 not restored.

Top

Restore Object (RSTOBJ)

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

Parameters
Examples
Error messages

The Restore Object (RSTOBJ) command restores to the system a single object or a group of objects in a single library that were saved on a tape or optical volume, or in a save file using a single command. Objects to be restored could have been saved by the Save Object (SAVOBJ), Save Changed Objects (SAVCHGOBJ) or Save Library (SAVLIB) command. The RSTOBJ command restores the object description and contents of each object specified in the command.

For job queues, user-defined message queues, logical files, and user queues, only the object descriptions are restored; the contents of those objects are not restored. However, logical file access paths can be saved by a save command with ACCPTH(*YES) specified; if this is done, the access paths can be restored. More information on restoring access paths is in the Database category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. The contents of spooled files on output queues can be restored, if they were saved, by specifying *NEW for the **Spooled file data (SPLFDTA)** parameter.

The command can be used to restore the objects even if the object storage was freed when the objects were saved, or if the objects were deleted after they were saved. If the storage was not freed as part of a save operation, each object is restored in the same area of storage that it previously occupied. If the version of the object being restored is larger than the version in the system (for example, data records that were deleted from the system still exist in the saved version of a file), the additional storage needed for the object is acquired. If the saved version of the object is smaller (for example, data records that are added to the system), the space that was acquired for the object remains assigned to that object and available for use by that object.

If logical file access paths were saved (ACCPTH(*YES) was specified when the objects were saved), the access paths are restored if (1) all based-on physical files are also being restored by the same restore command, (2) the logical file is also being restored by the same restore command, or the logical file already exists on the system (the same file exists, not a re-created version), and (3) MAINT(*IMMED or *DLY) is in effect for the logical file if it still exists on the system.

If the storage was freed, the system finds the storage space needed to store the contents (only the data portion) of each file, module, program, service program, journal receiver, and Structured Query Language (SQL) package. If the objects do not exist on the system because they were deleted or they are being restored in a different system, the system must find the storage to store everything (the description and the data portion) about each unknown object.

If an object is being restored over an existing object on the system, the owner, primary group, public and private authorities, authorization list, and object auditing value of the existing object are retained. The **Allow object differences (ALWOBJDIF)** parameter may affect whether the object is restored, and the **Private authorities (PVTAUT)** parameter may add saved private authorities to the existing private authorities. Existing objects that are in use, such as output queues that are actively spooling, are not restored. Existing data queues are not restored.

If an object is being restored as new to the system, the saved owner name, primary group name, public authority, and object auditing value are used. The ALWOBJDIF parameter may affect whether the saved authorization list name is used, and the PVTAUT parameter may restore saved private authorities. If the owner user profile does not exist, the system default owner (QDFTOWN) becomes the owner of the object. If the primary group does not exist, *NONE is used.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

ATTENTION: Do not use this command to restore licensed programs into library QSYS. Unpredictable results can occur.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must have use (*USE) authority for the Create Save File (CRTSAVF) command when restoring a save file that does not currently exist on the system.
- When saving or restoring to an existing database file using the OUTFILE parameter, you must have execute (*EXECUTE) authority to the library where the output database file is located.
- You must have save system (*SAVSYS) special authority, or all of the following object authorities:
 - Add (*ADD) and execute (*EXECUTE) authorities for the specified library and save file library.
 - Object existence (*OBJEXIST) authority for, or be the owner of, each object specified if the object already exists in the library on the system. *OBJEXIST and use (*USE) authorities are required for message queue objects. If the object does not already exist in the library on the system, you must have *ADD authority for the user profiles that own each object being restored. For spooled file data, *OBJEXIST authority is required for the output queue to which it is restored.
 - If VOL(*SAVVOL) is specified, *USE authority to the saved-from library.
- The RSTOBJ command does not restore the library’s data dictionary or its associated database files. To do this, the RSTLIB command must be used.
- You must have *USE authority for the save file to restore from the save file. In addition, the user must have *USE authority for the device description when restoring from a tape or optical device.
- When using a media definition, you must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
- If this command is used to restore an OPM program, the copy of that program that is currently in the system must not be running while the program is being restored. If this occurs, the running program may fail or behave unpredictably.
- Objects saved by separate commands must also be restored by separate commands. If a single command is used, some of the objects are not restored.
- If the user domain object user space (*USRSPC), user index (*USRIDX), or user queue (*USRQ) is restored to a library that is not permitted in the system value QALWUSRDMN (allow user domain objects in libraries), the object is converted to a system domain object.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
SAVLIB	Saved library	Single values: *ANY Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
DEV	Device	Single values: *SAVE, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3

Keyword	Description	Choices	Notes
OBJTYPE	Object types	Single values: *ALL Other values (up to 74 repetitions): *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NWSCFG, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDL0D, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *TIMZON, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	Optional, Positional 4
VOL	Volume identifier	Single values: *MOUNTED , *SAVVOL Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 5
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional, Positional 6
LABEL	Label	<i>Character value</i> , *SAVLIB	Optional
ENDOPT	End of media option	*REWIND , *LEAVE, *UNLOAD	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	
OPTION	Option	*ALL , *NEW, *OLD, *FREE	Optional
FILEMBR	File member	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: File	<i>Name</i> , *ALL	
	Element 2: Member	Single values: *ALL , *NONE Other values (up to 50 repetitions): <i>Generic name, name</i>	
MBROPT	Data base member option	*MATCH , *ALL, *NEW, *OLD	Optional
DFRID	Defer ID	<i>Name</i> , *NONE	Optional
SPLFDTA	Spooled file data	*NEW , *NONE	Optional
PVTAUT	Private authorities	*NO , *YES	Optional
SAVDATE	Date when saved	<i>Date</i>	Optional
SAVTIME	Time when saved	<i>Time</i>	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE , *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: *SYSVAL , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	*YES	
	Element 2: Objects to convert	*RQD , *ALL	
RSTLIB	Restore to library	<i>Name</i> , *SAVLIB	Optional
OUTPUT	Output	*NONE , *PRINT, *OUTFILE	Optional
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL , *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
RSTASPDEV	Restore to ASP device	<i>Name</i> , *SAVASPDEV	Optional
RSTASP	Restore to ASP number	1-32, *SAVASP	Optional

Keyword	Description	Choices	Notes
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	
INFTYPE	Type of output information	<i>*OBJ, *MBR</i>	Optional
OMITLIB	Libraries to omit	Single values: <i>*NONE</i> Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *NONE, *ALL</i>	
	Qualifier 2: Library	<i>Generic name, name, *ALL</i>	
	Element 2: Object type	<i>Character value, *ALL</i>	

Top

Objects (OBJ)

Specifies the names of one or more objects to restore. The objects you specify will be restored from the first file that meets the search value and contains any of the objects. If a tape contains multiple files for the same library, you may need to specify the tape file sequence number to ensure that all of the correct objects are restored.

If the file does not contain all of the objects you specify, diagnostic messages will be issued for the objects not found. The completion message will contain a count of objects restored and objects not restored.

If the **Object types (OBJTYPE)** parameter is not specified when the command is run, all the object types listed in the description of that parameter are restored, if they are in the specified library on the media or in the save file, and if they have the specified names.

This is a required parameter.

Single values

***ALL** All the objects saved from the specified library are restored, depending on the values specified for the **Object types (OBJTYPE)** and **Option (OPTION)** parameters.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of objects in the specified library to restore. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk is not specified with the name, the system assumes that the name is a complete object name.

name Specify one or more names of specific objects to restore. Both generic names and specific names can be specified in the same command.

Top

Saved library (SAVLIB)

Specifies the libraries that contained the saved objects. If *SAVLIB is specified or defaulted for the **Restore to library (RSTLIB)** parameter, this is also the name of the library to which the objects are restored.

Notes:

1. If you specify more than one library, or a single value, or a generic library value:
 - The **Device (DEV)** parameter must not specify *SAVF.
 - The **Restore to library (RSTLIB)** parameter must specify *SAVLIB.
 - The **Label (LABEL)** parameter must specify *SAVLIB.
 - The **Optical file (OPTFILE)** parameter must specify * or end with /*.
2. The tape or optical file that you select to restore may contain a library that does not match the value that you specify for this parameter. This parameter is used to determine the default LABEL value for a tape file and the default OPTFILE value for an optical file, but the restore operation does not verify that this parameter matches the saved library.

This is a required parameter.

Single values

***ANY** Restores objects from the first version of all saved libraries found on the tape beginning with the sequence number specified for the **Sequence number (SEQNBR)** parameter, or restores objects from all saved libraries found on the optical media in the directory specified for the OPTFILE parameter.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be restored.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library to restore. The name of the library being restored must be the same as the name that was used when the library was saved.

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

***MEDDFN**

The restore operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

Object types (OBJTYPE)

Specifies the types of system objects that are restored.

Single values

***ALL** All object types that are specified by name and were saved from the specified library are restored. If *ALL is also specified for the **Objects (OBJ)** parameter, all objects saved for that library are restored.

Other values

object-type

Specify the value for each of the types of objects to be restored, such as command (*CMD), file (*FILE), or program (*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Note: *CFGL object types can be restored using the Restore Object (RSTOBJ) command only from objects saved on releases prior to V2R2M0. *CFGL object types saved on V2R2M0 and newer releases are restored using the Restore Configuration (RSTCFG) command.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Note: The version of the objects restored is the first version found in the specified location, unless a specific version is identified by the **Date when saved (SAVDATE)** and **Time when saved (SAVTIME)** parameters.

Single values

***MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)**

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.

Note: This value cannot be specified when using an optical media library device.

***SAVVOL**

The system, by using the save or restore history information, determines which volumes contain the most recently saved version of the objects.

Note: If this is a restore from an optical device, only the first 6 characters of the volume name are available. If the volume name of the optical media exceeds 6 characters, the volume may not be found. You should specify the complete volume or file name on the command instead of using VOL(*SAVVOL) when the name of the optical media exceeds 6 characters.

When *SAVVOL is specified, the following operational characteristics and restrictions apply:

- If the characteristics of the device specified in the DEV parameter do not match the device and location of the most recently saved version of the library, an error message is sent to the **user**, and the function is ended.
- If the wrong volume is placed in a device in the location specified by the command, a message is sent to the **system operator** that identifies the first volume that must be placed in the device before the objects can be restored.
- When OBJ(*ALL) and OBJTYPE(*ALL) is specified, the objects are considered in the order they would appear in a display produced by the Display Library (DSPLIB) command. The object names and types specified in the RSTOBJ command are used to determine which file of saved objects is used in the restore operation. One file is produced for each SAVLIB or SAVOBJ command run. The file chosen is the one in which the first considered object was last saved. Objects that were not saved in the file chosen to be processed, or that were more recently saved, are not restored; an error message is sent to the user for each unrestored object.
- If *SAVVOL is specified, the SAVDATE and SAVTIME parameters cannot be specified.
- If *SAVVOL is specified and the RSTLIB value is equal to the SAVLIB value, OPTION(*NEW) cannot be specified.
- If *SAVVOL is specified, SEQNBR(*SEARCH) and LABEL(*SAVLIB) must be specified.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

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

***SEARCH**

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified *LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If *LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

Label (LABEL)

Specifies the name that identifies the data file to be used for the restore operation. This label must have been specified on the save command.

*SAVLIB

The file label is the name specified for the **Saved library (SAVLIB)** parameter.

character-value

Specify the data file identifier of the data file used for the restore operation. A maximum of 17 characters can be used. This option is valid only for a single-library restore.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Option (OPTION)

Specifies how to handle restoring each object.

- *ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.
- *NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.
- *OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- *FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

File member (FILEMBR)

Specifies the database file members to restore. This parameter is made up of two parts: the file name and the member name. This parameter cannot be specified if *MATCH is specified for the **Data base member option (MBROPT)** parameter.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by a specific name, a generic name, or *ALL. The **Object types (OBJTYPE)** parameter must either be *ALL or include *FILE. Generic names are not valid for the database file name but are allowed for the member name. Duplicate file names are not allowed.

There may be up to 50 of the file/member element list combinations specified for this parameter.

Element 1: File

***ALL** The list of member name values that follows this value applies to all files indicated on the OBJ parameter.

name Specify the name of the database file from which the listed members are restored.

Element 2: Member

Single values

***ALL** All members are restored from the specified file.

***NONE**

No members are restored from the specified file. Only the file description is restored.

Other values (up to 50 repetitions)

generic-name

Specify the generic name of the members to be restored from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk is not specified with the name, the system assumes that the name is a complete member name.

name Specify the name of the member to be restored from the specified file.

Top

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

*ALL All members in the saved file are restored.

*NEW Only new members (members not known to the system) are restored.

*OLD Only members already known to the system are restored.

Top

Defer ID (DFRID)

Specifies the identifier to be used if you want to defer the restore of objects that depend on other objects that are not yet available. To complete the restore of deferred objects, restore the objects that they depend on, and specify the same Defer ID. If any objects remain deferred when the other objects are available, use the Restore Deferred Objects (RSTDFROBJ) command, and specify the same Defer ID. This parameter allows you to restore all objects in a set of libraries when the libraries with dependent objects are restored before the libraries with the objects they depend on.

Deferred objects can be logical files or SQL materialized query tables (MQTs). A deferred logical file is not created until the restore is complete. A deferred MQT is created, but until the restore is complete, any functions performed on the MQT that require access to the based-on files will fail.

Notes:

1. If the following conditions are true, the restore of a deferred object may be completed automatically when the objects it depends on are restored:
 - The deferred object is restored to the same library from which it was saved.
 - The same Defer ID is specified for the restore operations for both the deferred object and the objects it depends on.

*NONE

Objects will not be restored or deferred if they depend on other objects that are not available.

name Specify an identifier to defer the restore of objects that depend on other objects that are not yet available. You must have save system (*SAVSYS) special authority to specify a name.

Top

Spooled file data (SPLFDTA)

Specifies whether to restore spooled file data and attributes.

***NEW** For each output queue that is restored, spooled file data that was saved with the output queue is restored, if it does not already exist on the system.

***NONE** No spooled file data is restored.

Top

Private authorities (PVTAUT)

Specifies whether to restore private authorities with the objects that are restored.

***NO** No private authorities are restored.

***YES** Private authorities are restored with the objects. Objects will be restored only from save operations that specified that private authorities should be saved with the objects.

Note: You must have all object (*ALLOBJ) special authority to specify this value.

Top

Date when saved (SAVDATE)

Specifies the date when the objects were saved. If the most recently saved version is not the one being restored, or if multiple saved versions exist on the media volume, specify the date that indicates which version of the objects to restore. If a volume identifier or *MOUNTED is specified for the **Volume identifier (VOL)** parameter, but SAVDATE is not, the version of the objects that is restored is the first version found on the volume. This parameter is valid only if a volume identifier or *MOUNTED is specified for the VOL parameter or if a save file is specified for the **Save file (SAVF)** parameter. This parameter is ignored when the **Sequence number (SEQNBR)** parameter is specified.

date Specify the date that the library to be restored was saved. The date must be entered in the job date format.

Top

Time when saved (SAVTIME)

Specifies the time at which the objects were saved.

If a volume identifier or *MOUNTED is specified for the **Volume identifier (VOL)** parameter, but this parameter is not specified, the version of the objects to be restored is the first version found on the volume. This parameter is valid only if the **Date when saved (SAVDATE)** parameter is also specified.

This parameter is ignored when a number is specified for the **Sequence number (SEQNBR)** parameter.

time Specify the time that the library to be restored was saved. The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from

the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- Authorization list: The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.
Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and

the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***FILELVL**

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP**

Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

***SYSVAL**

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If `FRCOBJCVN(*YES *RQD)` is specified, then the `QFRCCVNRST` system value must have a value of "0", "1", or "2". `FRCOBJCVN(*YES *RQD)` will override a `QFRCCVNRST` value of "0" or "1". If `FRCOBJCVN(*YES *ALL)` is specified, then `QFRCCVNRST` can have any valid value and `FRCOBJCVN(*YES *ALL)` overrides the `QFRCCVNRST` system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Restore to library (RSTLIB)

Specifies whether the objects are restored to a different library or to the same library where they were saved.

***SAVLIB**

The objects are restored to the same library from which they were saved.

name Specify the name of the library to which the saved objects are restored.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

***NONE**

No output is created.

***PRINT**

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

***OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the `OUTFILE` parameter when `OUTPUT(*OUTFILE)` is specified.

Top

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to restore the data. For information about creating and using a media definition, see the Recovering your system book, SC41-5304, and the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

Qualifier 1: Media definition

name Specify the name of the media definition to be used.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

***** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'optical-directory-path-name'

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

***SAVASPDEV**

The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Top

Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Recovering your system book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

*SAVASP

The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

*LIBL The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

*CURLIB

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Type of output information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

***OBJ** The list contains an entry for each object requested to be restored.

***MBR** The list contains an entry for each object, database file member, and spooled file requested to be restored.

Top

Libraries to omit (OMITLIB)

Specifies a list of libraries to be excluded from the restore operation.

Single values

*NONE

No libraries are excluded from the restore operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

library-name

Specify the name of the library to be excluded from the restore operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

*ALL All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Examples

Example 1: Restoring Most Recently Saved Version

```
RSTOBJ OBJ(PAYROLL) SAVLIB(LIBX) DEV(TAP01)
OBJTYPE(*PGM) VOL(*SAVVOL)
```

This command restores to LIBX the program named PAYROLL that was saved from LIBX. The tape drive named TAP01 is used to restore the most recently saved version of the program.

Example 2: Specifying Save Date and Time

```
RSTOBJ OBJ(PAY*) SAVLIB(LIBX) DEV(OPT01) VOL(ABCD)
OPTION(*OLD) SAVDATE(102202)
SAVTIME(143000) RSTLIB(LIBY)
```

All objects whose names start with PAY and that were saved from the library named LIBX on optical volume ABCD at 14:30:00 on 10/22/02 are restored to LIBY. Volume ABCD must be put on the optical device named OPT01. Because OPTION(*OLD) is specified, the only objects restored are those having the same object name and type both in LIBY on the system and in LIBX on the optical volume.

Example 3: Adding a New Program to the QGPL Library

```
RSTOBJ OBJ(NEWPROG) SAVLIB(QGPL) DEV(OPT01) OBJTYPE(*PGM)
VOL(PGMS) OPTION(*NEW) ALWOBJDIF(*AUTL)
```

A new program named NEWPROG is added to the general purpose library, QGPL. It is restored from a volume labeled PGMS that is inserted in the optical device named OPT01. If the object was saved on a different system and was linked to an authorization list, it will be linked to the authorization list with the same name on this system.

Example 4: Printing An Output List

```
RSTOBJ OBJ(*ALL) SAVLIB(LIB) DEV(TAP01) OBJTYPE(*PGM)
VOL(SVOL) SEQNBR(2) SAVDATE(082392)
SAVTIME(143000) OUTPUT(*PRINT)
```

All programs that were saved from the library named LIB that exist in the saved version on the tape volume named SVOL, sequence number 2, saved at 14:30:00 on 08/23/92, are restored to the library named LIB. An output list of all objects restored as well as objects not restored is provided.

Example 5: Restoring Journal Receivers

```
RSTOBJ OBJ(*ALL) SAVLIB(BACKUP) DEV(*SAVF)
OBJTYPE(*JRNRCV) SAVF(SAVEJ) RSTASP(3)
```

All journal receivers that were saved from the library named BACKUP into the save file named SAVEJ are restored to the library named BACKUP. The journal receivers are restored to basic user ASP 3 (unless they already exist in the library named BACKUP and are in a different ASP or ASP 3 contains a library).

Top

Error messages

*ESCAPE Messages

CPF370C

Not authorized to ALWOBJDIF parameter.

CPF3705

&2 &1 in &3 not journaled.

CPF3706

&2 &1 not restored to library &3.

CPF3707

Save file &1 in &2 contains no data.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3730

Not authorized to &2 &1 in library &3.

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3738

Device &1 used for save or restore is damaged.

CPF3739

Database file &1 member in &3 damaged.

CPF374C

No objects restored to ASP &2.

CPF3743

File cannot be restored, displayed, or listed.

CPF3748

Object information for library &1 damaged.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF3769

Data encrypted or not save data.

CPF3770

No objects saved or restored for library &1.

CPF3773

&1 objects restored. &2 not restored to &4.

CPF3780

Specified file for library &1 not found.

CPF3781
Library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3783
Cannot determine VOL(*SAVVOL) location. No objects restored.

CPF3784
Restore device specified in the DEV parameter does not match VOL(*SAVVOL) device.

CPF3791
While processing &2 &1 in &3, encountered end of file &4.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

CPF37A5
RSTASP &1 is not online.

CPF37B8
Not authorized to restore private authorities.

CPF37C2
Not authorized to use DFRID.

CPF3805
Objects from save file &1 in &2 not restored.

CPF3807
Data decompression error for save file &1 in &2.

CPF3812
Save file &1 in &2 in use.

CPF384F
&2 &1 not restored to library &3.

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF3872
&1 objects restored. &2 not restored to &4.

CPF38A2
ASP device &1 not correct.

CPF38A6
Only one library allowed with specified parameters.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9825
Not authorized to device &1.

CPF9829
Auxiliary storage pool &1 not found.

CPFB8ED
Device description &1 not correct for operation.

OPT1498
Volume name list exhausted on device &1.

OPT1502
Attempted to process past the end of a multi-volume set.

OPT1605
Media or device error occurred.

Top

Restore Performance Collection (RSTPFRCOL)

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

Parameters
Examples
Error messages

The Restore Performance Collection (RSTPFRCOL) command restores to the system a performance collection or a group of performance collections in a single library that were saved in a save file with the Save Performance Collection (SAVPFRCOL) command.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must have add (*ADD) and execute (*EXECUTE) authorities for the specified library and save file library.

Top

Parameters

Keyword	Description	Choices	Notes
COL	Collection	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
SAVLIB	Saved library	<i>Name</i> , <u>QPFRDATA</u>	Optional
COLTYPE	Collection type	Single values: *ALL Other values (up to 10 repetitions): <i>Character value</i> , <u>*CSFILE</u>	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	
RSTLIB	Restore to library	<i>Name</i> , <u>*SAVLIB</u>	Optional

Top

Collection (COL)

Specifies the names of one or more performance collections to restore.

If the save file does not contain all of the performance collections you specify, diagnostic messages will be issued for the performance collections not found. The completion message will contain a count of performance collections restored and performance collections not restored.

If the **Collection type (COLTYPE)** parameter is not specified when the command is run, all the collection types listed in the description of that parameter are restored, if they are in the specified library on the save file, and if they have the specified names.

This is a required parameter.

Single values

***ALL** All the performance collections saved from the specified library are restored, depending on the values specified for the **Collection type (COLTYPE)** and **Option (OPTION)** parameters.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of performance collections in the specified library to restore. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk is not specified with the name, the system assumes that the name is a complete performance collection name.

name Specify one or more names of specific performance collections to restore. Both generic names and specific names can be specified in the same command.

Top

Saved library (SAVLIB)

Specifies the library that contained the saved performance collections. If *SAVLIB is specified or defaulted for the **Restore to library (RSTLIB)** parameter, this is also the name of the library to which the performance collections are restored.

QPFRRDATA

The saved performance collections were located in the IBM-supplied performance library, QPFRRDATA.

name Specify the name of the library to restore. The name of the library being restored must be the same as the name that was used when the library was saved.

Top

Collection type (COLTYPE)

Specifies the type of collections to be restored.

Single values

***ALL** All types of performance collections with the same name as the one located in the **Collection (COL)** parameter are to be restored. This includes file-based collections and object-based collections.

Other values (up to 10 repetitions)

*CSFILE

Only the Collection Services file-based collections specified in the **Collection (COL)** parameter are to be restored.

*CSMGTCOL

Only the Collection Services object-based collections specified in the **Collection (COL)** parameter are to be restored.

*DWFIL

Only the Disk Watcher file-based collections specified in the **Collection (COL)** parameter are to be restored.

*JWFIL

Only the Job Watcher file-based collections specified in the **Collection (COL)** parameter are to be restored.

*PEXFILE

Only the Performance Explorer file-based collections specified in the **Collection (COL)** parameter are to be restored.

*PEXMGTCOL

Only the Performance Explorer object-based collections specified in the **Collection (COL)** parameter are to be restored.

type Specify the type of collections to be restored.

Valid values depend on the performance collections supported by the system. You can press F4 while prompting this command parameter to see a list of valid collection type values.

Top

Save file (SAVF)

Specifies the save file used to restore the performance collections. This is a required parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Restore to library (RSTLIB)

Specifies whether the performance collections are restored to a different library or to the same library where they were saved.

*SAVLIB

The performance collections are restored to the same library from which they were saved.

name Specify the name of the library to which the saved performance collections are restored.

Top

Examples

Example 1: Restoring all types of Performance Collections

```
RSTPFRCOL COL(PFRCOL) COLTYPE(*ALL)
          SAVF(QGPL/MYSAVF)
```

This command restores to the IBM-supplied performance data library, QPFRDATA, all the performance collections with name PFRCOL. The save file MYSAVF in library QGPL is used to restore to the system the performance collection.

Error messages

*ESCAPE Messages

CPF3706

&2 &1 not restored to library &3.

CPF3707

Save file &1 in &2 contains no data.

CPF3730

Not authorized to &2 &1 in library &3.

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3739

Database file &1 member in &3 damaged.

CPF3743

File cannot be restored, displayed, or listed.

CPF3748

Object information for library &1 damaged.

CPF3770

No objects saved or restored for library &1.

CPF3773

&1 objects restored. &2 not restored to &4.

CPF3780

Specified file for library &1 not found.

CPF3781

Library &1 not found.

CPF3782

File &1 in &2 not a save file.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3796

Storage limit exceeded for user profile &4.

CPF37B8

Not authorized to restore private authorities.

CPF3805

Objects from save file &1 in &2 not restored.

CPF3812

Save file &1 in &2 in use.

CPF384F

&2 &1 not restored to library &3.

Restore S/36 File (RSTS36F)

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

Parameters
Examples
Error messages

The Restore System/36 File (RSTS36F) command restores to the system a single file or a group of files from a save all set. A single file can be restored from a database physical file, a diskette file or a tape file. A group of files can be restored from a save all set on diskette or tape.

The restore operation reads the description of the file from the database physical file, diskette, or tape, creates a database physical or logical file and copies any data from the database physical file, diskette file or tape file into the new database file. The database physical file must have been created with the Save System/36 File (SAVS36F) command. The diskette or tape file may have been created on a System/36 using the SAVE system operator control language (OCL) procedure (or the equivalent OCL use of the \$COPY SSP utility), or by using the Save System/36 File (SAVS36F) command. The Restore System/36 File (RSTS36F) command accepts diskette or tape files created on a System/34 or System/32 using the \$COPY utility.

The RSTS36F command accepts a diskette file created as a compressed file.

If the file being restored does not exist in the library specified on the TOLIB parameter, it is created. A physical file member is added using the name syntax 'Myymmdd', which identifies the original creation date of the file. This naming convention is needed by the System/36 environment in order to support date-differentiated files.

If a file name is specified for the TOFILE parameter, the name must meet the i5/OS naming standards. For more information about i5/OS naming conventions, see Chapter 2, "Control Language Syntax", in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If TOFILE(*SET) is specified, the files that are restored may have names that contain characters not allowed in an i5/OS simple object name. In this case, the file name is changed to an i5/OS extended name and the file is restored.

If the name contains a blank, a single quotation mark, a double quotation mark, an asterisk, a question mark, or a device control character (hexadecimal 00 through 3F or hexadecimal FF), the invalid characters are replaced with underlines. The file is then restored using the resulting simple or extended name; for example, A_? would become "A_"). If a file already exists with this new name, it is handled like any other name (see the MBROPT parameter).

If a file name is changed because of invalid characters, an informational message (CPF 2C1F) is sent to the recursion level above the program that is running this command. If the name is changed from a simple name to an extended name, no message is sent.

If the restore function creates the file and the file was not previously secured, the new file is owned by the user issuing the RSTS36F command and the file is created with a default authority of *ALL (that is the same as AUT(*ALL)).

If the file was saved from the S/36 where the attributes were an extend value of zero or no value specified, then a default value of 32 767 divided by the record length is assigned. If an extend value of

zero is required, use the change physical file (CHGPF) command (after the restore is completed) to set SIZE(*EXTEND) to zero. If the file was saved using i5/OS, the file is restored and the extend value does not change.

Note: This function is intended only for exchanging files with a System/36.

Restrictions:

1. This command is shipped with public *EXCLUDE authority.
2. The following authorities are required when running on a system using resource security:
 - *USE, *SECADM, and *ALLOBJ authorities for this command
 - *USE authority for the library specified in the TOLIB parameter
 - *CHANGE authority for the library specified in the TOLIB parameter when restoring a file that does not already exist on the system
 - *CHANGE and *OBJMGMNT authority for the existing file (needed to add a new member) when restoring a date-differentiated physical file and a file by the same name but with a different creation date
 - *ALL authority for the file when restoring to an existing physical file with the same creation date and MBROPT(*REPLACE) specified
 - *CHANGE authority for the based-on physical file (this physical file was referred to as the *parent file* on System/36) if the file being restored is a System/36 alternative index file (that is, a logical file)
 - *USE authority for the diskette device description object and *USE authority for device file QSYSDKT in library QSYS, when restoring from diskette
 - *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS, when restoring from tape
 - *USE authority for the file and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
 - If the file doesn't exist on the system but a file authorization holder object by the same name does exist, *ALL authority or ownership for the authorization holder object
3. There is no replace function supported when restoring System/36 alternative index files (logical files). If restoring an alternative index file, no file object by the same name can already exist in the specified library.
4. If restoring a logical file, the based-on physical file must already exist in the library specified for the TOLIB parameter.
5. The i5/OS files that are the same as the System/36 date-differentiated files are multiple-member physical files. Date-differentiated alternative index files are not supported. The data for a physical file is stored in a member that is named using the syntax 'Myymmdd' where 'yymmdd' represents the original creation date (in year/month/day format) of the file.

Because all members of a physical file share the same file attributes (for example, record length and key information), date-differentiated files with the same name that are restored to the same library are required to have the same file attributes. If an attribute mismatch is present, the files are not restored.
6. Object-level and record-level functions, including read operations, should not be used for a file being restored by the RSTS36F command. If another operation is being done at the same time on the file (for example, moving the file or reading or adding records), the restore operation stops if it cannot allocate the file.

Top

Parameters

Keyword	Description	Choices	Notes
TOFILE	To file	<i>Name</i> , *SET	Required, Positional 1
TOLIB	To library	<i>Name</i>	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
SET	Set identifier	<i>Character value</i> , # <u>SAVE</u>	Optional
IGCDTA	User specified DBCS data	* <u>NO</u> , *YES	Optional
FROMLABEL	File label	<i>Character value</i>	Optional
CRTDATE	Creation date	<i>Date</i> , * <u>NONE</u>	Optional
SEQNBR	Sequence number	1-9999, * <u>SEARCH</u>	Optional
VOL	Volume identifier	Values (up to 50 repetitions): <i>Character value</i> , * <u>MOUNTED</u>	Optional
ENDOPT	End of tape option	* <u>REWIND</u> , *LEAVE, *UNLOAD	Optional
PHYFILE	Physical file	<i>Qualified object name</i>	Optional
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
MBROPT	Data base member option	* <u>NOREPLACE</u> , *REPLACE	Optional
DATDIFF	Date differentiated file	* <u>NO</u> , *YES	Optional

Top

To file (TOFILE)

Specifies the name given to a single file when it is restored to the system or to a group of files from a save all set that are restored to the system. If a single file is restored, this parameter allows the file to be renamed at the time it is restored to the system.

This is a required parameter.

*SET Specifies that a group of files from a save all set on diskette or tape are restored to the system.

file-name

Specifies the file name that is given to a single file when the file is restored to the system.

Top

To library (TOLIB)

Specifies which library should contain the file being restored.

This is a required parameter.

Top

Device (DEV)

Specifies the name of a diskette unit, the names of one or more tape devices, or an indication that the file to be restored is in a database physical file. A maximum of four tape device names can be specified. If more than one tape device is used, enter the names of the devices in the order in which they are used. Each device name must be already known on the system by a device description.

***PHYFILE**

The file to be restored is in a database physical file. The name of the database physical file is specified by the **Physical file** prompt (PHYFILE parameter).

device-name

Specify the name of the diskette unit or the names of one or more tape devices that are used for the restore operation. If more than one tape device is used, enter the names of the devices in the order in which they are used. A maximum of four tape device names can be specified.

Top

Set identifier (SET)

Specifies the name used to identify the save all set files saved on the diskette or tape by the SAVE procedure or the \$COPY utility on the System/36, System/34, or System/32.

#SAVE

The files are restored from a save all set with a set identifier of **#SAVE**.

set-identifier

Specifies the set-identifier of the save all set. The set-identifier can be from 1-8 characters long. The first character must be alphabetic (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas(,), apostrophes('), and blanks.

Top

User specified DBCS data (IGCDTA)

Specifies whether the file contains double-byte character set (DBCS) data.

Note: If a file already exists by the name specified on the **To file** prompt (TOFILE parameter), the double-byte character set (DBCS) capability of the existing file is not changed by the restore operation.

***NO**

The files being restored may not contain DBCS data. If a file already exists in the specified library with the same name as the file being restored, and the file allows DBCS data, an informational message is sent, and the restore operation continues. The resulting file allows DBCS data (the file's IGC attributes are not changed by the restore operation).

***YES**

The files being restored may contain DBCS data. If a file already exists in the specified library with the same name as the file being restored, and the file does not allow DBCS data, an informational message is sent, and the restore operation continues. The resulting file will not allow DBCS data (the file's IGC attributes are not changed by the restore operation).

Top

File label (FROMLABEL)

For a single file restore operation, this parameter can be used to specify the label of the diskette or tape file that contains the file that is to be restored to the system. If no value is specified, the file name specified for the **To file** prompt (TOFILE parameter) is used as the diskette or tape file label.

For a group restore operation, this parameter can be used to specify the diskette or tape file label within a save all set where the restore operation is to begin. If no value is specified, the restore operation begins with the first file in the set.

If a label is specified, it must be a maximum of eight characters long.

Top

Creation date (CRTDATE)

Specifies the creation date of the diskette file or tape file used for the restore operation. The specified date is changed to Julian format (cyydd) for tape or international format (yyymmdd) for diskette.

Top

Sequence number (SEQNBR)

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

*SEARCH

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified *LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If *LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the tapes or diskettes used for restoring the file.

*MOUNTED

The volume currently placed in the device is used.

volume-identifier

Specify the volume identifiers of the tapes or diskettes used for restoring the file. A maximum of 50 volume identifiers can be specified.

Top

End of tape option (ENDOPT)

Specifies, only when tape is used, what positioning operation is automatically performed on the tape volume after the restore operation ends. This parameter applies only to the last reel used.

***REWIND**

The tape is rewound, but not unloaded.

***LEAVE**

The tape is not rewound.

***UNLOAD**

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

Top

Physical file (PHYFILE)

Specifies the name of the database physical file that is used as the input file for the restore process. If the specified file does not exist, or is not a physical file, or the file contains no members, a message is sent. If the file contains multiple members, the first member of the file is used.

The possible library values are:

***LIBL** The library list is used to locate the file.

***CURLIB**

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

library-name

Specify the library where the file is located.

Top

Data base member option (MBROPT)

Specifies whether the data of an existing physical file member is replaced.

***NOREPLACE**

Data in an existing physical file member with the same name is *not* replaced, and an error message is sent to the user.

***REPLACE**

Data in an existing physical file member with the same name is replaced.

Top

Date differentiated file (DATDIFF)

Specifies whether the restore operation allows multiple files with the same name but different file creation dates.

***NO** Multiple files with the same name but different file creation dates are not allowed.

If the restore operation is done with *NO specified, and a file already exists in the specified library with the same name, the following actions are taken:

- If the MBROPT parameter specifies the member should be replaced (*REPLACE), the data in the member is replaced with the data from the saved file.
- If the MBROPT parameter specifies the member should not be replaced (*NOREPLACE), a message is sent and the file is not restored.
- If no file exists by the name specified on the TOFILE parameter in the specified library, the file is restored normally.

***YES** Multiple files with the same name but different file creation dates are allowed.

If the restore operation is done with ***YES** specified, and a file already exists in the specified library with the same name, the following actions are taken:

- If a member does not exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, a new member is added to the file and the data from the saved file is copied to it.
- If a member does exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, and the MBROPT parameter specifies the member should be replaced (***REPLACE**), the data in the member is replaced with the data from the saved file.
- If a member does exist with the name of 'Myymmdd', where 'yymmdd' is the creation date of the saved file, and the MBROPT parameter specifies the member should not be replaced (***NOREPLACE**), a message is sent and the file is not restored.

If a file does not already exist in the specified library with the same name, a new file is created, a member is added to the file and the data from the saved file is copied into the new member.

Top

Examples

Example 1: Restoring From Diskette

```
RSTS36F  TOFILE(ACCTRCV)  TOLIB(QS36F)  DEV(I1)
          CRTDATE('01/22/85')  VOL(SAVE1)
```

This command restores the file ACCTRCV into library QS36F. Assuming that I1 is the name of a diskette device description object, the file is restored from the diskette placed in the diskette device. The diskette must have a volume name of SAVE1. The diskette file used for the restore must have a file label of ACCTRCV and a creation date of January 22, 1985 (assuming the job date format is *MDY and the date separator is a '/').

Example 2: Restoring From Tape

```
RSTS36F  TOFILE(PAY.VIEW)  TOLIB(PAYLIB)  DEV(T1)  FROMLABEL('P*V')
          ENDOPT(*LEAVE)
```

The file P*V is restored from device T1 as a file named PAY.VIEW in library PAYLIB. Assuming T1 is a tape device, the file is copied from one or more tapes that are on device T1. No check is made on the tape volume name. When the restore operation ends, the tape is left positioned at the end of tape file P*V.

Example 3: Restoring from a Physical File

```
RSTS36F  TOFILE(ACCTPAY)  TOLIB(QS36F)  DEV(*PHYFILE)
          PHYFILE(NETLIB/SENDFILE)
```

This command restores the file ACCTPAY in library QS36F from physical file SENDFILE in library NETLIB.

Example 4: Specifying Sequence Numbers

```
RSTS36F  TOFILE(*SET)  TOLIB(QS36F)  DEV(T1 T2)
          SET(PAYFILES) FROMLABEL(FILE10)
          MBROPT(*REPLACE) DATDIFF(*YES)
          SEQNBR(*SEARCH) VOL(*MOUNTED) ENDOPT(*REWIND)
```

This command restores a subset of the files in the save all set called PAYFILES to library QS36F from tape. The restore operation begins with a tape file whose label is file 10. If one of the files being restored already exists in library QS36F with the same creation date as the saved file, the file is replaced. If a file already exists in library QS36F with a different creation date, a new date-differentiated number is added to the file. The restore operation uses the tape volumes that are placed in tape drives T1 and T2. After the restore operation is complete, the last tape volume is rewound to the beginning of the tape.

Top

Error messages

*ESCAPE Messages

CPF2C4A

Device &1 not correct for command.

CPF2C4B

Duplicate device &1 specified in device name list.

CPF2C4C

Diskette device &1 included in multiple device specification.

CPF2C4D

Not all files were restored.

CPF2C4E

Restore operation ended before all files were restored.

CPF2C45

Input file &1 cannot be processed by RSTS36F.

CPF2C47

Existing file &1 or member &3 in library &2 not replaced.

CPF2C48

Input file &1 in &2 not correct for command.

CPF2C49

Output file &1 in &2 not correct for command.

CPF2C5A

Alternate index file &1 in library &2 not replaced.

CPF2C5E

Input file &1 in &2 not correct for command.

CPF2C50

File description for file &1 is not available.

CPF2C52

Error occurred during attempt to create file &1 in library &2.

CPF2C53

Member &3 not added to file &1 in library &2 because error occurred.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9826

Cannot allocate file &2.

CPF9830

Cannot assign library &1.

CPF9831

Cannot assign device &1.

CPF9845

Error occurred while opening file &1.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

STATUS Messages*CPI2C11**

Copying records to file &1 in library &2 member &3.

Top

Restore S/36 Library Members (RSTS36LIBM)

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

Parameters
Examples
Error messages

The Restore System/36 Library Members (RSTS36LIBM) command reads a file containing library members, creates database source or data file members on this system, and copies the member data from the file into each restored member.

The input file can be a diskette file, tape file, or database physical file on this system.

The file could have been created on a System/36, a System/34, or a System/32 using either the FROMLIBR or SAVELIBR system operator control language (OCL) procedure (or the equivalent use of \$MAINT), or by using the Save System/36 Library Members (SAVS36LIBM) command.

Diskette files created on a System/34 using the BACKUP procedure or \$BACK utility are not accepted by this command. Compressed SAVELIBR diskette files (used by IBM to distribute system libraries for System/36 after release 5.0) are not accepted by this command.

In System/36 terms, the input file format could be a SAVELIBR diskette file or tape file, a record-mode LIBRFILE diskette file, tape file, or physical file, or a sector-mode LIBRFILE diskette file, tape file, or physical file. In other words, the input file can be any diskette file, tape file, or physical file created by the System/36 \$MAINT SSP utility.

If the library identified by the value specified for the **To library** prompt (TOLIB parameter) does not exist, it is created. Also, the source files QS36SRC and QS36PRC are created if they do not exist in the restore-to library. If the restore operation creates the library, the new library is owned by the user running the RSTS36LIBM command and the library is created with a default authority of *ALL (that is, the same as AUT(*ALL)).

If a sector-mode FROMLIBR file or a SAVELIBR file created on a System/36 is being restored, data files QS36LOD and QS36SBR may also be created to hold restored load and subroutine members. Restored load and subroutine members are not converted on the current system.

Restrictions:

- The following authorities are required when running on a system using resource security:
 1. *SECADM and *ALLOBJ authorities
 2. *USE authority for this command and *USE authority for the Create Library (CRTLIB) command if the library needs to be created
 3. *USE authority for the CRTSRCPF command if QS36SRC or QS36PRC must be created
 4. *USE authority for the CRTPF command if QS36LOD or QS36SBR must be created
 5. *CHANGE authority for the library specified in the TOLIB parameter
 6. *CHANGE and *OBJMGMT authority for file QS36SRC in the specified library if restoring source library members
 7. *CHANGE and *OBJMGMT authority for file QS36PRC in the specified library if restoring procedure library members
 8. *CHANGE and *OBJMGMT authority for file QS36LOD in the specified library if restoring load library members
 9. *CHANGE and *OBJMGMT authority for file QS36SBR in the specified library if restoring subroutine library members

10. *USE authority for the diskette device description object, *USE authority for device file QSYSDKT in library QSYS if restoring from diskette
 11. *USE authority for the tape device description object and *USE authority for device file QSYSTAP in library QSYS, if restoring from tape
 12. *USE authority for the file and *USE authority for the library that contains the file (PHYFILE parameter) if restoring from a database physical file
- No object-level or record-level operations should be active for files QS36SRC, QS36PRC, QS36SBR, and QS36LOD while members are being restored by RSTS36LIBM. If the necessary files cannot be allocated exclusively, no members are restored.
 - The member name or generic member name specified (TOMBR parameter) must meet i5/OS naming standards.

If a generic member name or *ALL is specified, a member may be selected to be restored that has a name containing characters not allowed in an i5/OS *simple* object name. In this case, the member name is restored using the i5/OS *extended* name syntax (for example, A!B would become "A!B").

If the name contains a blank, a single quotation mark (') a double quotation mark ("), an asterisk (*), a question mark (?), or a device control character (hexadecimal '00'-'3F' or hexadecimal 'FF'), these characters are replaced by underlines and the member is restored using the resulting simple or extended name (for example, A*/? would become A_/_ and A? would become A_).

An informational message is sent each time invalid characters are replaced to get a valid name. An additional informational message is sent if the resulting name change caused a member to be replaced. No message is sent if a member is restored using the extended name syntax without replacing invalid characters.

Top

Parameters

Keyword	Description	Choices	Notes
TOMBR	To member	<i>Generic name, name, *ALL</i>	Required, Positional 1
TOLIB	To library	<i>Name</i>	Required, Positional 2
DEV	Device	Single values: *PHYFILE Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
SRCMBRS	S/36 source members	<u>*ALL</u> , *SRC, *PRC, *NONE	Optional
MBROPT	Data base member option	<u>*NEW</u> , *OLD, *ALL	Optional
IGCDTA	User specified DBCS data	<u>*NO</u> , *YES	Optional
FROMLABEL	File label	<i>Character value</i>	Optional
CRTDATE	Creation date	<i>Date</i> , <u>*NONE</u>	Optional
SEQNBR	Sequence number	1-9999, <u>*SEARCH</u>	Optional
VOL	Volume identifier	Values (up to 50 repetitions): <i>Character value</i> , <u>*MOUNTED</u>	Optional
ENDOPT	End of tape option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
PHYFILE	Physical file	<i>Qualified object name</i>	Optional
	Qualifier 1: Physical file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , <u>*LIBL</u> , *CURLIB	
OBJMBRS	S/36 object members	<u>*NONE</u> , *SBR, *LOD, *ALL	Optional

Top

To member (TOMBR)

Specifies the names of the members to restore.

This is a required parameter.

***ALL** All members of the specified member types are restored.

member-name

The members having the specified member name are restored.

generic-member-name*

All members that have the specified generic member name are restored.

Top

To library (TOLIB)

Specifies the library to contain the members to restore from the input file.

This is a required parameter.

Top

Device (DEV)

Specifies the names of the devices to use for the restore operation. A maximum of four device names may be specified.

This is a required parameter.

***PHYFILE**

The input file is the database physical file specified by the **Physical file** prompt (PHYFILE parameter).

device-name

Specify the name of the diskette unit or the names of one or more tape devices that are to be used for the restore operation. If more than one tape device is used, specify the names of the devices in the order in which they are used. A maximum of four tape device names can be specified.

Top

S/36 source members (SRCMBRS)

Specifies which source member types (source and procedure members on System/36) are restored.

***ALL** Restore all source and procedure members in the input file that match the member name specified on the **To member** prompt (TOMBR parameter).

***SRC** Restore only System/36 source members (to file QS36SRC) that match the member name specified on the **To member** prompt (TOMBR parameter).

***PRC** Restore only System/36 OCL procedure members (to file QS36PRC) that match the member name specified on the **To member** prompt (TOMBR parameter).

***NONE**

Do not restore any System/36 source or procedure library members.

Data base member option (MBROPT)

Specifies, for database files currently on the system, which file members are restored.

- *NEW** Only new members (members that don't already exist in the appropriate file) are restored.
- *OLD** Only old members (members that already exist in the appropriate file) are restored. The existing member is replaced by the copy of the member restored from the file.
- *ALL** All members are restored. Members that don't already exist are created, and members that do exist are replaced.

Top

User specified DBCS data (IGCDTA)

Specifies whether the source and procedure members being restored can contain double-byte character set (DBCS) data. This attribute is used if the restore operation needs to create source files QS36SRC and QS36PRC to hold the restored library members.

Note: If the QS36SRC or QS36PRC source file already exists in the library specified on the **To library** prompt (TOLIB parameter) and the file's DBCS capability does not match this parameter, an error message is sent and no members are restored.

- *NO** The source or procedure members being restored cannot contain double-byte character set data.
- *YES** The source or procedure members being restored can contain double-byte character set data.

Top

File label (FROMLABEL)

Specifies the label value (eight characters maximum) of the diskette or tape file that contains the members to be restored. If *PHYFILE is not specified for the **Device** prompt (DEV parameter), a value must be specified here.

Top

Creation date (CRTDATE)

Specifies the creation date of the diskette file or tape file used for the restore operation. The specified date is changed to Julian format (cyydd) for tape or international format (yymmdd) for diskette.

Top

Sequence number (SEQNBR)

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

***SEARCH**

The volume in the device is searched for a data file with an identifier that matches the FROMLABEL parameter value; when a match is found, the object is restored. If the last operation on the device specified *LEAVE for the **End of tape option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with

the first data file beyond the current tape position. If *LEAVE was not used for the **End of tape option (ENDOPT)** parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the tapes or diskettes used for restoring the file.

***MOUNTED**

The volume currently placed in the device is used.

volume-identifier

Specify the volume identifiers of the tapes or diskettes used for restoring the file. A maximum of 50 volume identifiers can be specified.

Top

End of tape option (ENDOPT)

Specifies, only when tape is used, what positioning operation is automatically performed on the tape volume after the restore operation ends. This parameter applies only to the last reel used.

***REWIND**

The tape is rewound, but not unloaded.

***LEAVE**

The tape is not rewound.

***UNLOAD**

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

Top

Physical file (PHYFILE)

Specifies the name of the database physical file that is used as the input file for the restore operation. If the specified file does not exist or is not a physical file, no library members are restored. If the file contains multiple members, the first member of the file is used.

The possible library values are:

***LIBL** The library list is used to locate the file.

***CURLIB**

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

library-name

Specify the library where the file is located.

Top

S/36 object members (OBJMBRS)

Specifies which object member types (load and subroutine members on System/36) are restored. Because the System/36 and the i5/OS are not object compatible, any restored members are not immediately useable after the restore operation. IBM-supplied commands or user-written operations must be run to convert the object member to a useable i5/OS object.

*NONE

System/36 load or subroutine library members are not restored.

***SBR** Only System/36 subroutine members (to file QS36SBR) that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

***LOD** Only System/36 load members (to file QS36LOD) that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

***ALL** All load and subroutine members in the input file that match the member name specified on the **To member** prompt (TOMBR parameter) are restored.

Top

Examples

Example 1: Restoring All Members

```
RSTS36LIBM TOMBR(XYZ1) TOLIB(JOHNSON) DEV(I1)
           SRCMBRS(*PRC) MBROPT(*ALL) FROMLABEL('XYZ1')
```

This command restores the single OCL procedure member XYZ1 as a member of source file QS36PRC in library JOHNSON. Assuming I1 refers to a diskette device, the input diskette file must have the label XYZ1.

Example 2: Restoring Members from a File

```
RSTS36LIBM TOMBR(X*) TOLIB(ORDER) DEV(*PHYFILE)
           PHYFILE(NETLIB/S36SRC)
```

This command restores all source and procedure members with names starting with the character 'X' and that do not already exist as members of QS36SRC and QS36PRC in library ORDER. The members are restored from file S36SRC in library NETLIB.

Top

Error messages

*ESCAPE Messages

CPF2C4A

Device &1 not correct for command.

CPF2C4B

Duplicate device &1 specified in device name list.

CPF2C4C

Diskette device &1 included in multiple device specification.

CPF2C41
&2 members restored, &3 members not restored, &4 members excluded.

CPF2C42
No members restored to library &1, &4 members excluded.

CPF2C46
Input file &1 cannot be processed by RSTS36LIBM.

CPF2C48
Input file &1 in &2 not correct for command.

CPF2C49
Output file &1 in &2 not correct for command.

CPF2C5E
Input file &1 in &2 not correct for command.

CPF2C50
File description for file &1 is not available.

CPF2C52
Error occurred during attempt to create file &1 in library &2.

CPF2C53
Member &3 not added to file &1 in library &2 because error occurred.

CPF2C57
DBCS attribute not same as existing file &1 in &2.

CPF2C70
Input file &1 contains incorrect control statement record.

CPF2C71
NAME keyword on COPY statement missing or incorrect.

CPF2C72
LIBRARY keyword on COPY statement missing or incorrect.

CPF9807
One or more libraries in library list deleted.

CPF9808
Cannot allocate one or more libraries on library list.

CPF9810
Library &1 not found.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9824
Not authorized to command &1 in library &2.

CPF9825
Not authorized to device &1.

CPF9826

Cannot allocate file &2.

CPF9830

Cannot assign library &1.

CPF9845

Error occurred while opening file &1.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9849

Error while processing file &1 in library &2 member &3.

STATUS Messages*CPI2C12**

Copying file &1 in library &2 label &3 to work file.

Top

Restore System Information (RSTSYSINF)

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

Parameters
Examples
Error messages

The Restore System Information (RSTSYSINF) command restores the subset of system data and objects saved by the Save System Information (SAVSYSINF) command.

RSTSYSINF is not to be used for system upgrades or migrations.

Security related system values may not be restored if they have been locked. For information on how to lock and unlock security related system values, see the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

The QPWDLVL (Password level) system value will not be restored. See Planning Password Level Changes in Chapter 7 of the System i Security Reference, SC41-5302 before changing the QPWDLVL system value. Also, based on the current setting of the QPWDLVL system value, QPWDMAXLEN (Maximum password length), QPWDMINLEN (Minimum password length), and QPWDVLDPGM (Password validation program) system values may not be restored.

Note: The RSTSYSINF command issues several restore commands for restoring objects. Parameters may or may not be used for all restore commands.

For more information, refer to the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 1
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	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	
OPTFILE	Optical file	<i>Path name</i> , _	Optional
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	

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

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Single values

*MOUNTED

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

Specifies the sequence number that is used for the restore operation.

*SEARCH

The volume placed in a device is searched for a data file containing the saved system information objects. When a match is found, the system information objects are restored.

1-16777215

Specify the sequence number of the file to be used for the restore operation.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

***** The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'optical-directory-path-name'

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

***NONE**

No output is created.

***PRINT**

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

***OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the OUTFILE parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Examples

Example 1: System Information Restored from TAP01

```
RSTSYSINF  DEV(TAP01)
```

This command restores the system information from the tape put on the TAP01 tape drive.

Example 2: System Information Restored from SAVF and Printed Output Generated

```
RSTSYSINF  DEV(*SAVF) SAVF(QGPL/SAVF) OUTPUT(*PRINT)
```

This command restores the system information from the save file named SAVF in library QGPL. Information about what was restored will be written to a spooled file.

[Top](#)

Error messages

***ESCAPE Messages**

CPF38A8

RSTSYSINF completed. One or more objects not restored.

[Top](#)

Restore User Profiles (RSTUSRPRF)

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

Parameters
Examples
Error messages

The Restore User Profile (RSTUSRPRF) command restores the basic parts of a user profile or a set of user profiles that were saved by the Save System (SAVSYS) command or Save Security Data (SAVSECDTA) command. The Restore User Profile (RSTUSRPRF) command restores only the special authority granted in the Create User Profile (CRTUSRPRF) command; it does not restore the authority for the named objects owned by other users. To restore authority for objects owned by other users, the Restore Authority (RSTAUT) command must be used after the profiles, libraries, and objects are restored. If all user profiles are being restored, authorization lists and authority holders that existed when the SAVSYS or SAVSECDTA command was run are also restored.

If you specify USRPRF(*ALL) or SECDDTA(*DCM), all other operations on the system must be stopped. This requires ending all subsystems through the End Subsystem (ENDSBS(*ALL)) command or End System (ENDSYS) command or specifying this command when the operating system is started. The RSTUSRPRF command is normally used after the restore of the operating system but before the user libraries are restored. The user profiles must be restored before any libraries or objects belonging to them can be restored. After the libraries and their objects are restored, the authority for the objects is restored to the user profiles by the RSTAUT command. At the completion of the command, either message CPF3775 or message CPC3705 is sent to QHST. More information on restoring the system is in the Recovering your system book, SC41-5304.

The following situations may apply to user profiles being restored by the RSTUSRPRF command:

- If a user profile exists on the system, but not on the media, the system profile remains.
- If a user profile exists on the media, but not on the system, a new user profile is created.
- If the user profile exists on both the media and the system, the media user profile is restored.
- If the user profile exists on the media and is being restored individually, the new user profile is created without its password or group connection.
- If the user profile exists on both the media and the system, and it is being restored individually, the media user profile is restored. However, the password and group connection on the system remains unchanged.
- If all user profiles are being restored, the passwords and group connections are also restored from the media.
- If user profiles exist on the system, there are no changes to the existing object authorities.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- You must have save system (*SAVSYS) special authority to run this command.
- This command is shipped with no public authority (*EXCLUDE).
- If you specify USRPRF(*ALL) or SECDDTA(*DCM), all other operations on the system must be ended. The End System (ENDSYS) or End Subsystem (ENDSBS) command can be used to end these operations. You must have job control (*JOBCTL) special authority to use the ENDSYS or ENDSBS command.
- You must specify USRPRF(*ALL) to restore authorization lists and authority holders.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 1
USRPRF	User profile	Single values: *ALL, *NEW, *NONE Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional, Positional 2
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, *SEARCH	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MAIL	Mail	*NO, *YES	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE, *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional
OMITUSRPRF	User profiles to omit	Single values: *NONE Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
SECDTA	Security data	*USRPRF, *PVTAUT, *PWDGRP, *DCM	Optional
OMITSECDTA	Omit security data	Single values: *NONE Other values (up to 3 repetitions): *AUTL, *DCM, *FCNUSG	Optional
OUTPUT	Output	*NONE, *OUTFILE	Optional
OPTFILE	Optical file	<i>Path name, *</i>	Optional
SAVASPDEV	Saved from ASP device	<i>Name, *ANY, *, *SYSBAS, *CURASPGRP</i>	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	*REPLACE, *ADD	

Top

Device (DEV)

Specifies the name of the device used for the restore operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The restore operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the restore operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the restore operation.

tape-device-name

Specify the names of one or more tape devices used for the restore operation. If a virtual tape device is used, it must be the only device specified. If you are using more than one tape device (up to a maximum of four), specify the names of the devices in the order in which they are used. When more than one tape volume is to be restored, using more than one tape device permits one tape volume to be rewound while another tape device processes the next tape volume.

Top

User profile (USRPRF)

Specifies the user profiles to be restored. The user profiles must exist on the media from the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command in order to be restored.

Single values

***ALL** All the user profiles, authorization lists, authority holders, and internal authority objects that were saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command are restored.

***NEW** All the user profiles, authorization lists, authority holders, and internal authority objects that were saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) command which currently do not exist on the system are restored.

***NONE** No user profiles are restored. This value may be specified only if ***DCM** is specified for the **Security data (SECDTA)** parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of sets of user profiles to restore. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete user profile name.

name Specify one or more names of specific user profiles that are restored. Both generic names and specific names can be specified in the same command.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the media or the cartridge identifiers of tapes in a tape media library device, from which the objects are being restored. The volumes must be in the same order as they were when the data was saved. The volume that contains the beginning of the file to be restored should be placed in the device.

Single values

***MOUNTED**

The objects are restored from the volumes placed in the device specified for the **Device (DEV)** 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to restore the data.

Top

Sequence number (SEQNBR)

Specifies the sequence number of the tape file used for the restore process.

*SEARCH

The volume placed in the device is searched for a file containing the saved user profiles; when a match is found, the user profiles are restored. If a match is not found, you must load another tape and try the command again.

If the last operation on the device specified *LEAVE for the **End of media option (ENDOPT)** parameter, indicating that the tape is positioned at the location where the last operation ended, the file search starts with the first data file beyond the current tape position. If *LEAVE was not used for the ENDOPT parameter of the last operation, or if the tape was manually rewound since the operation, the search starts with the first data file on the volume.

1-16777215

Specify the sequence number of the file to be used to restore user profiles.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the restore operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Save file (SAVF)

Specifies the save file used to restore the data.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Mail (MAIL)

Specifies whether the OfficeVision distribution objects saved from a release before V2R2M0 are restored.

Note: You can specify *YES on this parameter only if you specify *ALL for the **User profile (USRPRF)** parameter.

***NO** Distribution objects that are part of your mail are not restored along with restoring the user profile.

***YES** Distribution objects that are part of your mail are restored along with restoring the user profile if the save data was created before release V2R2M0. Otherwise, no distribution objects are restored. For saved distribution objects created on V2R2M0 or later, specify DLO(*MAIL) on the Restore Document Library Objects (RSTDLO) command to restore your mail.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- Authorization list: The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.

Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

***ALL** All of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

Other values (up to 3 repetitions)

***AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Note: The *PGP value does not apply to user profiles. User profiles with primary group differences are always restored.

Top

User profiles to omit (OMITUSRPRF)

Specifies user profiles to be omitted from the restore.

Single values

*NONE

None of the user profiles will be omitted from the restore.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of sets of user profiles to be omitted from the restore. A generic name is a character string that contains one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid character. A generic names specifies all user profiles that begin with the prefix. If an asterisk is not included with the generic name, the system assumes it to be the complete object name.

name Specify one or more names of specific user profiles that are to be omitted from the restore. Both generic names and specific names can be specified in the same command.

Top

Security data (SECDDTA)

Specifies what authority information is restored for the user profiles specified for the **User profile (USRPRF)** parameter.

*USRPRF

The specified user profiles and their private authorities are restored. If *ALL is specified for the USRPRF parameter, the passwords and group linkages are also restored. Otherwise, the passwords and group linkages for the specified user profiles are not restored.

***PVTAUT**

Only the private authorities for the specified user profiles and auxiliary storage pools are restored. The information is used by the Restore Authority (RSTAUT) command to restore the private authorities to the referenced objects. This value cannot be specified if *NEW is specified for the USRPRF parameter.

***PWDGRP**

The specified user profiles, their private authorities, and their passwords and group linkages are restored.

***DCM** Only the internal objects required by Digital Certificate Manager (DCM) are restored. No user profiles are restored. If this value is specified, then *NONE must be specified for the USRPRF parameter.

Top

Omit security data (OMITSECDDTA)

Specifies authority information to be omitted from the restore operation when *ALL is specified for the **User profile (USRPRF)** parameter.

Single values

*NONE

No security information is omitted.

Other values (up to 3 repetitions)

***AUTL**

Authorization list (*AUTL) and authority holder (*AUTHLR) objects are omitted. However, for any of these objects that already exist on the system, any specific user authorities will be restored. You need to run the Restore Authority (RSTAUT) command to complete the restore of these authorities.

***DCM** The internal objects required by Digital Certificate Manager (DCM) are omitted.

*FCNUSG

Function usage information is omitted. However, for any function identifiers that already exist on the system, any specific user settings will be restored. You need to run the Restore Authority (RSTAUT) command to complete the restore of these settings.

Top

Output (OUTPUT)

Specifies whether a listing that shows information about the status of the objects is created and directed to an output file. The listing shows the restore information and shows all objects restored, not restored, and excluded. Information about each object's security is listed for the restored objects.

*NONE

No output is created.

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the OUTFILE parameter when *OUTFILE is specified for this parameter.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the restore operation, beginning with the root directory of the volume.

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

*
_ The system searches the root directory of the optical volume for the default name generated by the corresponding save operation.

'*optical-directory-path-name*'*

The system searches the specified directory of the optical volume for the default name generated by the corresponding save operation.

Top

Saved from ASP device (SAVASPDEV)

Specifies the name of the auxiliary storage pool (ASP) device from which private authority information was saved. The private authority information is restored for later use by the Restore Authority (RSTAUT) function.

***ANY** The private authority information saved from all ASPs included in the save operation is restored.

* The private authority information saved from the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group is restored.

*SYSBAS

The private authority information saved from the system ASP and all basic user ASPs is restored.

***CURASGRP**

If the current thread has an ASP group, the private authority information saved from all independent ASPs in the ASP group is restored.

name Specify the name of the ASP device from which private authority information was saved.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASRRSTO with format name QSRRST as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Examples

Example 1: Restoring All Profiles

```
RSTUSRPRF  DEV(TAP01)  SEQNBR(*SEARCH)  ENDOPT(*REWIND)
```

This command restores all user profiles contained on the tape currently put on the tape drive named TAP01 to the system. The tape is searched for the file, and the tape is rewound on completion or at the end of restore.

Example 2: Restoring Specific User Profiles

```
RSTUSRPRF  DEV(TAP01)  USRPRF(USRA USRB USRC USER*)
```

This command restores user profiles USRA, USRB, and USRC, along with all the user profiles whose names start with USER. The saved version of all the user profiles must exist on the tape placed on tape drive TAP01.

Example 3: Restoring User Profiles from a Save File

```
RSTUSRPRF  DEV(*SAVF)  USRPRF(USRX USRY)  SAVF(QGPL/SAVESEC)
```

This command restores user profiles USRX and USRY to the system from the save file SAVESEC in library QGPL.

Example 4: Reporting Information about User Profiles Restored and Not Restored

```
RSTUSRPRF  DEV(TAP01)  USRPRF(*ALL)  OUTPUT(*OUTFILE)
           OUTFILE(PRFS92)  OUTMBR(FOURQT *ADD)
```

This command restores all user profiles from the tape device TAP01. A list reporting information about user profiles restored and not restored is directed to the output file PRFS92. The output is received in the member FOURQT as an addition to existing information in the member.

Error messages

*ESCAPE Messages

CPD3774

USRPRF(*ALL) required when MAIL(*YES) specified.

CPF2206

User needs authority to do requested function on object.

CPF222E

&1 special authority is required.

CPF370C
Not authorized to ALWOBJDIF parameter.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3733
&2 &1 in &3 previously damaged.

CPF3738
Device &1 used for save or restore is damaged.

CPF3743
File cannot be restored, displayed, or listed.

CPF3748
Object information for library &1 damaged.

CPF376B
File &1 not found.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3775
Not all user profiles or authority objects restored.

CPF3780
Specified file for library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3785
Not all subsystems ended.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3796
Storage limit exceeded for user profile &4.

CPF380C
Library &1 not restored.

CPF3812
Save file &1 in &2 in use.

CPF908A
Requester &1 not enrolled.

CPF9812
File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9833

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

CPF8ED

Device description &1 not correct for operation.

Top

Return from Subroutine (RTNSUBR)

Where allowed to run:

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

Threadsafe: Yes

Parameters
Examples
Error messages

The Return from Subroutine (RTNSUBR) command returns control to the command immediately following the Call Subroutine (CALLSUBR) command that called the subroutine. The RTNSUBR command must be used within a subroutine, and multiple RTNSUBR commands may be used in a 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 subroutine.

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
:
```

```
IF (&A *LT 30) THEN(RTNSUBR RTNVAL(-1))
:
ENDSUBR
ENDPGM
```

This CL subroutine is exited by the RTNSUBR command if &A is less than 30, with a return value of -1.

[Top](#)

Error messages

None

[Top](#)

Retrieve Auth List Entry (RTVAUTLE)

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

Parameters
Examples
Error messages

The Retrieve Authorization List Entry (RTVAUTLE) command is used in a CL program or REXX procedure to retrieve the authorities that a user has on the authorization list. It can be used with the Change Authorization List Entry command to change the user's authorities to include new authorities in addition to the existing authorities for the user.

The authorization list name and user name must be specified. The variables for each of the authorities the user might have are returned blank if the user does not have the authority; they are returned with the correct value for the Change Authorization List Entry (CHGAUTLE) command if the user has the authority. The values are returned in the specified variables for the specified user.

Users with authorization list management (*AUTLMGT) authority, or who own the authorization list, can retrieve authority for any user on the list. Other users can get their own authorities or the authority of *PUBLIC.

Top

Parameters

Keyword	Description	Choices	Notes
AUTL	Authorization list	Name	Required, Positional 1
USER	User	Name, *PUBLIC	Required, Positional 2
ALL	CL var for *ALL (10)	Character value	Optional
CHANGE	CL var for *CHANGE (10)	Character value	Optional
USE	CL var for *USE (10)	Character value	Optional
EXCLUDE	CL var for *EXCLUDE (10)	Character value	Optional
OBJALTER	CL var for *OBJALTER (10)	Character value	Optional
OBJEXIST	CL var for *OBJEXIST (10)	Character value	Optional
OBJMGT	CL var for *OBJMGT (10)	Character value	Optional
OBJOPR	CL var for *OBJOPR (10)	Character value	Optional
OBJREF	CL var for *OBJREF (10)	Character value	Optional
READ	CL var for *READ (10)	Character value	Optional
ADD	CL var for *ADD (10)	Character value	Optional
UPDATE	CL var for *UPD (10)	Character value	Optional
DELETE	CL var for *DLT (10)	Character value	Optional
EXECUTE	CL var for *EXECUTE (10)	Character value	Optional
AUTLMGT	CL var for *AUTLMGT (10)	Character value	Optional

Top

Authorization list (AUTL)

Specifies the authorization list that the user's authorities come from.

This is a required parameter.

name Specify the name of the authorization list to be changed.

Top

User (USER)

Specifies the user whose information is to be retrieved. If a variable is specified, it must be a 10-character field which contains a user name or the value *PUBLIC.

This is a required parameter.

*PUBLIC

The information returned in the specified parameter is for the users who do not have any specific authority to the authorization list, and whose groups do not have any specific authority to the authorization list.

name Specify the name of the user profile of the user whose information is to be retrieved.

Top

CL var for *ALL (10) (ALL)

Specifies the name of a variable that is used to return the special value of *ALL, if the user has *ALL authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *CHANGE (10) (CHANGE)

Specifies the name of a variable that is used to return the special value of *CHANGE, if the user has change authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *USE (10) (USE)

Specifies the name of a variable that is used to return the special value of *USE, if the user has use authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *EXCLUDE (10) (EXCLUDE)

Specifies the name of a variable that is used to return the special value of *EXCLUDE, if the user has that authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

[Top](#)

CL var for *OBJALTER (10) (OBJALTER)

Specifies the name of a variable that is used to return *OBJALTER if the user has *OBJALTER authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJALTER authority.

Blanks are returned for the variable if the user does not have this authority.

[Top](#)

CL var for *OBJEXIST (10) (OBJEXIST)

Specifies the name of a variable that is used to return the special value of *OBJEXIST, if the user has that authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

[Top](#)

CL var for *OBJMGT (10) (OBJMGT)

Specifies the name of a variable that is used to return the special value of *OBJMGT, if the user has object management authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

[Top](#)

CL var for *OBJOPR (10) (OBJOPR)

Specifies the name of a variable that is used to return the special value of *OBJOPR, if the user has object operation authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

[Top](#)

CL var for *OBJREF (10) (OBJREF)

Specifies the name of a variable that is used to return *OBJREF if the user has *OBJREF authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *OBJREF authority.

Blanks are returned for the variable if the user does not have this authority.

CL var for *READ (10) (READ)

Specifies the name of a variable that is used to return the special value of *READ, if the user has read authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *ADD (10) (ADD)

Specifies the name of a variable that is used to return the special value of *ADD, if the user has add authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *UPD (10) (UPDATE)

Specifies the name of a variable that is used to return the special value of *UPD, if the user has update authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *DLT (10) (DELETE)

Specifies the name of a variable that is used to return the special value of *DLT, if the user has delete authority. In CL, this should be a 10-character variable.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *EXECUTE (10) (EXECUTE)

Specifies the name of a variable that is used to return *EXECUTE if the user has *EXECUTE authority. In CL programs, the variable has a length of 10 characters. Blanks are returned in the variable if the user does not have *EXECUTE authority.

Blanks are returned for the variable if the user does not have this authority.

Top

CL var for *AUTLMGT (10) (AUTLMGT)

Specifies the name of a variable that is used to return the special value of *AUTLMGT, if the user has authorization list management authority.

Blanks are returned for the variable if the user does not have this authority.

Top

Examples

```
ADDAUTLE  AUTL(PAYROLL)  USER(TOM)
          AUT(*OBJOPR *READ *UPD *AUTLMGT)
```

When user Smith calls a CL program containing the following:

```
DCL      &CHG          *CHAR 10
DCL      &ALL          *CHAR 10
DCL      &USE          *CHAR 10
DCL      &EXCL        *CHAR 10
DCL      &OBJJOP       *CHAR 10
DCL      &ALTER        *CHAR 10
DCL      &REFER        *CHAR 10
DCL      &READ         *CHAR 10
DCL      &ADD          *CHAR 10
DCL      &UPD          *CHAR 10
DCL      &DLT          *CHAR 10
DCL      &EXEC         *CHAR 10
DCL      &AUTLM        *CHAR 10
:
RTVAUTLE  AUTL(PAYROLL)  USER(TOM)  USE(&USE)  +
          OBJJOPR(&OBJJOP)  AUTLMGT(&AUTLM)
```

This command retrieves the following authorities from the authorization list PAYROLL for user TOM: *USE, *OBJJOPR, and *AUTLMGT. If TOM does not have the authority, blanks are returned.

Top

Error messages

*ESCAPE Messages

CPF22A7

User &1 not on authorization list &2, no authorities retrieved.

CPF22A8

Not authorized to retrieve authorities for user &1.

CPF2204

User profile &1 not found.

CPF2283

Authorization list &1 does not exist.

CPF2289

Unable to allocate authorization list &1.

Retrieve Backup Options (RTVBCKUP)

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

Parameters
Examples
Error messages

The Retrieve Backup (RTVBCKUP) command is used in a CL program or REXX procedure allows the user to retrieve the options in one of the predefined backups into CL variables. More information on backup is in the Recovering your system book, SC41-5304.

Top

Parameters

Keyword	Description	Choices	Notes
BCKUPOPT	Backup options	*DAILY, *WEEKLY, *MONTHLY	Required, Positional 1
DEV	CL var for DEV (43)	Character value	Optional
TAPSET	CL var for TAPSET (34)	Character value	Optional
CLRTAP	CL var for CLRTAP (4)	Character value	Optional
SBMJOB	CL var for SBMJOB (4)	Character value	Optional
CHGONLY	CL var for CHGONLY (4)	Character value	Optional
PRTRPT	CL var for PRTRPT (4)	Character value	Optional
LIB	CL var for LIB (10)	Character value	Optional
FLR	CL var for FLR (10)	Character value	Optional
DIR	CL var for DIR (10)	Character value	Optional
SECDTA	CL var for SECDTA (4)	Character value	Optional
CFG	CL var for CFG (4)	Character value	Optional
MAIL	CL var for MAIL (4)	Character value	Optional
CAL	CL var for CAL (4)	Character value	Optional
EXITPGM	CL var for EXITPGM (10)	Character value	Optional
EXITPGMLIB	CL var for EXITPGMLIB (10)	Character value	Optional

Top

Backup options (BCKUPOPT)

Specifies the backup options to be retrieved.

This is a required parameter.

***DAILY**

The daily backup options are retrieved.

***WEEKLY**

The weekly backup options are retrieved.

***MONTHLY**

The monthly backup options are retrieved.

CL var for DEV (43) (DEV)

Specifies the name of the CL variable that receives the device value. The variable has a minimum length of 43 characters. The value returned is a character string of four 10-character device names, separated by blanks.

Top

CL var for TAPSET (34) (TAPSET)

Specifies the name of the CL variable that receives the tape set names. The variable has a minimum length of 34 characters (seven 4-character tape set names, separated by blanks).

Top

CL var for CLRTAP (4) (CLRTAP)

Specifies the name of the CL variable that receives the indicator for clearing the tape for backup. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for SBMJOB (4) (SBMJOB)

Specifies the name of the CL variable that receives the indicator of whether the backup is run as a batch job. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for CHGONLY (4) (CHGONLY)

Specifies the name of the CL variable that receives the indicator for saving changed objects only. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for PRTRPT (4) (PRTRPT)

Specifies the name of the CL variable that receives the indicator for printing a report of saved objects. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for LIB (10) (LIB)

Specifies the name of the CL variable that receives the value specifying the libraries to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR, *FROMLIST, or *NONE is returned.

Top

CL var for FLR (10) (FLR)

Specifies the name of the CL variable that receives the value specifying the folders to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALL, *FROMLIST, or *NONE is returned.

Top

CL var for DIR (10) (DIR)

Specifies the name of the CL variable that receives the value specifying the user directories to save with this backup. The variable must have a minimum length of 10 characters. A value of *ALLUSR or *NONE is returned.

Top

CL var for SECDTA (4) (SECDTA)

Specifies the name of the CL variable that receives the indicator for saving security data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for CFG (4) (CFG)

Specifies the name of the CL variable that receives the indicator for saving configuration data. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for MAIL (4) (MAIL)

Specifies the name of the CL variable that receives the indicator for saving OfficeVision mail. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for CAL (4) (CAL)

Specifies the name of the CL variable that receives the indicator for saving OfficeVision calendars. The variable must have a minimum length of 4 characters. The value returned is either *YES or *NO.

Top

CL var for EXITPGM (10) (EXITPGM)

Specifies the name of the CL variable that receives the name of the user program to call before and after the backup is run. The variable must have a minimum length of 10 characters. If no exit program is specified, *NONE is returned.

Top

CL var for EXITPGMLIB (10) (EXITPGMLIB)

Specifies the name of the CL variable that receives the name of the library that contains the exit program. The variable must have a minimum length of 10 characters. If no exit program is specified, blanks are returned. If *LIBL is returned, the program uses the library list.

Top

Examples

```
RTVBCKUP  BCKUPOPT(*DAILY)  SBMJOB(&SBMJOBVAR)  LIB(&LIBVAR)
```

This command retrieves the SBMJOB and LIB values for the daily backup into the CL variables SBMJOBVAR and LIBVAR respectively.

Top

Error messages

*ESCAPE Messages

CPF1EE3

Not authorized to backup options.

CPF1E6C

Backup options in use.

CPF1E67

Backup options and library backup list damaged.

CPF1E99

Unexpected error occurred.

Top

Retrieve Binder Source (RTVBNDSRC)

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

Parameters
Examples
Error messages

The Retrieve Binder Source (RTVBNDSRC) command can be used to retrieve the exports from a set of modules, a service program, or both, and place them (along with the binder language statements needed for the exports) in a specified file member. This file member can later be used as input to the Create Service Program (CRTSRVPGM) command SRCMBR parameter. After the binder language has been retrieved into a source file member, you can edit the binder language to make changes as needed.

By default, the CRTSRVPGM command has a binder language file specified on the EXPORT and SRCFILE parameters to identify the exports from the service program. The RTVBNDSRC command can be useful in helping you automatically create this binder language.

Restrictions:

- You must have use (*USE) authority to the Create Source Physical File (CRTSRCPF) command, if the file does not exist.
- You must have *USE authority to the Reorganize Physical File Member (RGZPFM) command.
- You must have *USE authority to the Add Physical File Member (ADDPFM) command, if the member does not exist.
- You must have *USE authority to the modules or service program from which the exports are being retrieved.
- You must have execute (*EXECUTE) authority to the libraries in which the modules or service program exist.
- If the source file and member to receive the binder language exist, you must have change (*CHANGE) authority and either object alter (*OBJALTER) or object management (*OBJMGT) authority to the file, and *EXECUTE authority to the library that contains the file.
- If the source file exists but the source member needs to be created, you must have *CHANGE authority and either *OBJALTER or *OBJMGT authority to the file, and *EXECUTE, read (*READ) and add (*ADD) authorities to the library that contains the file.
- If the source file and member need to be created, you must have *EXECUTE, *READ and *ADD authorities to the library.

Top

Parameters

Keyword	Description	Choices	Notes
MODULE	Module	Values (up to 300 repetitions): <i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Module	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *USRLIBL</i>	
SRVPGM	Service program	<i>Qualified object name</i>	Optional
	Qualifier 1: Service program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	

Keyword	Description	Choices	Notes
SRCFILE	Export source file	<i>Qualified object name</i>	Optional
	Qualifier 1: Export source file	<i>Name, <u>QSRVSR</u></i>	
	Qualifier 2: Library	<i>Name, *<u>LIBL</u>, *CURLIB</i>	
SRCMBR	Export source member	<i>Name, *<u>DFT</u></i>	Optional
MBROPT	Replace or add records	<i>*ADD, *<u>REPLACE</u></i>	Optional

Top

Module (MODULE)

Specifies the list of modules from which to retrieve the exported symbols. If duplicate module and library specifications are found, only the first instance of the duplicate module and library is used.

At least one value must be specified for this parameter or the **Service program (SRVPGM)** parameter.

You can specify up to 300 values for this parameter.

Qualifier 1: Module

***ALL** The exported symbols from all of the modules in the specified library are retrieved.

generic-name

Specify a generic module name from which to retrieve the exported symbols. All modules that have names with the same prefix in the specified library or libraries are used. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

name Specify the name of the module from which to retrieve the exported symbols.

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.

***USRLIBL**

Only the libraries in the user portion of the job's library list are searched.

name Specify the name of the library to be searched.

Top

Service program (SRVPGM)

Specifies the service program from which to retrieve the exported symbols. At least one value must be specified for this parameter or the **Module (MODULE)** parameter.

Qualifier 1: Service program

name Specify the name of the service program from which to retrieve the exported symbols.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Export source file (SRCFILE)

Specifies the source file that is to hold the binder language for the exported symbols. If the source file does not exist, it is created.

Notes:

- Only a database physical file of type *SRC can be specified. Distributed data management (DDM) files are not supported.
- If the source file to receive the binder language exists, its record length must be a minimum of 92 bytes.

Qualifier 1: Export source file

QSRVSRC

The source file name is QSRVSRC.

name Specify the name of the source file.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found. If a source file by the name specified is not found in the library list, it is created in the current library. If there is no current library, the QGPL library is used.

***CURLIB**

The current library for the job is searched. If a source file by the name specified does not exist, it is created in the current library. If there is no current library, the QGPL library is used.

name Specify the name of the library to be searched. If a source file by the name specified is not found in this library, the source file is created in this library.

Top

Export source member (SRCMBR)

Specifies the member in the source file that is to hold the binder language for the exported symbols. Only one source member will contain the binder language.

If the member does not exist in the source file specified, the member is created.

***DFT** The name of the source file member is taken from the value specified for the **Service program (SRVPGM)** parameter, if a service program is specified. Otherwise, the name of the source file member is taken from the value specified for the **:Module (MODULE)** parameter,

- If only one module is specified, the name of that module is the member name used.
- If more than one module is specified, the name of the first module specified is used.

- If the value *ALL or a generic name is specified, the module name of the first occurrence found is the source member name used.

name Specify the name of the member that will contain the generated source binder language.

Top

Replace or add records (MBROPT)

Specifies whether the generated binder language statements are replaced or added to the existing statements.

*REPLACE

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

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

Note: If the member already exists and, for example, already contains STRPGMEXP and ENDPGMEXP statements, the member may contain multiple STRPGMEXP and ENDPGMEXP statements in the binder language at the end of this operation. You must edit these multiple statements in order to use the binder language with the Create Service Program (CRTSRVPGM) command.

Top

Examples

```
RTVBNDSRC  MODULE(MYLIB/*ALL)
           SRCFILE(MYLIB/MYBINDFILE)  MBROPT(*ADD)
```

This command retrieves the exports from all modules in the library MYLIB, and places them in the source member with the name of the first module found. If this source member does not exist in the file MYBINDFILE in the library MYLIB, it is created. The export statements are added to the end of the member. If multiple start and end program export statements exist in the file when this command is ended, the source member must be edited before it is used to create a service program. Either the extra STRPGMEXP, ENDPGMEXP statements can be removed, or the PGMLVL parameter can be added to the STRPGMEXP statements, if some of the export blocks are for previous versions of the service program.

Top

Error messages

*ESCAPE Messages

CPF5CA5

Record length too small for data base source file.

CPF5D06

Not authorized to library &2, or file &1 in library &2, or member &3.

CPF5D08

File &1 in library &2 not supported file type.

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.

CPF9832

Function not supported for DDM file &2.

CPF9834

Input file &1 in library &2 is not a source file.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9899

Error occurred during processing of command.

Top

Retrieve Configuration Source (RTVCFGSRC)

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

Parameters
Examples
Error messages

The Retrieve Configuration Source (RTVCFGSRC) command is used to retrieve the CL source statements which describe existing configuration objects. These source statements are placed into a source file member. The CL source may be used to re-create the configuration objects.

Top

Parameters

Keyword	Description	Choices	Notes
CFGD	Configuration description	Single values: *ALL Other values (up to 256 repetitions): <i>Generic name, name</i>	Required, Positional 1
CFGTYPE	Type	*ALL, *NWS D, *NWID, *LIND, *CTLD, *DEVD, *MODD, *COSD, *CNNL, *NTBD	Required, Positional 2
SRCFILE	Source file	<i>Qualified object name</i>	Optional, Positional 3
	Qualifier 1: Source file	<i>Name, <u>QCLSRC</u></i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *CFGD</i>	Optional, Positional 4
RTVOPT	Retrieve option	*NET, *OBJ	Optional
MBROPT	Member option	*ADD, *REPLACE	Optional
TEXT	Text 'description'	<i>Character value, *CFGDTXT, *BLANK</i>	Optional

Top

Configuration description (CFGD)

Specifies the names of configuration objects to be retrieved.

This is a required parameter.

***ALL** All configuration objects of the specified configuration type (CFGTYPE parameter) are retrieved.

generic-configuration-object-name

Specify the generic name of the configuration description name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

configuration-description-name

Specify the user-defined name of the configuration description.

Up to 256 names can be specified. These can be any combination of configuration object and generic names.

You can enter multiple values for this parameter.

Top

Type (CFGTYPE)

Specifies the type of configuration object to be retrieved.

***ALL** All network server descriptions, network interfaces, lines, controllers, devices, connection lists, modes, classes-of-service, and NetBIOS descriptions matching the specified names are retrieved in the following order:

1. Connection Lists
2. Network server descriptions
3. Network Interfaces
4. Non-TDLC line descriptions
5. Non-TDLC controller descriptions
6. TDLC line descriptions
7. TDLC controller descriptions
8. Device descriptions
9. Mode descriptions
10. Class-of-service descriptions
11. NetBIOS descriptions
12. SWTCTLLST for line descriptions
13. SWTLINLST for controller descriptions
14. SWTNWILST for line descriptions
15. Printer for remote displays

***NWSD**

All network server descriptions matching the specified names are retrieved.

***NWID**

All network interface descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***LIND**

All line descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***CTLD**

All controller descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***DEV D**

All device descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***MODD**

All mode descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***COSD**

All class-of-service descriptions that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***CNNL**

All connection lists that match the name specified on the **Configuration description** prompt (CFGD parameter) are retrieved.

***NTBD**

All NetBIOS descriptions matching the specified names are retrieved.

Top

Source file (SRCFILE)

Specifies the name and library of the previously created database source file that contains the source file member.

The possible **source-file** values are:

QCLSRC

The source file named QCLSRC is used.

source-file

Specify the name of a source file.

The possible library values are:

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

***CURLIB**

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

library-name

Specify the library where the source file is located.

Top

Source member (SRCMBR)

Specifies the name of the database source file member into which the CL source statements are written.

***CFGD**

The name of the source member is the configuration object name if there is only one name specified and the name is not generic or *ALL. Otherwise, the name is CFGSRC.

source-member-name

Specify the name of the source file member that contains the CL source statements.

Top

Retrieve option (RTVOPT)

Specifies which attachment information is retrieved for the specified objects.

***NET** For network interfaces, each network interface description and the descriptions of its attached nonswitched lines, controllers, and devices are retrieved. For lines, each line description and the descriptions of its attached nonswitched controllers and devices are retrieved. For controllers, each controller description and the descriptions of its attached devices are retrieved. For network interfaces, lines, and controllers, switched attachment information is also retrieved.

***OBJ** Each specified configuration object description is retrieved with all attachment information.

RTVOPT is ignored if the CFGTYPE parameter is *ALL.

Top

Member option (MBROPT)

Specifies whether the new records replace or are added to the existing records.

*REPLACE

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

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

Top

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

*CFGDTXT

The text description is the same as the text description of the object specified by the CFGD parameter unless there is more than one object specified or the name is a generic name. In this case, *BLANK is used for the text description.

***BLANK**

No text is specified.

'description'

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

Top

Examples

```
RTVCFGSRC  CFGD(CTL*)  CFGTYPE(*CTLD)
           SRCMBR(CTLS) RTVOPT(*OBJ)
```

This command places CL source statements in the file member CTLS in the source file QCLSRC. These source statements can be used to re-create object descriptions for all existing controllers with names beginning with CTL.

Top

Error messages

*ESCAPE Messages

CPF2207

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

CPF263E

File member &1 contains its maximum sequence number.

CPF263F

No objects found.

CPF264A

Record length of &1 in library &2 is too small.

CPF264C

Source file member &1 not found.

CPF9810

Library &1 not found.

CPF9820

Not authorized to use library &1.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9848

Cannot open file &1 in library &2 member &3.

CPF9849

Error while processing file &1 in library &2 member &3.

Top

Retrieve Configuration Status (RTVCFGSTS)

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

Parameters
Examples
Error messages

The Retrieve Configuration Status (RTVCFGSTS) command provides a CL program with a numeric value that represents the status of a configuration object.

Top

Parameters

Keyword	Description	Choices	Notes
CFGD	Configuration description	<i>Name</i>	Required, Positional 1
CFGTYPE	Type	*NWI, *NWS, *LIN, *CTL, *DEV	Required, Positional 2
STSCDE	CL variable for status code	<i>Decimal number</i>	Required, Positional 3

Top

Configuration description (CFGD)

Specifies the name of the configuration description for which the status will be retrieved.

This is a required parameter.

Top

Type (CFGTYPE)

Specifies the type of description for which you want to retrieve status.

- *NWS The object is a network server description.
- *NWI Status for a network interface is retrieved.
- *LIN Status for a line is retrieved.
- *CTL Status for a controller is retrieved.
- *DEV Status for a device is retrieved.

This is a required parameter.

Top

CL variable for status code (STSCDE)

Specifies the name of a variable which will contain the retrieved status. In a CL program, this should be a decimal variable of length (5 0).

The possible values which can be returned are:

Value	Definition
-------	------------

0	VARIED OFF - The system is not using the description.
10	VARY OFF PENDING - The description is being varied off. During this time the system may be taking down the tasks which managed the resource, and so on.
20	VARY ON PENDING - The description is being varied on. During this time the system may be putting tasks in place to manage the resource, downloading a program to an I/O processor, communicating with data circuit-terminating equipment (DCE), and so on.
30	VARIED ON - The tasks that manage the network interface, network server description, line, controller, or device have been put in place by the system, and the system has the capability to communicate with it.
32	VARY ON/CNN PENDING - The first of a pair of OptiConnect controllers is varied on but its attached device is not yet in a varied on state.
40	CONNECT PENDING - This status is only valid for switched SDLC, IDLC, BSC, or asynchronous lines. The line is in this status while waiting for the switched connection to be established; this can be either a dial or an answer connection.
50	SIGNON DISPLAY - This status is only valid for display devices. The system is preparing the device to receive the sign-on display, or sending the sign-on display, or the actual sign-on display is at the display station.
51	ACTIVE/CNN PENDING - The first of a pair of OptiConnect controllers and its attached device are varied on and waiting for the OptiConnect path to be established.
60	ACTIVE - The object is successfully placed in VARIED ON status. In addition: for network interfaces and the network server descriptions, one or more attached lines is in a VARY ON PENDING status or higher; for lines, one or more attached controllers is in a VARY ON PENDING status or higher; for controllers, one or more attached devices is in a VARY ON PENDING status or higher; for devices, active status varies depending on the type of device — more information is in the Communications Configuration book, SC41-5401 book. A display device which is at a second sign-on display as a result of pressing the system request key will be considered ACTIVE.
63	ACTIVE READER - The device is in use by a spool reader.
66	ACTIVE WRITER - The device is in use by a spool writer.
67	AVAILABLE - The independent auxiliary storage pool (ASP) device is available for use without function restrictions.
70	HELD - This status is only valid for Device Descriptions. The user or the system held the communications device to prevent it from communicating. The Release Communications Device (RLSCMNDEV) command can be used to release the device.
80	RCYPND - Error recovery is pending for the line, controller, or device. A message indicating what error occurred appears on the QSYSOPR message queue.
90	RCYCNL - Error recovery is canceled for the network interface, line, controller, or device. An error occurred, and the operator replied with a C (to cancel error recovery) to a message, or the operator used a command (ENDNWIRCY, ENDLINRCY, ENDCTLR CY, ENDDEVRCY) to end error recovery.

- 95 SYSTEM REQUEST - The display device has been requested by the system and its associated job has been suspended. This occurs as a result of a user pressing the System Request key.
- 100 FAILED - An error occurred for the network interface, network server description, line, controller, or device that can be recovered only by varying off and on again.
- 103 FAILED READER - An error occurred for the device while in use by a spool reader.
- 106 FAILED WRITER - An error occurred for the device while in use by a spool writer.
- 107 SHUTDOWN - The NWSD was shut down using an Application Program Interface (API).
- 110 DIAGNOSTIC MODE - The network interface, network server description, line, controller, or device resource is being used by problem analysis procedures to diagnose problems, and the resource cannot be used by other users.
- 111 DAMAGED - The network interface, network server description, line, controller, or device description is damaged. This is a system error condition. Information indicating when this damage occurred appears in the history log (QHST). Further information may be in the vertical licensed internal code (VLIC) logs. The description must be deleted and created once more before it can be used again.
- 112 LOCKED - The actual status of the resource cannot be determined because another job has an exclusive lock on the description. Retry at a later time, or use the Work With Object Lock (WRKOBJLCK) command to determine which job has the lock on the description.
- 113 UNKNOWN - The status indicator of the description cannot be determined. This is a system error condition. Use the Dump Object (DMPOBJ) command to dump the contents or attributes of the description to a spooled printer file, and contact your IBM representative.

This is a required parameter.

More information is in the Communications Configuration book, SC41-5401.

Top

Examples

```
RTVCFGSTS CFGD(ND01) CFGTYPE(*LIN) STSCDE(&STSCODE)
```

This command retrieves the configuration status of the line configuration description ND01 for use in the CL variable &STSCODE.

Top

Error messages

*ESCAPE Messages

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

Top

Retrieve C Locale Description (RTVCLDSRC)

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

Parameters
Examples
Error messages

The Retrieve C/400* Locale Description Source (RTVCLDSRC) command retrieves the locale that you specified on the SRCFILE parameter. The *SYSVAL locale can be retrieved and used as a model to create a tailored locale description. The locale commands and formats that you use in your locale source are outlined in the C/400 User's Guide. Once your locale source is complete, use the CRTCLD command to process the description.

Error messages for RTVCLDSRC

*ESCAPE Messages

PSE1708

Member &3 is not found in the specified file and library.

PSE1709

File &1 is not found in library &2.

PSE1739

The necessary storage could not be allocated.

PSE1751

Source &1 in file &3 was not retrieved.

Top

Parameters

Keyword	Description	Choices	Notes
CLD	Locale name	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Locale name	<i>Name, *SYSVAL</i>	
	Qualifier 2: Library	<i>Name, *CURLIB</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Optional, Positional 2
	Qualifier 1: Source file	<i>Name, <u>QCLDSRC</u></i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *CLD</i>	Optional, Positional 3
TEXT	Text description	<i>Character value, *SRCMBRTXT, *BLANK</i>	Optional, Positional 4

Top

Locale name (CLD)

Specifies the C/400 locale description that is to be retrieved. You can retrieve the system values if you use *SYSVAL, instead of specifying a locale name and library.

locale name

Enter the name of the locale description.

***SYSVAL**

The system default values are used as the basis for the retrieved locale description source. The default locale description values are used for any locale description information that does not have a corresponding system value.

The possible library values are:

***CURLIB**

The current library will be searched for the source file specified. If you have not specified a current library, QGPL will be used.

library-name

Enter the name of the library containing the locale description.

Top

Source file (SRCFILE)

Specifies the name and library of the file that receives the retrieved C/400 locale description source. If the specified library does not exist, the process ends. If the specified file does not exist, it is created.

QCLDSRC

The default source file name for the C locale description source. Use this default if you want to browse your source and its associated listing using SEU.

source-file-name

Enter the name of the file where you want to store the C locale description source.

The possible library values are:

***LIBL** The system searches the library list for the library that contains the locale description source file.

***CURLIB**

The current library is used to store the retrieved locale. If you have not specified the current library, QGPL will be used.

library-name

Enter the name of the library to store the retrieved locale.

Top

Source member (SRCMBR)

Specifies the source file member that will store the C/400 locale description source after processing. If the specified member does not exist, it is created.

***CLD** Specifies that the C locale description name specified by the CLD parameter is used as the source file member name. For example, if you retrieved the system values, the locale source is stored in a member called SYSVAL.

source-file-member-name

Enter a name for the source file member that will store the retrieved C locale description.

Top

Text description (TEXT)

Specifies the text you want associated with the source file member you are processing.

*SRCMBRTXT

Specifies that the same text description you supplied for the C locale description is used.

*BLANK

No text appears.

'description'

Enter the descriptive text to accompany the retrieved C locale description. You can enter up to 50 characters of text.

Top

Examples

None

Top

Error messages

*ESCAPE Messages

PSE1708

Member &3 is not found in the specified file and library.

PSE1709

File &1 is not found in library &2.

PSE1739

The necessary storage could not be allocated.

PSE1751

Source &1 in file &3 was not retrieved.

Top

Retrieve Cleanup (RTVCLNUP)

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

Parameters
Examples
Error messages

The Retrieve Cleanup (RTVCLNUP) command retrieves a cleanup operation value for use in a CL or REXX program. The value is returned (copied) to the specified CL variable in the program.

Top

Parameters

Keyword	Description	Choices	Notes
ALWCLNUP	CL var for ALWCLNUP (4)	Character value	Optional
STRTIME	CL var for STRTIME (10)	Character value	Optional
USRMSG	CL var for USRMSG (5)	Character value	Optional
SYSMSG	CL var for SYSMSG (5)	Character value	Optional
CRITSYSMSG	CL var for CRITSYSMSG (5)	Character value	Optional
SYSPRT	CL var for SYSPRT (5)	Character value	Optional
SYSLOG	CL var for SYSLOG (5)	Character value	Optional
JOBQ	CL var for JOBQ name (10)	Character value	Optional
JOBQLIB	CL var for JOBQ library (10)	Character value	Optional
RUNPTY	CL var for RUNPTY (2 0)	Decimal number	Optional
JRNRCVSIZ	CL var for JRNRCVSIZ (7 0)	Decimal number	Optional
CALITM	CL var for CALITM (5)	Character value	Optional

Top

CL var for ALWCLNUP (4) (ALWCLNUP)

Specifies the name of the CL variable that receives the allow cleanup value. The variable named has a minimum length of 4 characters. A '*YES' is returned if the cleanup operation is allowed to run. Otherwise, an '*NO' is returned.

Top

CL var for STRTIME (10) (STRTIME)

Specifies the name of the CL variable that receives the time cleanup starts each day. The variable named has a minimum length of 10 characters. A special value of '*NONE' or '*SCDPWROFF,' or the start time is returned.

Top

CL var for USRMSG (5) (USRMSG)

Specifies the name of the CL variable that receives the value for cleaning up user messages on user profile message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days user messages are kept before they are deleted is returned.

[Top](#)

CL var for SYSMSG (5) (SYSMSG)

Specifies the name of the CL variable that receives the value for cleaning up messages on the QSYSOPR message queue and on work station message queues. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system messages are kept before they are deleted is returned.

[Top](#)

CL var for CRITSYSMSG (5) (CRITSYSMSG)

Specifies the name of the CL variable that receives the value for cleaning up messages on the QSYSMSG message queue. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days critical system messages are kept before they are deleted is returned.

[Top](#)

CL var for SYSPRT (5) (SYSPRT)

Specifies the name of the CL variable that receives the value for cleaning up job logs and other system output. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days job logs are kept before they are deleted is returned.

[Top](#)

CL var for SYSLOG (5) (SYSLOG)

Specifies the name of the CL variable that receives the value for cleaning up system journals, history files, problem log files, alert database, and program temporary fixes. The variable named has a minimum length of 5 characters. The special value *KEEP or the number of days system journals and system logs are kept before they are deleted is returned.

[Top](#)

CL var for JOBQ name (10) (JOBQ)

Specifies the name of the CL variable that receives the name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters. The name of the job queue under which cleanup batch jobs are run is returned.

[Top](#)

CL var for JOBQ library (10) (JOBQLIB)

Specifies the name of the CL variable that receives the library name of the job queue to which the cleanup batch jobs are submitted. The variable named has a minimum length of 10 characters.

Top

CL var for RUNPTY (2 0) (RUNPTY)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the run (or processing) priority for routing steps that are part of the job. For additional information on run priority, refer to this parameter description in the CHGJOB (Change Job) command. The variable must be a 2-digit decimal variable specified with no decimal positions.

Top

CL var for JRNRCVSIZ (7 0) (JRNRCVSIZ)

Specifies the name of the CL variable that receives the value for the journal receiver size ranging from 1 through 1,919,999 (KB). When the size of the space for the journal receiver is larger than the size specified by this value, Operational Assistant (OA) automatic cleanup function will detach it. The variable must be an 7-digit decimal variable specified with no decimal positions.

Top

CL var for CALITM (5) (CALITM)

In V5R1 and later releases, this parameter is not supported.

Top

Examples

Example 1: Retrieving Number of Days Messages are Kept

```
DCL VAR(&UMSGDAYS) TYPE(*CHAR) LEN(5)
RTVCLNUP USRMSG(&UMSGDAYS)
```

These commands retrieve the number of days that user messages are kept before being deleted.

Example 2: Retrieving Time Cleanup Operation Starts

```
DCL VAR(&CLNUPTIME) TYPE(CHAR) LEN(10)
RTVCLNUP STRTIME(&CLNUPTIME)
```

These commands retrieve the time that the cleanup operation starts.

Example 3: Retrieving Run Priority

```
DCL VAR(&RPTY) TYPE(*DEC) LEN(2 0)
RTVCLNUP RUNPTY(&RPTY)
```

This command retrieves the run priority for the cleanup job. The run priority number is copied into the CL variable &RPTY. The variable must be a 2-digit decimal variable with no decimal positions.

Example 4: Retrieving Journal Receiver Size

```
DCL VAR(&JRNSIZ) TYPE(*DEC) LEN(7 0)
RTVCLNUP JNRCSIZ(&JRNSIZ)
```

This command retrieves the journal receiver size for the cleanup job. The journal receiver size is copied into the CL variable &JRNSIZ. The variable must be a 7-digit decimal variable with no decimal positions.

Top

Error messages

***ESCAPE Messages**

CPF1E2B

Power scheduler and cleanup options not found.

CPF1E33

Cleanup options or power schedule in use by another user.

CPF1E99

Unexpected error occurred.

Top

Retrieve CL Source (RTVCLSRC)

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

Parameters
Examples
Error messages

The Retrieve CL Source (RTVCLSRC) command is used to retrieve the source statements from an OPM CL program that were used to compile that program. These source statements are placed into a source file member, which can be used as input when recompiling the CL program.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Program	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *PGM</i>	Optional, Positional 3
RTVINC SRC	Retrieve included source	<i>*NO, *YES</i>	Optional

Top

Program (PGM)

Specifies the OPM CL program whose source is to be retrieved.

Note: RTVCLSRC cannot be used to retrieve the source for ILE CL programs and modules.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program.

Qualifier 2: Library

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

*CURLIB

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

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

Top

Source file (SRCFILE)

Specifies the database source file into which the CL source statements are to be written. This file must exist when the command is run.

This is a required parameter.

Qualifier 1: Source file

name Specify the name of the source file.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the database source file member into which the CL source statements are to be written. If the name is not specified, the CL program name is used. If the member existed before the command was run, it is cleared before any source statements are written into it. If the member did not exist, it is created.

***PGM** The name of the CL program is used as the member name.

name Specify the name of the source file member that will contain the CL source statements.

Top

Retrieve included source (RTVINCSRC)

Specifies whether or not to retrieve CL commands that were included by INCLUDE (Include CL Source) commands in the original source CL program.

***NO** If the original CL source program contained any INCLUDE commands, the retrieved source will contain the same INCLUDE commands, but will not contain the CL commands which were included when the CL source was compiled.

Specifying *NO for this parameter will produce CL source which most closely matches the original CL source.

***YES** If the original CL source program contained any INCLUDE commands, the retrieved source will contain the included CL source commands instead of the original INCLUDE commands.

Specifying *YES for this parameter will produce CL source which most closely matches the full CL source used to create the CL program.

Top

Examples

Example 1: Retrieving the Original CL Source

```
RTVCLSRC  PGM(JOHN1/TEXT1) SRCFILE(JOHN2) SRCMBR(JOHN3)
```

This command retrieves the source statements from the CL program named TEXT1 in library JOHN1. The retrieved source statements are placed into the file named JOHN2, and are named as member JOHN3. If the original CL source program contained any INCLUDE commands, the retrieved source will contain the same INCLUDE commands.

Example 2: Retrieving the Expanded CL Source

```
RTVCLSRC  PGM(MYAPP1) SRCFILE(RTVAPPSRC) RTVINCSRC(*YES)
```

This command retrieves the source statements from the CL program named MYAPP1 located using the library list. The retrieved CL source statements are placed into member MYAPP1 of source file RTVAPPSRC, which is also located using the library list. If the original CL source contained INCLUDE commands, the retrieved CL source will contain the included CL source in place of the INCLUDE commands.

Top

Error messages

*ESCAPE Messages

CPF0560

Program &1 in &2 not a CL program.

CPF0561

Unable to retrieve CL source from CL program &2.

CPF0562

File &1 in &2 not a data base source file.

CPF0563

Record length too small for data base source file.

CPF0564

Unable to add data base member &3 to file.

CPF0565

Source from CL program &4 not retrieved.

CPF0566

Source not available for CL program &1 in &2.

CPF9801

Object &2 in library &3 not found.

CPF9803

Cannot allocate object &2 in library &3.

CPF9805

Object &2 in library &3 destroyed.

CPF9806

Cannot perform function for object &2 in library &3.

- CPF9807**
One or more libraries in library list deleted.
- CPF9808**
Cannot allocate one or more libraries on library list.
- CPF9809**
Library &1 cannot be accessed.
- CPF9810**
Library &1 not found.
- CPF9811**
Program &1 in library &2 not found.
- CPF9820**
Not authorized to use library &1.
- CPF9821**
Not authorized to program &1 in library &2.
- CPF9822**
Not authorized to file &1 in library &2.
- CPF9830**
Cannot assign library &1.
- CPF9848**
Cannot open file &1 in library &2 member &3.
- CPF9849**
Error while processing file &1 in library &2 member &3.

Top

Retrieve Current Directory (RTVCURDIR)

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

Parameters
Examples
Error messages

The Retrieve Current Directory (RTVCURDIR) command is used in a control language (CL) program to retrieve the name of the current directory into the specified CL variable. An absolute path name containing no symbolic links is retrieved. The length of the name of the current directory is also retrieved.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

- Execute (*X) authority is required to the current directory and the user must have read and execute (*RX) authority to each directory in the path.
- This command is valid only within a CL program.
- The maximum length of a directory name that can be retrieved is limited by the maximum length of a character variable.

Note: The maximum length of a character variable cannot exceed 9999 bytes.

Top

Parameters

Keyword	Description	Choices	Notes
RTNDIR	CL var for RTNDIR (9999)	<i>Character value</i>	Required, Positional 1
DIRNAMLEN	CL var for DIRNAMLEN (7 0)	<i>Decimal number</i>	Required, Positional 2

Top

CL var for RTNDIR (9999) (RTNDIR)

Specifies the name of the CL variable that receives the name of the current directory. The variable must be a character variable. If the current directory name has fewer characters than the variable allows, the value is not padded.

Top

CL var for DIRNAMLEN (7 0) (DIRNAMLEN)

Specifies the name of the CL variable that receives the length (in bytes) of the current directory name. This length can be longer than the length of the character variable to receive the directory name. The variable must be a 7-digit decimal variable specified with no decimal positions.

Examples

Example 1: Retrieving the Current Directory

```
RTVCURDIR  RTNDIR(&CD)  DIRNAMLEN(&CDLEN)
```

This command retrieves the name of the current directory and the length of the name of the current directory into the &CD and &CDLEN CL program variables.

Error messages

*ESCAPE Messages

CPFA085

Home directory not found for user &1.

CPFA09C

Not authorized to object. Object is &1.

CPFA0A1

An input or output error occurred.

CPFA0A9

Object not found. Object is &1.

Retrieve Directory Information (RTVDIRINF)

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

Parameters
Examples
Error messages

The Retrieve Directory Information (RTVDIRINF) command is used to collect attributes for directories and files in the Integrated File System. The collected information is stored in database files that are named using the information file prefix specified by the INFFILEPFX parameter. The files are created in the library specified by the INFLIB parameter.

You can run the Print Directory Information (PRTDIRINF) command to print reports using the retrieved directory information.

To get the most accurate results, this command should be run at a time when there is very little activity for files in the specified directory. If SUBTREE(*ALL) is specified, try to run this command when there is very little activity for files in all subdirectories of the specified directory.

Note: If there is more than one member in the files, the results of running this command can be unpredictable.

Restrictions:

- Directory information can be retrieved only for mounted file systems.
- Directory information can be retrieved only for local file systems.
- You must have all object (*ALLOBJ) special authority to run this command.

Top

Parameters

Keyword	Description	Choices	Notes
DIR	Directory	<i>Path name</i>	Required, Positional 1
SUBTREE	Directory subtree	<u>*ALL</u> , *NONE	Optional
INFFILEPFX	Information file prefix	<i>Simple name</i> , <u>*GEN</u>	Optional
INFLIB	Information library	<i>Name</i> , <u>QUSRSYS</u>	Optional

Top

Directory (DIR)

Specifies the path name of the integrated file system object for which information will be collected.

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

This is a required parameter.

directory-path-name

Specify the path name of the integrated file system object.

Top

Directory subtree (SUBTREE)

Specifies whether or not to collect directory information for subdirectories of the directory specified by the DIR parameter.

***ALL** Directory information for subdirectories of the specified directory will be collected.

***NONE**

Directory information will only be collected for the files in the specified directory. No directory information will be collected for subdirectories.

Top

Information file prefix (INFFILEPFX)

Specifies the file name prefix of the database files where the retrieved directory information is to be stored.

***GEN** The database files will be created with a unique prefix generated by this command. The prefix will begin with QAEZD followed by four digits. The files created to store the collected information will be named using this prefix followed by either the letter 'D' (for the file which contains directory information) or the letter 'O' (for the file that contains information about objects in directories). For example, the first time the command is run with *GEN specified, files QAEZD0001D and QAEZD0001O will be created in the library specified by the INFLIB parameter. Informational message CPI1E30 will be sent to the job log and will contain the names of the files created.

information-file-prefix

Specify the file prefix to use for creating the database files used to store the collected directory information. The prefix can be up to nine letters. The files created to store the collected information will be named using this prefix followed by either the letter 'D' (for the file which contains directory information) or the letter 'O' (for the file that contains information about objects in directories). For example, if the prefix specified is MYDIR, database files MYDIRD and MYDIRO will be created in the library specified by the INFLIB parameter.

Top

Information library (INFLIB)

Specifies the library where the database files used to store the directory information will be created.

QUSRSYS

The files will be created in library QUSRSYS.

library-name

Specify the name of the library to create the database files.

Top

Examples

Example 1: Retrieve Information, Including Subdirectories

```
RTVDIRINF  DIR('/MYDIR/MYDOCS') SUBTREE(*ALL)
           INFFILEPFX(*GEN) INFLIB(QUSRSYS)
```

This command retrieves directory information about directory /MYLIB/MYDOCS, including information for all nested subdirectories, and stores it in the database files created in library QUSRSYS. The database files will be created with unique names that begin with 'QAEZD' followed by four digits. If this is the first time the RTVDIRINF command is run, the file names will be QAEZD0001O and QAEZD0001D.

Example 2: Retrieve Information for Specified Directory Only

```
RTVDIRINF  DIR('/') SUBTREE(*NO) INFFILEPFX(MYROOTDIR)
           INFLIB(MYLIB)
```

This command retrieves directory information about the root directory without inspecting nested subdirectories and stores it in the database files MYROOTDIRO and MYROOTDIRD in library MYLIB. If database files with either of those names already exist in library MYLIB, an error message will be sent and no directory information will be retrieved.

Top

Error messages

*ESCAPE Messages

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.

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.

CPFA0B2

No objects satisfy request.

CPF1ED2

File &1 is in use and cannot be accessed.

CPF1ED4

Not authorized to collect directory information.

CPF1E99

Unexpected error occurred.

[Top](#)

Retrieve DLO Authority (RTVDLOAUT)

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

Parameters
 Examples
 Error messages

The Retrieve Document Library Object Authority (RTVDLOAUT) command retrieves the authority assigned to a folder or filed document.

Restrictions:

- A user must be in the system distribution directory entry to retrieve the various authorities.
- A user must have at least use (*USE) authority to the filed document or folder.
- A user with less than all (*ALL) authority to the filed document or folder will only be able to retrieve that user's authority or the owner.
- A user must have *ALL authority to the object, or all object (*ALLOBJ) special authority, or be the owner of the filed document or folder, to retrieve all the authorities.
- A user must have *ALLOBJ special authority to retrieve the *ROOT folder public authority.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value, *DOCID, *LADNTSP, *SYSOBJNAM, *ROOT	Required, Positional 1
FLR	Folder	Character value, *NONE	Optional, Positional 2
DOCID	Document identifier	Character value, *NONE	Optional
LADNTSP	LADN timestamp	0000000000000001-FFFFFFFFFFFFFFFF, *NONE	Optional
SYSOBJNAM	System object name	Name, *NONE	Optional
STRUSRAUTE	Starting user authority entry	1-99999, <u>1</u>	Optional
OWNER	CL var for OWNER (10)	Character value	Optional
AUTL	CL var for AUTL (10)	Character value	Optional
SENSITIV	CL var for SENSITIV (20)	Character value	Optional
PUBAUT	CL var for PUBAUT (10)	Character value	Optional
CHKOUTUSR	CL var for CHKOUTUSR (32)	Character value	Optional
ACC	CL var for ACC (220)	Character value	Optional
USRAUT	CL var for USRAUT (1020)	Character value	Optional
GRPAUT	CL var for GRPAUT (340)	Character value	Optional
PGP	CL var for PGP (10)	Character value	Optional

Top

Document library object (DLO)

Specifies the name of the document or folder for which authorities are retrieved.

***ROOT**

The root folder contains all first-level folders. If folder *ROOT is specified only public authority will be returned by PUBAUT parameter. No other authorities will be returned.

***LADNTSP**

The timestamp from the library-assigned document name (LADN) specified on the LADNTSP parameter is used to identify the document or folder.

***SYSOBJNAM**

The system object name specified on the SYSOBJNAM parameter is used to identify the document or folder.

name Specify the user-assigned name of the document or folder.

Top

Folder (FLR)

Specifies the name of the folder where the object specified on the DLO parameter is located.

***NONE**

The name of folder that contains the object is not specified, the object is not contained in a folder, or the object is specified using the LADNTSP, or SYSOBJNAM parameter, or *ROOT was specified in the DLO parameter.

name Specify the name of the folder that contains the object.

Note: FLR(*NONE) must be specified if the object is a first-level folder.

Top

Document identifier (DOCID)

Specifies the library-assigned name of the document or folder.

***NONE**

The object is not identified using its document identifier (DOCID)

document-identifier

Specify the document identifier of the document or folder. The document identifier is 24 hexadecimal characters in length in the format YYYYMMDDHHMNSSHSSNSNSNSN, where:

- YYYY = year
- MM = month
- DD = day
- HH = hour
- MN = minute
- SS = second
- HS = hundredths of a second
- SNSNSNSN = system name

Top

LADN timestamp (LADNTSP)

Specifies the LADN timestamp of the document or folder.

*NONE

The object is not identified using its LADN timestamp.

timestamp

Specify the LADN timestamp of the document or folder. The LADN timestamp is 16 hexadecimal characters in length in the format YYYYMMDDHHMNSSHS, where:

- YYYY = year
- MM = month
- DD = day
- HH = hour
- MN = minute
- SS = second
- HS = hundredths of a second

Top

System object name (SYSOBJNAM)

Specifies the system object name.

*NONE

The object is not identified using its system object name.

name Specify the 10-character system object name of the document or folder.

Top

Starting user authority entry (STRUSRAUTE)

Allows a user to specify the starting user authority entry number to use when retrieving specific user authorities (USRAUT parameter). STRUSRAUTE will enable retrieving specific user authorities for a DLO that has more than 50 specific user authorities. If no CL variable is provided for the USRAUT parameter, this parameter is ignored.

1 User authority will be returned starting with the first specific user authority.

2-99,999

User authority will be returned starting with the specified entry number. If the value is greater than the number of specific user authorities for the DLO, an error message will be sent and no specific user authorities will be returned.

Top

CL var for OWNER (10) (OWNER)

Specifies the name of a 10-character CL variable used to retrieve the owner of the selected document or folder.

Top

CL var for AUTL (10) (AUTL)

Specifies the name of a 10-character CL variable used to retrieve the authorization list assigned to the selected document or folder. The value *NONE is returned if no authorization list has been assigned.

CL var for SENSITIV (20) (SENSITIV)

Specifies the name of a 20-character CL variable used to retrieve the sensitivity assigned to the selected document or folder.

*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

CL var for PUBAUT (10) (PUBAUT)

Specifies the name of a 10-character CL variable used to retrieve the public authority assigned to the selected document or folder.

***USE** User can view, print, or copy the document or folder.

*CHANGE

User can perform all operations listed for *USE and can also edit and mark the document for offline storage.

***ALL** User can perform all operations on the document, except change ownership of the document or folder or give themselves authority to work with the document after authority has been revoked.

*EXCLUDE

All users who are not otherwise authorized to this document or folder are denied access.

USER DEF

Authority to this document is user-defined and is not one of the system-defined sets of authorities (*ALL, *CHANGE, *USE, *EXCLUDE).

*AUTL

Authority specified in the authorization list being used by this document should determine public authority.

Top

CL var for CHKOUTUSR (32) (CHKOUTUSR)

Specifies the name of a 32-character CL variable used to retrieve the user profile who has the document checked out and the user profile on whose behalf the document was checked out. If no user has checked out the document, the 32-character variable will be blank. If the document was not checked out by a user working on behalf of another user, the last 16 characters will be blanks.

Top

CL var for ACC (220) (ACC)

Specifies the name of a 200-character CL variable used to retrieve the access codes assigned to the specified document or folder.

Top

CL var for USRAUT (1020) (USRAUT)

Specifies the name of a 1020-character CL variable used to retrieve the specific user authority assigned to the document or folder. USRAUT will return a maximum of 50 specific authorities per invocation of the RTVDLOAUT command. If the document or folder has more than 50 specific user authorities associated with it, you can use the STRUSRAUTE parameter on subsequent invocations of RTVDLOAUT to return user authorities starting with the specified entry numbers.

Within the 1020-character CL variable are the total number of authorized users, the number of authority entries returned, the starting entry number and the ending entry number.

For example:

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR) USRAUT(&RTNUSRAUT) + STRUSRAUTE(1)
```

In the example above, MYDOC has 55 authorized users. The total number of specific user authorities will be 55. The number of authority entries returned will be 50. The starting authority entry number will be 1. The ending authority entry number will be 51.

Top

CL var for GRPAUT (340) (GRPAUT)

Specifies the name of a 340-character CL variable used to retrieve the group authority assigned to the specified document or folder. Only the groups associated to the user of this command will be returned.

Top

CL var for PGP (10) (PGP)

Specifies the name of a 10-character CL variable used to retrieve the primary group assigned to the specified document or folder. The value *NONE is returned if no primary group has been assigned.

Top

Examples

Example 1: Retrieve the Owner of a Document Library Object

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR) OWNER(&OWNER)
```

This command retrieves the owner of document library object MYDOC in folder MYFLR and stores the value in the variable &OWNER.

Example 2: Retrieve Private User Authorities for a Document

```
RTVDLOAUT DLO(MYDOC) FLR(MYFLR)
           USRAUT(&RTNUSRAUT) STRUSRAUTE(1)
```

This command retrieves the user authorities for document MYDOC in folder MYFLR and stores the list in the &RTNUSRAUT variable. The list will contain up to 50 privately authorized users and their authorities.

[Top](#)

Error messages

*ESCAPE Messages

CPF8AC0

&1 command failed.

[Top](#)

Retrieve DLO Name (RTVDLONAM)

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

Parameters
 Examples
 Error messages

The Retrieve Document Library Object Name (RTVDLONAM) command is used to retrieve an alternate name for a filed document, folder, or distribution document and place the value into the specified variables. This command is valid only within a CL program or REXX exec.

Restrictions:

- A user must have use (*USE) authority to the filed document or folder to retrieve the various forms of the name.
- A user must have all object (*ALLOBJ) special authority to retrieve the various forms of the name for a distribution document.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Character value, *DOCID, *LADNTSP, *SYSOBJNAM	Required, Positional 1
FLR	Folder	Character value, *NONE	Optional, Positional 2
DOCID	Document identifier	Character value, *NONE	Optional
LADNTSP	LADN timestamp	0000000000000001-FFFFFFFFFFFFFF, *NONE	Optional
SYSOBJNAM	System object name	Name, *NONE	Optional
OBJCLS	Object class	*DOC, *FLR, *DST	Optional
RTNDLO	CL var for RTNDLO (12)	Character value	Optional
RTNFLR	CL var for RTNFLR (63)	Character value	Optional
RTNDOCID	CL var for RTNDOCID (24)	Character value	Optional
RTNLADNTSP	CL var for RTNLADNTSP (16)	Character value	Optional
RTNOBJNAM	CL var for RTNOBJNAM (10)	Character value	Optional
RTNOBJCLS	CL var for RTNOBJCLS (8)	Character value	Optional
RTNASP	CL var for RTNASP (2 0)	Decimal number	Optional
RTNOVRFLW	CL var for RTNOVRFLW (1)	Character value	Optional

Top

Document library object (DLO)

Specifies the document or folder for which a name is to be retrieved.

This is a required parameter.

***DOCID**

The document or folder is identified using its library-assigned document name specified in the DOCID parameter. The **Folder (FLR)** parameter must be *NONE.

***LADNTSP**

The document or folder is identified using its library-assigned document name (LADN) timestamp specified in the LADNTSP parameter. The FLR parameter must be *NONE.

***SYSOBJNAM**

The document or folder is identified using its system object name specified in the SYSOBJNAM parameter. The FLR parameter must be *NONE.

name Specify the user-assigned name of the document or folder. The folder containing the specified document or folder is specified in the FLR parameter.

This is a required parameter.

Top

Folder (FLR)

Specifies the folder path that contains the document or folder specified in the **Document library object (DLO)** parameter. The FLR parameter must be *NONE if the DLO parameter is *DOCID, *LADNTSP, or *SYSOBJNAM.

***NONE**

The document or folder is not contained in a folder, or is identified using the **DOCID**, **LADNTSP**, or **SYSOBJNAM** parameter.

name Specify the name of the folder that contains the document or folder named in the DLO parameter.

Top

Document identifier (DOCID)

Specifies the library-assigned name of the document. This is the name assigned to the document by the system when it was created. Documents filed outside the local system have only library-assigned document names. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format:

YYYYMMDDHHMNSSHSSNSNSNSN

where:

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of a second
SNSNSNSN = system name

***NONE**

A library-assigned document name is not specified for the object.

name Specify the library-assigned name of the document or folder object. The library-assigned document name can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) or other command.

Top

LADN timestamp (LADNTSP)

Specifies the library assigned document name (LADN) timestamp of the document or folder object. A LADN timestamp is specified only if *LADNTSP is specified on the **Document library object (DLO)** parameter.

***NONE**

A LADN timestamp is not specified for the object.

timestamp

Specify the LADN timestamp of the document library object.

Top

System object name (SYSOBJNAM)

Specifies the system object name of the document or folder object. A system object name is specified only if *SYSOBJNAM is specified on the **Document library object (DLO)** parameter.

***NONE**

A system object name is not specified for the object.

name Specify the system object name of the document library object.

Top

Object class (OBJCLS)

Specifies the class of the object to locate.

***DOC** The specified DLO is a filed document.

***FLR** The specified DLO is a folder.

***DST** The specified DLO is a distribution document.

Top

CL var for RTNDLO (12) (RTNDLO)

Specifies the name of a 12-character CL variable used to retrieve the user-assigned name of the selected document or folder. The *NONE value is returned for a distribution document or a document without a folder.

Top

CL var for RTNFLR (63) (RTNFLR)

Specifies the name of a 63-character CL variable used to retrieve the folder path of the selected document or folder. The *NONE value is returned for a distribution document, a document without a folder, or a first-level folder.

Top

CL var for RTNDOCID (24) (RTNDOCID)

Specifies the name of a 24-character CL variable used to retrieve the library-assigned document name of the selected object. Refer to the **Document identifier (DOCID)** parameter for the format of the library-assigned document name.

Top

CL var for RTNLADNTSP (16) (RTNLADNTSP)

Specifies the name of a 16-character CL variable used to retrieve the timestamp from the LADN of the selected object. The variable is in the form YYYYMMDDHHMNSSNN.

Top

CL var for RTNOBJNAM (10) (RTNOBJNAM)

Specifies the name of a 10-character CL variable used to retrieve the system object name of the selected object.

Top

CL var for RTNOBJCLS (8) (RTNOBJCLS)

Specifies the name of a 8-character CL variable used to retrieve the object class. A value of *DOC is returned for a filed document, *FLR for a folder, and *DST for a distribution document.

Top

CL var for RTNASP (2 0) (RTNASP)

Specifies the name of a variable used to return the auxiliary storage pool ID. In control language (CL) programs, this should be a decimal variable of length (2 0). The following values can be returned:

- 1 The object is in the system auxiliary storage pool.
- 2-32 The object is in a user auxiliary storage pool.

Top

CL var for RTNOVRFLW (1) (RTNOVRFLW)

Specifies the name of a 1-character CL variable used to retrieve the overflow status of the object, where:

- N = No, the object has not overflowed its ASP
- Y = Yes, the object has overflowed its ASP and part or all of the object resides in the system ASP

Examples

```
RTVDLONAM DLO(MYDOC) FLR(MYFLR) OBJCLS(*DOC) +  
          RTNDOCID(&DOCID)
```

This command finds the document MYDOC in folder MYFLR and returns its document identifier in the variable &DOCID.

Error messages

*ESCAPE Messages

CPF8AC0

&1 command failed.

CPF8AC1

Not authorized to distribution documents.

CPF8A75

Not authorized to access folder &1.

CPF8A77

Folder &1 not found.

CPF8A82

Document &2 not found in folder &1.

CPF8A83

Not authorized to access document &2 in folder &1.

Retrieve Document (RTVDOC)

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

Parameters
 Examples
 Error messages

The Retrieve Document (RTVDOC) command allows you to retrieve information from a specific document.

Restrictions:

- To retrieve any records from the document to a database file, you must have use (*USE) authority to the document or be working on behalf of a user that has *USE authority to the document.
- To check out the document, you must have at least change (*CHANGE) authority to the document, or be working on behalf of a user that has *CHANGE authority to the document.
- To work on behalf of another user, you must have either all object (*ALLOBJ) special authority or special permission (granted with the Grant User Permission (GRTUSRPMN) command).

Top

Parameters

Keyword	Description	Choices	Notes
FROMDOC	From document	Character value, *DOCID	Required, Positional 1
FROMFLR	From folder	Character value, *NONE	Optional, Positional 2
OUTFILE	File to receive output	Single values: *NONE Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , *FIRST	
	Element 2: Replace or add records	*REPLACE, *ADD	
OUTDTATYP	Type of data for output	Single values: *DFT, *ALL Other values (up to 21 repetitions): *DOCD, *DOCCLS, *SUBJECT, *AUTHOR, *FILCAB, *CPYLS, *KWD, *DOCDATE, *FILDATE, *CRTDATE, *CHGDATE, *EXPDATE, *ACTDATE, *CMPDATE, *REF, *STATUS, *PROJECT, *IDP, *DOC	Optional
USRID	User identifier	Single values: *CURRENT Other values: <i>Element list</i>	Optional
	Element 1: User ID	<i>Character value</i>	
	Element 2: Address	<i>Character value</i>	
DOCID	Document identifier	Character value, *NONE	Optional
CHKOUT	Check out	*NO, *YES	Optional

Keyword	Description	Choices	Notes
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEV Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	

Top

From document (FROMDOC)

Specifies the name of the document being retrieved.

name Specify the user-assigned name of the document being retrieved.

*DOCID

The document being retrieved is identified by the library-assigned document name that must be specified on the **Document identifier (DOCID)** parameter.

Top

From folder (FROMFLR)

Specifies the name of the folder that contains the document being retrieved. This is the name assigned to the folder when it is created. This parameter must be specified if a document name is specified.

*NONE

No folder name is specified when the document is identified by the library-assigned document name.

name Specify the name of the folder that contains the retrieved document.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

*CURLIB

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses QAOSIRTV in QSYS with a format name of OSRTVD as a model.

This file can be reused when other RTVDOC commands are used. Output can be added to the file or can replace the existing records. The IBM-supplied database file QAOSIRTV in library QSYS cannot be specified.

Top

Output member options (OUTMBR)

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

Element 1: Member to receive output

*FIRST

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

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

Element 2: Replace or add records

*REPLACE

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

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

Top

Type of data for output (OUTDTATYP)

Specifies the parts of information about the document that are written to the database file.

***DFT** The document information record is written. This is the same as specifying *DOCD and *DOC.

- Record code - 105, Document description
- Record code - 800, Document data

***ALL** All information records are written.

***ACTDATE**

The action due date record is written. The record code is 135.

***AUTHOR**

The author records are written. The record code is 145.

***CHGDATE**

The date last changed record is written. The record code is 130.

***CMPDATE**

The completion date record is written. The record code is 140.

***CPYLST**

The copy list records are written. The record code is 150.

- *CRTDATE**
The create date record is written. The record code is 110.
- *DOCCLS**
The document class record is written. The record code is 155.
- *DOCD**
The document description record is written. The record code is 105.
- *DOCDATE**
The document date record is written. The record code is 120.
- *EXPDATE**
The expiration date record is written. The record code is 115.
- *FILCAB**
The file cabinet reference record is written. The record code is 160.
- *FILDATE**
The file date record is written. The record code is 125.
- *IDP** The interchange document profile (IDP) is written. The record code is 500.
- *IDXDATE**
The last indexed date record is written to the output file.
- *KWD**
The keyword records are written. The record code is 170.
- *PROJECT**
The project record is written. The record code is 185.
- *REF** The reference record is written. The record code is 175.
- *REVDATE**
The date of the last revision to the document content is written to the output file.
- *STATUS**
The status record is written. The record code is 180.
- *SUBJECT**
The subject records are written. The record code is 165.
- *USEDATE**
The date last used record is written. The record code is 200.
- *DOC** The document data record is written. The record code is 800.

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 identifier (DOCID)

Specifies the library-assigned name of the document. This is the name assigned to the document by the system when it was created. Documents filed outside the local system have only library-assigned document names. The library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format:

YYYYMMDDHHMNSSHSSNSNSNSN

where:

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of a second
SNSNSNSN = system name

*NONE

No library-assigned document name is required when the document is identified on the **Document (DOC)** parameter.

name Specify the library-assigned name of the document being sent.

Top

Check out (CHKOUT)

Specifies that the document being retrieved can be replaced with new or changed data. If the document is read only, then specify *NO. If the document being retrieved cannot be replaced, and *YES is specified, this will cause an error to occur.

*NO The retrieve request only reads the data. Users requesting this function need only read (*READ) authority to the document. Public authority is *READ authority.

*YES The document data can be updated and replaced later. Users requesting this function must have change (*CHANGE) authority. The document will be unavailable for other users to update until the replacement of this document is done. The replacement can be done by using the Replace Document (RPLDOC) command.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values. The character identifier is related to the display device used to enter the command.

The value specified on the **User identifier (USRID)** parameter is translated to character set and code page '930 500'.

Single values

*SYSVAL

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

*DEV D

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Examples

Example 1: Copying All Information

```
RTVDOC FROMDOC(MYDOC) FROMFLR(PERSONAL) USRID(*CURRENT)
        OUTFILE(*CURLIB/MYFILE) OUTMBR(*FIRST) MBROPT(*ADD)
        OUTDTATYP(*ALL)
```

This command copies all information about document MYDOC located in folder PERSONAL for the current user of this command. CHECKOUT(*NO) is assumed; therefore, the document data can only be read. The output is directed to the database file MYFILE in the user's current library and is added to the first member in that file.

Example 2: Copying Default Information

```
RTVDOC FROMDOC(SECOP) FROMFLR(PERSONAL) USRID(MARY SYSTEM1 )
        CHKOUT(*YES)
        OUTFILE(MARLIB/SECFILE) OUTMBR(*FIRST *ADD)
```

This command copies the default information (*DOCD and *DOC) about document SECOP located in folder PERSONAL for MARY. The document can be updated with new data and then replaced. The current user of this command must have the authority to work on behalf of MARY given by Mary by using the GRTUSRPMN command. The output is directed to the database file SECFILE in Mary's library MARLIB. The output is added to the first member of SECFILE.

Error messages

*ESCAPE Messages

CPF900B

User ID and address &1 &2 not in System Distribution Directory.

CPF900C

Sign on and verify of user failed.

CPF905C

Error occurred trying to find a translation table.

CPF905F

Retrieval of document from library failed.

CPF9096

Cannot use CMDCHRID(*DEVVD), DOCCHRID(*DEVVD) in batch job.

CPF9860

Error occurred during output file processing.

Retrieve Disk Information (RTVDSKINF)

Where allowed to run:

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

Parameters
Examples
Error messages

Threadsafe: No

The Retrieve Disk Information (RTVDSKINF) command is used to collect disk space information. The collected information is stored in a database file in library QUSRSYS. The file name depends on the auxiliary storage pool (ASP) device for which disk space information is retrieved. If the information was retrieved from the system and basic ASPs, the collected information will be stored in file QAEZDISK. If the information was retrieved from an independent ASP device, the collected information will be stored in file QAEZnnnnn, where 'nnnnn' is the ASP number of the independent ASP. The information will be stored in a data base file member named QCURRENT.

Each time this command is run, existing information in QCURRENT is written over. To save existing information in member QCURRENT, rename file QAEZDISK or QAEZDnnnnn, or copy the member to another file.

To get the most accurate results, this command should be run at a time when there is very little system activity.

RTVDSKINF will also check that certain objects are addressable through a library. If the object is not attached to a library, it will be moved to the proper library or the QRCL library (for objects in the system or basic auxiliary storage pools) or library QRCLxxxxx (for objects in an independent auxiliary storage pool where 'xxxxx' is the number associated with the independent ASP). Objects found to be damaged or unusable will be deleted. A message will be sent for each object deleted by RTVDSKINF.

Note: Do not rename member QCURRENT within file QAEZDISK or QAEZDnnnnn. If there is more than one member in QAEZDISK or QAEZDnnnnn, the results of running this command can be unpredictable.

Top

Parameters

Keyword	Description	Choices	Notes
ASPDEV	ASP device	Name, <u>*SYSBAS</u>	Optional, Positional 1

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device for which disk space information is to be retrieved.

*SYSBAS

Disk information for the system ASP and all basic ASPs is retrieved. The system ASP has an ASP number of 1. Basic ASPs have ASP numbers from 2 through 32.

name Specify the name of the ASP device for which disk space information is to be retrieved. Independent ASP devices have ASP numbers greater than 32. You can submit multiple jobs, each performing RTVDSKINF on a different ASP device, to retrieve disk space information for multiple ASP devices in parallel.

Top

Examples

```
RTVDSKINF ASPDEV(*SYSBAS)
```

This command retrieves disk space information for the system and basic auxiliary storage pools (ASPs), and stores it in member QCURRENT of database file QAEZDISK. Any information in member QCURRENT is overwritten.

Some objects will be checked to verify that they are addressable through a library and if found floating, they will be moved into a library. Damaged or unusable objects will be deleted from the system.

Top

Error messages

*ESCAPE Messages

CPF1ED1

Not authorized to collect disk space information.

CPF1ED2

File &1 is in use and cannot be accessed.

CPF1E99

Unexpected error occurred.

Top

Retrieve Data Area (RTVDTAARA)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Retrieve Data Area (RTVDTAARA) command is used in a CL program or REXX procedure to retrieve all or part of a specified data area and copy it into a variable. RTVDTAARA does not retrieve any other attributes of the data area. Existence of the data area is not required at the time the CL program is compiled.

If the job is a group job, the data area specified may be the group data area (*GDA). This data area is automatically associated with the group, and it is inaccessible from jobs outside the group. The length of this character data area is 512 bytes. More information about group jobs is in the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

A local data area (*LDA) is a character data area that is 1024 bytes in length, and it is automatically associated with the job. Another job cannot access the local data area.

If the job is a prestart job, the data area specified may be the data area that contains program initialization parameter data (*PDA). This data area is automatically associated with the prestart job and is inaccessible from other jobs. The length of this character data area is 2000 bytes. More information about prestart jobs is in the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

When a data area other than a local data area, group data area, or program initialization parameter data area must be retrieved during the processing of the RTVDTAARA command, the data area is locked during the retrieval operation so that commands in other jobs cannot change or destroy it until the operation is complete. If the data area is shared with other jobs and is updated in steps involving more than one command in a job, the data area should be explicitly allocated to that job until all the steps have been performed. A data area other than a local data area, group data area, or program initialization parameter data area can be explicitly allocated with the Allocate Object (ALCOBJ) command. No allocation is necessary for a local data area, group data area, or program initialization parameter data area.

Restrictions

1. To use this command, the user must have *USE authority for the data area and *EXECUTE authority for the library where the data area is located. No specific authority is required to retrieve the value of a local data area or group data area.
2. This command is conditionally threadsafe. The following restrictions apply:
 - a. Retrieving DDM data areas in a job that allows multiple threads is not threadsafe.
 - b. Retrieving DDM data areas will fail when more than one thread is active in a job.

Top

Parameters

Keyword	Description	Choices	Notes
DTAARA	Data area specification	<i>Element list</i>	Required, Positional 1
	Element 1: Data area	Single values: *LDA, *GDA, *PDA Other values: <i>Qualified object name</i>	
	Qualifier 1: Data area	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Substring specifications	Single values: *ALL Other values: <i>Element list</i>	
	Element 1: Substring starting position	1-2000	
	Element 2: Substring length	1-2000	
RTNVAR	CL variable for returned value	<i>Not restricted</i>	Required, Positional 2

Top

Data area specification (DTAARA)

Specifies the name of the data area whose value is retrieved.

This is a required parameter.

Element 1: Data area

Single values

- *LDA The value of the local data area is being retrieved.
- *GDA The value of the group data area is being retrieved. This value is valid only if this job is a group job.
- *PDA The value of the program initialization parameter data area is being retrieved. This value is valid only if this is a prestart job.

Qualifier 1: Data area

name Specify the name of the data area.

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 data area is located.

Element 2: Substring specifications

Single values

***ALL** The entire data area is retrieved.

Element 1: Substring starting position

1-2000 Specify the starting position of the data area being retrieved.

Element 2: Substring length

1-2000 Specify the length of the data area substring being retrieved. It is not possible to retrieve data outside the data area. The combination of starting position and length must always specify positions within the data area.

Top

CL variable for returned value (RTNVAR)

Specifies the name of the variable that receives the contents of the data area.

No type conversion is performed by the RTVDTAARA command:

- If RTNVAR is declared as TYPE(*DEC), the data area retrieved must be TYPE(*DEC).
- If RTNVAR is declared as TYPE(*CHAR), the data area retrieved must be either TYPE(*CHAR) or TYPE(*LGL).
- If RTNVAR is declared as TYPE(*LGL), the data area retrieved must be either TYPE(*LGL) or TYPE(*CHAR) with a value of either '0' or '1'.

If a retrieved character string is shorter than the length of the variable specified by the RTNVAR parameter, the value is padded on the right with blanks. The retrieved string length must be less than or equal to the variable length.

When decimal data areas are retrieved, the decimals are aligned. The value of the integer portion of the data area must fit into the integer portions of the variable. Fractional data is truncated if the fraction contains more digits than the variable.

This is a required parameter.

unrestricted-value

Specify the program variable that will receive the contents of the data area.

Top

Examples

Assume data area DA1 has been created by the following command:

```
CRTDTAARA DTAARA(DA1) TYPE(*CHAR) LEN(3) VALUE(ABC)
```

and variable &CLVAR1 has been declared as:

```
DCL VAR(&CLVAR1) TYPE(*CHAR) LEN(5) VALUE(VWXYZ)
```

Example 1: Retrieving a Character Data Area

```
RTVDTAARA DTAARA(DA1) RTNVAR(&CLVAR1)
```

results in:

```
&CLVAR1 = 'ABC '
```

Example 2: Retrieving a Subset of a Character Data Area

```
RTVDTAARA DTAARA(DA1 (2 1)) RTNVAR(&CLVAR1)
```

results in:

```
&CLVAR1 = 'B '
```

Example 3: Retrieving a Decimal Data Area

Assume data area DA2 has been created with the following attributes:

```
CRTDTAARA DTAARA(DA2) TYPE(*DEC) LEN(5 2) VALUE(12.39)
```

and variable &CLVAR2 has been declared as:

```
DCL VAR(&CLVAR2) TYPE(*DEC) LEN(5 1) VALUE(4567.8)
```

Running this command:

```
RTVDTAARA DTAARA(DA2) RTNVAR(&CLVAR2)
```

results in:

```
&CLVAR2 = 0012.3
```

Note: Fractional digits are truncated instead of rounded.

[Top](#)

Error messages

*ESCAPE Messages

CPF0811

RTNVAR parameter has incorrect length for data area.

CPF0812

RTNVAR parameter type not valid for data area &1.

CPF0813
Value in data area &1 not logical value.

CPF101A
Operation on DDM data area &1 in &2 failed.

CPF1015
Data area &1 in &2 not found.

CPF1016
No authority to data area &1 in &2.

CPF1021
Library &1 not found for data area &2.

CPF1022
No authority to library &1 data area &2.

CPF1043
Boundary alignment for data area not valid.

CPF1044
AREA parameter not valid for data area.

CPF1045
CPYPTR parameter not valid for data area.

CPF1046
DTAARA(*GDA) not valid because job not group job.

CPF1063
Cannot allocate data area &1 in library &2.

CPF1067
Cannot allocate library &1.

CPF1072
DTAARA(*PDA) not valid because job not prestart job.

CPF1087
Substring not allowed for decimal or logical data area.

CPF1088
Starting position outside of data area.

CPF1089
Substring specified for data area not valid.

CPF180B
Function &1 not allowed.

CPF9899
Error occurred during processing of command.

Top

Retrieve Group Attributes (RTVGRPA)

Where allowed to run:

- Interactive program (*IPGM)
- Interactive REXX procedure (*IREXX)

Threadsafe: No

Parameters
Examples
Error messages

The Retrieve Group Attributes (RTVGRPA) command retrieves information about the group in which the job that issued the command belongs. The following attributes can be retrieved:

- The group job name of the job calling the RTVGRPA command
- A list containing information about all active jobs in the group
- A count of the number of active jobs in the group
- The name of the group message queue
- The library in which the group message queue resides
- The group job name and job number of the previously active job in the group
- A control code indicating why the currently active job in the group gained control

Top

Parameters

Keyword	Description	Choices	Notes
GRPJOB	CL var for GRPJOB (10)	Character value	Optional, Positional 1
GRPJOB L	CL var for GRPJOB list (1056)	Character value	Optional, Positional 2
GRPJOB CNT	CL var for GRPJOB CNT (3 0)	Decimal number	Optional, Positional 3
MSGQ	CL var for MSGQ (10)	Character value	Optional
MSGQLIB	CL var for MSGQLIB (10)	Character value	Optional
PRVGRPJOB	CL var for PRVGRPJOB (16)	Character value	Optional
CTLCDE	CL var for CTLCDE (3 0)	Decimal number	Optional

Top

CL var for GRPJOB (10) (GRPJOB)

Specifies the name of the CL variable that receives the group job name of the job. The variable must be a character variable with a minimum length of 10 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks.

Top

CL var for GRPJOB list (1056) (GRPJOB L)

Specifies the name of the CL variable that receives the list of jobs in the group. Each entry contains the job's group job name (10), job number (6), and the 50 characters of descriptive text. The maximum number of entries in the variable is 16. The entries are ordered by most recently active job. The variable must be a character variable with a minimum length of 1056 characters. If the group job list has fewer characters than the variable allows, the value is padded on the right with blanks.

Top

CL var for GRPJOB CNT (3 0) (GRPJOB CNT)

Specifies the CL variable that receives the count of active jobs in the group. The CL variable must be a three-position decimal variable with no decimal positions. The CL variable contains the number of non-blank entries in the group job list. The count includes all of the active jobs in the group. Jobs that have not completely ended (jobs that have been canceled) are not counted.

Top

CL var for MSGQ (10) (MSGQ)

Specifies the name of the CL variable that receives the group message queue name. This variable must be a character variable with a minimum length of 10 characters. If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no message queue associated with the group, the CL variable is set to the special value of *NONE.

Top

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the group message queue. This variable must be a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no message queue associated with the group, the CL variable is set to blanks.

Top

CL var for PRVGRPJOB (16) (PRVGRPJOB)

Specifies the name of the CL variable that receives the group job name and job number of the previously active job in the group. The variable must be a character variable with a minimum length of 16 characters. If the group job name has fewer characters than the variable allows, the value is padded on the right with blanks. If there is no previously active job in the group, the group job name portion of the CL variable is set to the special value of *NONE, and the job number portion of the CL variable is set to blanks. The CL variable is returned in the following format:

```
Group-job-name CHAR(10)
Job-number     CHAR(6)
```

Top

CL var for CTLCDE (3 0) (CTLCDE)

Specifies the name of the CL variable that receives information about why the active job in the group has gained control. The CL variable must be a three-position decimal variable with no decimal positions. The following control codes (and their meanings) are possible:

- 0 There was no previously active job (no Transfer to Group Job (TFRGRPJOB) commands have been run for this group).
- 10 The previously active job selected this job to be transferred to on the TFRGRPJOB command.
- 20 The previously active job's first group program ended normally, and this job was the most recently active job in the group.
- 30 The previously active job was ended by the End Group Job (ENDGRPJOB) command, and this job was selected to gain control (the **Group job to be resumed (RSMGRPJOB)** parameter specified this group job).
- 40 The previously active job was ended by the ENDGRPJOB command and selected a job other than this job to gain control (which was ended before it could be resumed). Since this job was the most recently active job in the group, control is passed to it.
- 50 The previously active job was ended by the ENDGRPJOB command, and this job was the most recently active job in the group (the RSMGRPJOB parameter specified *PRV).
- 60 The previously active job's first group program ended abnormally, and this job was the most recently active job in the group.
- 70 The previously active job was ended by the End Job (ENDJOB) command, and this job was the most recently active job in the group.

Top

Examples

Assume jobs 030001/QUSER/WORKST01 and 030002/QUSER/WORKST01 are group jobs with group job names GROUPJ1 and GROUPJ2, respectively. Also assume that message queue QGPL/GROUPMSGQ is associated with the group. If group job GROUPJ1 has just issued the TFRGRPJOB command to transfer to group job GROUPJ2, and GROUPJ2 called the following CL program:

PGM Example

```
DCL VAR(&GRPJOB) TYPE(*CHAR) LEN(10)
DCL VAR(&GRPJOB) TYPE(*CHAR) LEN(1056)
DCL VAR(&GRPCOUNT) TYPE(*DEC) LEN(3 0)
DCL VAR(&MSGQNAME) TYPE(*CHAR) LEN(10)
DCL VAR(&MSGQLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&PRVJOB) TYPE(*CHAR) LEN(16)
DCL VAR(&CTLCODE) TYPE(*DEC) LEN(3 0)
RTVGRPA GRPJOB(&GRPJOB) GRPJOB(&GRPJOB) +
        GRPJOBcnt(&GRPCOUNT) MSGQ(&MSGQNAME) +
        MSGQLIB(&MSGQLIB) PRVGRPJOB(&PRVJOB) +
        CTLCDE(&CTLCODE)
```

The contents of the CL variables returned are as follows:

```
&GRPJOB: GROUPJ2
&GRPJOB: GROUPJ2 030002 50 characters of text for
                    this group job...
                    GROUPJ1 030001 50 characters of text for
                    this group job...
```

Fourteen more entries, full of blanks

```
&GRPCOUNT:    002
&MSGQNAME:    GROUPMSGQ
&MSGQLIB:     QGPL
&PRVJOB:      GROUPJ1  030001
&CTLCODE:     010
```

Top

Error messages

*ESCAPE Messages

CPF1309

Subsystem cannot complete the &1 command.

CPF1311

Job is not a group job.

CPF1317

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

CPF1351

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

Top

Retrieve Image Catalog (RTVIMGCLG)

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

Parameters
Examples
Error messages

The Retrieve Image Catalog (RTVIMGCLG) command is used in a CL procedure to retrieve the name of the image catalog loaded in a virtual device, or to retrieve the name of the virtual device an image catalog is loaded in.

Restrictions:

- The following authorities are required to retrieve the name of the virtual device an image catalog is loaded in:
 1. Execute (*EXECUTE) authority to library QUSRSYS.
 2. Use (*USE) authority to the image catalog.
- Use (*USE) authority to the device is required to retrieve the name of the image catalog loaded in a virtual device.
- This command is valid only in a compiled CL procedure.

Top

Parameters

Keyword	Description	Choices	Notes
IMGCLG	Image catalog	<i>Name</i>	Optional
DEV	Virtual device	<i>Name</i>	Optional
DEVLOD	CL var for DEVLOD (10)	<i>Character value</i>	Optional
IMGCLGLOD	CL var for IMGCLGLOD (10)	<i>Character value</i>	Optional

Top

Image catalog (IMGCLG)

Specifies the image catalog whose associated virtual device name is to be retrieved.

Note: A value must be specified for this parameter or the **Virtual device (DEV)** parameter.

name Specify the name of the image catalog.

Top

Virtual device (DEV)

Specify the virtual device whose associated image catalog name is to be retrieved.

Note: A value must be specified for this parameter or the **Virtual device (IMGCLG)** parameter.

name Specify the name of the virtual device.

CL var for DEVL0D (10) (DEVL0D)

Specifies the name of the CL variable that receives the name of the current virtual device for which the specified image catalog is loaded. You must specify a CL variable declared as TYPE(*CHAR) with a minimum length of 10 characters. A value of *NONE will be returned if the name specified for the **Image catalog (IMGCLG)** parameter is not associated with a virtual device.

Top

CL var for IMGCLGLOD (10) (IMGCLGLOD)

Specifies the name of the CL variable that receives the name of the image catalog currently loaded for the specified virtual device. You must specify a CL variable declared as TYPE(*CHAR) with a minimum length of 10 characters. A value of *NONE will be returned if the name specified for the **Virtual device (DEV)** parameter is not associated with an image catalog.

Top

Examples

Example 1: Retrieving the Loaded Virtual Device Name

```
DCL VAR(&MYLODDEV) TYPE(*CHAR) LEN(10)
RTVIMGCLG IMGCLG(MYCLG) DEVL0D(&MYLODDEV)
```

This command will retrieve the name of the loaded virtual device associated with image catalog **MYCLG**.

Example 2: Retrieving the Image Catalog Loaded in a Virtual Device

```
DCL VAR(&MYCLGLOD) TYPE(*CHAR) LEN(10)
RTVIMGCLG DEV(OPTVRT01) IMGCLGLOD(&MYCLGLOD)
```

This command will retrieve the name of the image catalog associated with virtual device **OPTVRT01**.

Top

Error messages

*ESCAPE Messages

CPFBC45

Image catalog &1 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9820

Not authorized to use library &1.

CPF9825

Not authorized to device &1.

Retrieve Job Attributes (RTVJOBA)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Conditional

Parameters
Examples
Error messages

The Retrieve Job Attributes (RTVJOBA) command is used in a CL program or REXX procedure to retrieve the values of one or more job attributes and place the values into the specified variable. The attributes are retrieved for the job in which this command is used.

The CL prompt for this command lists the minimum length for retrieved variables next to the appropriate parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. This command is valid only within a CL program or a REXX procedure.
2. This command is conditionally threadsafe. Refer to **Considerations for Attribute Scope and Thread Safety** for additional information.

Considerations for Attribute Scope and Thread Safety:

This command is intended to be used to retrieve job attributes for the job in which the command is used. However, if a job attribute does not exist because the attribute is now supported at the thread level only, then the thread attribute is retrieved for the thread in which this command is used.

The Scope column shows whether the attribute is scoped to the Job or to the Thread.

The Threadsafe column indicates if the attributes are considered to be threadsafe.

Yes: - Attributes marked with this value can be retrieved safely from either the initial thread or from a secondary thread.

No: - Attributes marked with this value are not threadsafe, and should not be retrieved in a multi-threaded job.

Attribute Scope and Thread Safety Table:

Attribute	Scope	Threadsafe
-----	-----	-----
ASP group name (ASPGRP)	Thread	Yes
Break message handling (BRKMSG)	Job	Yes
Character identifier control (CHRIDCTL)	Job	Yes
Coded character set ID (CCSID)	Job	Yes
Country or region ID (CNTRYID)	Job	Yes
Current library (CURLIB)	Thread	Yes
Current user profile (CURUSER)	Thread	Yes
Date format (DATFMT)	Job	Yes
Date separator (DATSEP)	Job	Yes
Day of week (DAYOFWEEK)	Job	Yes
DDM conversation handling (DDMCNV)	Job	Yes
Decimal format (DECfmt)	Job	Yes
Default coded character set identifier (DFTCCSID)	Job	Yes
Default wait (DFTWAIT)	Job	Yes
Device recovery action (DEVRCYACN)	Job	Yes
End status (ENDSTS)	Job	Yes
Inquiry message reply (INQMSGRPY)	Job	Yes
Job accounting code (ACGCDE)	Job	Yes
Job date with century (CYMDDATE)	Job	Yes
Job date without century (DATE)	Job	Yes
Job local date and time (DATETIME)	Job	Yes
Job log output (LOGOUTPUT)	Job	Yes
Job message queue full action (JOBMSGQFL)	Job	Yes
Job message queue maximum size (JOBMSGQMX)	Job	Yes
Job name (JOB)	Job	Yes
Job number (NBR)	Job	Yes
Job subtype (SUBTYPE)	Job	Yes
Job switches (SWS)	Job	Yes
Job type (TYPE)	Job	Yes
Language ID (LANGID)	Job	Yes
Logging level (LOGLVL)	Job	Yes
Logging of CL programs (LOGCLPGM)	Job	Yes
Logging severity (LOGSEV)	Job	Yes
Logging text (LOGTYPE)	Job	Yes
Output queue library name (OUTQLIB)	Job	Yes
Output queue name (OUTQ)	Job	Yes
Print key format (PRTKEYFMT)	Job	Yes
Print text (PRTTXT)	Job	Yes
Printer device name (PRTDEV)	Job	Yes
Program return code (RTNCDE)	Job	No
Purge (PURGE)	Job	Yes
Resource affinity group (RSCAFNGRP)	Job	Yes
Run priority (RUNPTY)	Job	Yes
Sort sequence (SRTSEQ)	Job	Yes
Sort sequence library (SRTSEQLIB)	Job	Yes
Spooled file action (SPLFACN)	Job	Yes
Status message handling (STSMMSG)	Job	Yes
Submitter's message queue library name (SBMMSGQLIB)	Job	Yes
Submitter's message queue name (SBMMSGQ)	Job	Yes
System library list (SYSLIBL)	Thread	Yes
Thread resource affinity (THDRSCAFN)	Thread	Yes
Time separator (TIMSEP)	Job	Yes
Time slice (TIMESLICE)	Job	Yes
Time-slice end pool (TSEPOOL)	Job	Yes
Time zone abbreviated name (TIMZONABBR)	Job	Yes
Time zone description (TIMZON)	Job	Yes
Time zone full name (TIMZONFULL)	Job	Yes
Time zone offset (TIMOFFSET)	Job	Yes
User library list (USRLIBL)	Thread	Yes
User name (USER)	Job	Yes
Year offset (YEAROFS)	Job	Yes

Top

Parameters

Keyword	Description	Choices	Notes
JOB	CL var for JOB (10)	Character value	Optional, Positional 1
USER	CL var for USER (10)	Character value	Optional, Positional 2
NBR	CL var for NBR (6)	Character value	Optional, Positional 3
CURUSER	CL var for CURUSER (10)	Character value	Optional
TYPE	CL var for TYPE (1)	Character value	Optional
SUBTYPE	CL var for SUBTYPE (1)	Character value	Optional
SYSLIBL	CL var for SYSLIBL (165)	Character value	Optional
CURLIB	CL var for CURLIB (10)	Character value	Optional
USRLIBL	CL var for USRLIBL (2750)	Character value	Optional
ASPGRP	CL var for ASPGRP (10)	Character value	Optional
LOGLVL	CL var for LOGLVL (1)	Character value	Optional
LOGSEV	CL var for LOGSEV (2 0)	Decimal number	Optional
LOGTYPE	CL var for LOGTYPE (10)	Character value	Optional
LOGCLPGM	CL var for LOGCLPGM (10)	Character value	Optional
LOGOUTPUT	CL var for LOGOUTPUT (10)	Character value	Optional
JOBMSGQMX	CL var for JOBMSGQMX (2 0)	Decimal number	Optional
JOBMSGQFL	CL var for JOBMSGQFL (10)	Character value	Optional
INQMSGRPY	CL var for INQMSGRPY (10)	Character value	Optional
STSMSG	CL var for STSMSG (7)	Character value	Optional
BRKMSG	CL var for BRKMSG (7)	Character value	Optional
DEVRCYACN	CL var for DEVRCYACN (13)	Character value	Optional
RTNCDE	CL var for RTNCDE (5 0)	Decimal number	Optional
ENDSTS	CL var for ENDSTS (1)	Character value	Optional
PRTDEV	CL var for PRTDEV (10)	Character value	Optional
OUTQ	CL var for OUTQ (10)	Character value	Optional
OUTQLIB	CL var for OUTQLIB (10)	Character value	Optional
SPLFACN	CL var for SPLFACN (10)	Character value	Optional
PRTTXT	CL var for PRTTXT (30)	Character value	Optional
PRTKEYFMT	CL var for PRTKEYFMT (10)	Character value	Optional
SRTSEQ	CL var for SRTSEQ (10)	Character value	Optional
SRTSEQLIB	CL var for SRTSEQLIB (10)	Character value	Optional
LANGID	CL var for LANGID (3)	Character value	Optional
CNTRYID	CL var for CNTRYID (2)	Character value	Optional
CCSID	CL var for CCSID (5 0)	Decimal number	Optional
DFTCCSID	CL var for DFTCCSID (5 0)	Decimal number	Optional
CHRIDCTL	CL var for CHRIDCTL (10)	Character value	Optional
DECFMT	CL var for DECFMT (1)	Character value	Optional
DATFMT	CL var for DATFMT (4)	Character value	Optional
DATSEP	CL var for DATSEP (1)	Character value	Optional
TIMSEP	CL var for TIMSEP (1)	Character value	Optional

Keyword	Description	Choices	Notes
DATE	CL var for DATE (6)	Character value	Optional
CYMDDATE	CL var for CYMDDATE (7)	Character value	Optional
DATETIME	CL var for DATETIME (20)	Character value	Optional
DAYOFWEEK	CL var for DAYOFWEEK (4)	Character value	Optional
TIMZON	CL var for TIMZON (10)	Character value	Optional
TIMZONABBR	CL var for TIMZONABBR (10)	Character value	Optional
TIMZONFULL	CL var for TIMZONFULL (50)	Character value	Optional
TIMOFFSET	CL var for TIMOFFSET (3 0)	Decimal number	Optional
YEAROFS	CL var for YEAROFS (3 0)	Decimal number	Optional
ACGCDE	CL var for ACGCDE (15)	Character value	Optional
SWS	CL var for SWS (8)	Character value	Optional
RUNPTY	CL var for RUNPTY (2 0)	Decimal number	Optional
TIMESLICE	CL var for TIMESLICE (7 0)	Decimal number	Optional
PURGE	CL var for PURGE (10)	Character value	Optional
DFTWAIT	CL var for DFTWAIT (7 0)	Decimal number	Optional
SBMSGQ	CL var for SBMSGQ (10)	Character value	Optional
SBMSGQLIB	CL var for SBMSGQLIB (10)	Character value	Optional
DDMCNV	CL var for DDMCNV (5)	Character value	Optional
TSEPOOL	CL var for TSEPOOL (10)	Character value	Optional
THDRSCAFN	CL var for THDRSCAFN (20)	Character value	Optional
RSCAFNGRP	CL var for RSCAFNGRP (10)	Character value	Optional

Top

CL var for JOB (10) (JOB)

Specifies the name of the CL variable that receives the name of the job. The variable must be a character variable with a minimum length of 10 characters.

Top

CL var for USER (10) (USER)

Specifies the name of the CL variable that receives the name of the user profile associated with the job when the job was started. The user name is the second part of the qualified job name. The variable must be a character variable with a minimum length of 10 characters.

Top

CL var for NBR (6) (NBR)

Specifies the name of the CL variable that receives the 6-character number assigned to the job by the system. The job number is the first part of the qualified job name (job-number/user-name/job-name).

Top

CL var for CURUSER (10) (CURUSER)

Specifies the name of the CL variable that receives the name of the current user profile. The variable must be a character variable with a minimum length of 10 characters.

Top

CL var for TYPE (1) (TYPE)

Specifies, the name of the CL variable that receives the 1-character value representing the environment of the job. A character value of 0 indicates that the job is running as a batch job, and a 1 indicates an interactive job. The variable must be a character variable with a minimum length of 1 character.

Top

CL var for SUBTYPE (1) (SUBTYPE)

Specifies the name of the CL variable that receives the subtype value for the environment of the job. The variable must be a character variable with a minimum length of 1 character. The following values can be returned:

Value Environment

- * The job has no subtype
- E The job is running as an evoked job
- T The job is running as a Multiple Requester Terminal (MRT) job
- J The job is running as a prestart job
- P The job is running as a print driver

Top

CL var for SYSLIBL (165) (SYSLIBL)

Specifies the name of the CL variable that receives the system portion of the thread's library list. Each library name returned is left-justified in an 11-character field and padded on the right with blanks. The variable must be a character variable with a minimum of 165 characters.

Top

CL var for CURLIB (10) (CURLIB)

Specifies the name of a CL variable that receives the name of the current library for the thread. The variable must be a character variable with a minimum length of ten characters.

Note: If the thread does not have a current library, a value of *NONE is returned in this variable.

Top

CL var for USRLIBL (2750) (USRLIBL)

Specifies the name of the CL variable that receives the user portion of the thread's library list. Each library name returned is left-justified in an 11-character field and padded on the right with blanks. If the CL variable is too small for the library list, an exception is signaled and no library names are returned. The variable must be a character variable with a minimum length of 275 characters. A character variable with a minimum length of 2750 characters is needed when the user portion of the library list contains 250 library names.

[Top](#)

CL var for ASPGRP (10) (ASPGRP)

Specifies the name of a CL variable that receives the auxiliary storage pool (ASP) group name. The variable must be a character variable with a minimum length of 10 characters. The special value *NONE is returned if there is no ASP group for the thread.

[Top](#)

CL var for LOGLVL (1) (LOGLVL)

Specifies the name of the CL variable that receives the 1-character value, ranging from 0 through 4 that is the message logging level being used to determine the type of message information logged in the job log. The variable must be a character variable with a minimum length of 1 character.

[Top](#)

CL var for LOGSEV (2 0) (LOGSEV)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 00 through 99, which is the severity level that is used in conjunction with the logging level to determine which error messages are logged in the job log. The variable must be a 2-digit decimal variable specified with no decimal positions.

[Top](#)

CL var for LOGTYPE (10) (LOGTYPE)

Specifies the name of the CL variable that receives the special value that indicates the level of text that appears for any message that is written to the job log. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for LOGCLPGM (10) (LOGCLPGM)

Specifies the name of the CL variable that receives the special value that indicates whether processed commands in a CL program are being logged in the job log. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for LOGOUTPUT (10) (LOGOUTPUT)

Specifies the name of the CL variable that receives the special value that indicates how the job's job log is to be produced when the job completes. The variable must be a character variable with a minimum length of 10 characters. The special value *JOBEND, *JOBLOGSVR, or *PND is returned.

[Top](#)

CL var for JOBMSGQMX (2 0) (JOBMSGQMX)

Specifies the name of a CL variable that receives the maximum size of the job message queue. The variable must be a 2-digit decimal variable with no decimal positions.

[Top](#)

CL var for JOBMSGQFL (10) (JOBMSGQFL)

Specifies the action that should be taken when the job message queue is full. The variable must have a minimum length of 10 characters. The special value *NOWRAP, *WRAP, or *PRTWRAP is returned.

[Top](#)

CL var for INQMSGRPY (10) (INQMSGRPY)

Specifies the name of the CL variable that receives the special value that indicates how inquiry messages are being handled by the job. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for STSMMSG (7) (STSMMSG)

Specifies the name of the CL variable that receives the special value indicating how status messages are handled for the job. The variable must be a character variable with a minimum length of 7 characters.

[Top](#)

CL var for BRKMSG (7) (BRKMSG)

Specifies the name of a CL variable that receives the special value that indicates the mode for break message handling that is in effect for the job. The variable must be a character variable with a minimum length of seven characters.

[Top](#)

CL var for DEVRCYACN (13) (DEVRCYACN)

Specifies the name of the CL variable that receives the special value indicating the recovery action to take for the job when an I/O error is encountered on the *REQUESTER device for interactive jobs. The variable must be a character variable with a minimum length of 13 characters.

[Top](#)

CL var for RTNCDE (5 0) (RTNCDE)

Specifies the name of the CL variable that receives the 5-digit decimal return code of an RPG, COBOL, DFU, or sort utility program. The return code is set by these programs before they return to the programs that call them. The return code indicates the completion status of the last program (of these types) that has completed processing within the job, as follows:

- 0 Normal return (RPG, COBOL, DFU, or Sort Utility)
- 1 LR (last record) indicator on (RPG)
- 2 Error - no halt indicator set (RPG, COBOL, DFU, or Sort Utility)
- 3 Halt indicator set on (one of the RPG indicators H1 through H9)

The CL variable must be a five-digit decimal variable with no decimal positions.

[Top](#)

CL var for ENDSTS (1) (ENDSTS)

Specifies the name of the CL variable that receives the cancellation status. The single-character value indicates whether a controlled cancellation that affects the job is currently being performed. A value of 1 indicates that either the system, the subsystem in which the job is running, or the job itself is being canceled; a 0 indicates no controlled cancellation is being performed. The CL variable must be a character variable with a minimum length of 1 character.

[Top](#)

CL var for PRTDEV (10) (PRTDEV)

Specifies the name of a CL variable that receives the name of the printer device. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for OUTQ (10) (OUTQ)

Specifies the name of the CL variable that receives the name of the output queue that is used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of the CL variable that receives the name of the library containing the output queue that is used by the job for spooled output. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for SPLFACN (10) (SPLFACN)

Specifies the name of a CL variable that receives the special value indicating whether spooled files are kept with the job or detached when the job ends. The variable must be a character variable with a minimum length of 10 characters. The special value *KEEP or *DETACH is returned.

[Top](#)

CL var for PRTTXT (30) (PRTTXT)

Specifies the name of the CL variable that receives the print text for the job. The variable must be a character variable with a minimum length of 30 characters.

[Top](#)

CL var for PRTKEYFMT (10) (PRTKEYFMT)

Specifies the name of the CL variable that receives the print key format for the job. The variable must have a minimum length of 10 characters. The special value *NONE, *PRTBDR, *PRTHDR, or *PRTALL is returned.

[Top](#)

CL var for SRTSEQ (10) (SRTSEQ)

Specifies the name of the CL variable that receives the name of the sort sequence table used for the job. The special value *LANGIDUNQ, *LANGIDSHR, or *HEX can be returned to the variable. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for SRTSEQLIB (10) (SRTSEQLIB)

Specifies the name of the CL variable that receives the name of the library containing the sort sequence table to be used for the job. The variable must be a character variable with a minimum length of 10 characters. If SRTSEQ is *LANGIDUNQ, *LANGIDSHR, or *HEX, blanks are returned in the variable.

[Top](#)

CL var for LANGID (3) (LANGID)

Specifies the name of the CL variable that receives the value indicating the language identifier to be used for the job. The variable must be a character variable with a minimum length of 3 characters.

[Top](#)

CL var for CNTRYID (2) (CNTRYID)

Specifies the name of the CL variable that receives the value indicating the country or region identifier to be used for the job. The variable must be a character variable with a minimum length of 2 characters.

[Top](#)

CL var for CCSID (5 0) (CCSID)

Specifies the name of a CL variable that receives the coded character set identifier value being used. The variable must be a 5-digit decimal variable specified with no decimal positions.

[Top](#)

CL var for DFTCCSID (5 0) (DFTCCSID)

Specifies the name of a CL variable that receives the default coded character set identifier value being used for the job. The variable must be a 5-digit decimal variable with no decimal positions.

[Top](#)

CL var for CHRIDCTL (10) (CHRIDCTL)

Specifies the name of a CL variable that receives the value being used as the character identifier control for the job. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for DECFMT (1) (DECFMT)

Specifies the name of a CL variable that receives the character being used as the decimal format for the job. The variable must be a character variable with a minimum length of 1 character.

[Top](#)

CL var for DATFMT (4) (DATFMT)

Specifies the name of a CL variable that receives the special value being used as the date format for the job. The variable must be a character variable with a minimum length of four characters.

[Top](#)

CL var for DATSEP (1) (DATSEP)

Specifies the name of a CL variable that receives the character being used as the date separator character for the job. The variable must be a character variable with a minimum length of one character.

[Top](#)

CL var for TIMSEP (1) (TIMSEP)

Specifies the name of a CL variable that receives the character being used as the time separator character for the job. The variable must be a character variable with a minimum length of one character.

[Top](#)

CL var for DATE (6) (DATE)

Specifies the name of the CL variable that receives the date assigned to the job by the system when the job is started. The variable must be a character variable with a minimum length of 6 characters. The job date is returned in the job-date format.

[Top](#)

CL var for CYMDDATE (7) (CYMDDATE)

Specifies the name of the CL variable that receives the date assigned to the job by the system when the job was started. The variable must be a character variable with a minimum length of 7 characters. The job date is returned in the format `CYYMMDD`, where `C` is the century, `YY` is the year, `MM` is the month and `DD` is the day.

[Top](#)

CL var for DATETIME (20) (DATETIME)

Specifies the name of a CL variable that receives the current local date and time of the job. The variable must be a character variable with a minimum length of 20 characters. The local job date and time is returned in the format `YYYYMMDDHHNNSXXXXXX` where `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the hours, `NN` is the minutes, `SS` is the seconds, and `XXXXXX` is the microseconds.

[Top](#)

CL var for DAYOFWEEK (4) (DAYOFWEEK)

Specifies the name of a CL variable that receives the current local day of the week for the job. The variable must be a character variable with a minimum length of 4 characters. The special value `*SUN`, `*MON`, `*TUE`, `*WED`, `*THU`, `*FRI`, or `*SAT` is returned.

[Top](#)

CL var for TIMZON (10) (TIMZON)

Specifies the name of a CL variable that receives the name of the current time zone description used to calculate local job time. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for TIMZONABBR (10) (TIMZONABBR)

Specifies the name of a CL variable that receives the current abbreviated, or short, name for the time zone used to calculate local job time. This value will contain either the Standard Time or Daylight Saving Time abbreviated name depending on whether or not Daylight Saving Time is in effect. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for TIMZONFULL (50) (TIMZONFULL)

Specifies the name of a CL variable that receives the current full, or long, name for the time zone used to calculate local job time. This value will contain either the Standard Time or Daylight Saving Time full name depending on whether or not Daylight Saving Time is in effect. The variable must be a character variable with a minimum length of 50 characters.

[Top](#)

CL var for TIMOFFSET (3 0) (TIMOFFSET)

Specifies the name of a CL variable that receives the current offset in minutes used to calculate local job time. This value has been adjusted for Daylight Saving Time if necessary. The variable must be a 3-digit decimal variable with no decimal positions.

[Top](#)

CL var for YEAROFS (3 0) (YEAROFS)

Specifies the name of a CL variable that receives the year offset for the job. This value specifies the number of years between the current Gregorian year and the current year in the calendar system used by the job. The range is -140 to 140. The variable must be a 3-digit decimal variable with no decimal positions.

[Top](#)

CL var for ACGCDE (15) (ACGCDE)

Specifies the name of the CL variable that receives the accounting code for the job. The variable must be a character variable with a minimum length of 15 characters.

[Top](#)

CL var for SWS (8) (SWS)

Specifies the name of the CL variable that receives the value of the eight job switches used by the job. The job switches are retrieved as a single 8-character value with each of the characters specifying a 1 or 0 as the value of the associated switch. The CL variable must be a character variable with a minimum length of 8 characters.

[Top](#)

CL var for RUNPTY (2 0) (RUNPTY)

Specifies the name of the CL variable that receives the 2-digit value, ranging from 1 through 99, that is the processing priority for the job in which this command is used. This value is the highest run priority allowed for any thread within the job. Individual threads within the job may have a lower run priority. The variable must be a 2-digit decimal variable specified with no decimal positions.

[Top](#)

CL var for TIMESLICE (7 0) (TIMESLICE)

Specifies the name of the CL variable that receives the 7-digit value, ranging from 8 through 9999999, that is the maximum number of milliseconds that a thread within this job can run when it is given processing time. The variable must be a 7-digit decimal variable specified with no decimal positions.

[Top](#)

CL var for PURGE (10) (PURGE)

Specifies the name of the CL variable that receives the special value which indicates whether this job is eligible to be moved out of main storage and placed into auxiliary storage at the end of a time slice or when entering a long wait. For additional information on job purging, refer to this parameter description under the Change Job (CHGJOB) command. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for DFTWAIT (7 0) (DFTWAIT)

Specifies the name of the CL variable that receives the 7-digit value, ranging from 1 through 9999999 (or -1 if the value is set to *NOMAX), that is the default for the maximum number of seconds that the system waits for a machine instruction to acquire a resource. The variable must be a 7-digit decimal variable specified with no decimal positions.

[Top](#)

CL var for SBMMSGQ (10) (SBMMSGQ)

Specifies the name of the CL variable that receives the name of a message queue. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for SBMMSGQLIB (10) (SBMMSGQLIB)

Specifies the name of the CL variable that receives the name of the library containing the message queue described previously. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for DDMCNV (5) (DDMCNV)

Specifies the name of a CL variable that receives the special value that indicates the action taken for distributed data management (DDM) conversations on the job. The variable must be a character variable with a minimum length of 5 characters.

[Top](#)

CL var for TSEPOOL (10) (TSEPOOL)

Specifies the name of the CL variable that receives the special value indicating whether interactive jobs are moved to another main storage pool when they reach the time slice end. The variable must be a character variable with a minimum length of 10 characters.

Top

CL var for THDRSCAFN (20) (THDRSCAFN)

Specifies the name of a CL variable that receives the special value indicating whether or not secondary threads have affinity to the same group of system resources as the initial thread. The special value *NOGROUP or *GROUP is returned in the first 10 characters. The second 10 characters contain the special value indicating the level of affinity. The special value *NORMAL or *HIGH is returned. The variable must be a character variable with a minimum length of 20 characters.

Top

CL var for RSCAFNGRP (10) (RSCAFNGRP)

Specifies the name of a CL variable that receives the special value indicating that the job has affinity to the same group of processors and memory as other jobs using the same routing entry or prestart job entry. The variable must be a character variable with a minimum length of 10 characters. The special value *NO or *YES is returned.

Top

Examples

```
RTVJOBA  NBR(&JOBNBR)  DATE(&JOBDATE)  DFTCCSID(&DFTCCSID)
```

This command retrieves the job number, job date, and default coded character set identifier for the job in which this command is run. The 6-digit job number is copied into the CL variable &JOBNBR. The job date is copied into the CL variable &JOBDATE; the values for both &JOBNBR and &JOBDATE must be 6 characters in length. The 5-digit DFTCCSID value is copied into the CL variable &DFTCCSID; this value must be 5 characters in length. The format of the date is determined by the contents of the system value QDATEFMT, which controls the system date format.

```
/* Declare Variables */
DCL  &LIBL *CHAR 2750
DCL  &CHGLIBL *CHAR 2760
/* save library list */
RTVJOBA  USRLIBL(&LIBL)
:
/* Temporarily change library list */
CHGLIBL  LIBL(MYLIB QGPL)
:
/* Build command string */
CHGVAR  &CHGLIBL ('CHGLIBL (' *CAT &LIBL *TCAT '))
/* restore library list */
CALL  QCMDEXC (&CHGLIBL 2760)
```

The above command retrieves the user portion of the library list so that it later can be restored from its temporary state, where only MYLIB and QGPL were in the user portion of the library list, to its original state.

If there are no libraries on the user portion of the library list, blanks are returned in the variable. If a library on the library list has been deleted, the value '*DELETED*' is put in the variable position for that name.

[Top](#)

Error messages

***ESCAPE Messages**

CPF098A

USRLIBL parameter size is too small.

CPF9899

Error occurred during processing of command.

[Top](#)

Retrieve Journal Entry (RTVJRNE)

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

Parameters
Examples
Error messages

The Retrieve Journal Entry (RTVJRNE) command allows you to get a particular journal entry and place the results in CL variables. The CL variables contain information, such as the sequence number of the retrieved entry, and are useful in automating certain types of recovery functions. The search for a journal entry can be restricted to an object, to a range of journal receivers, to a range of journal entries, to a journal code, to an entry type, to a job, to a program, to a user profile, or to a commit cycle identifier. Multiple limitation criteria can be specified. If more than one journal entry satisfies the search values specified, the first occurrence of a journal entry satisfying all of the specified search values is returned. If there is no journal entry satisfying the search values specified, the command ends with an escape message, and the return CL variables (RTNSEQNBRLRG, RTNSEQNBR, RTNJRNCD, RTNENTTYP, RTNRCV, RTNRCVLIB and RTNJRNE) remain the same.

The order of the search through the journal entries can be ascending or descending. The search order is determined by the value specified in the SEARCH parameter. The value for the FROM parameter must come before the value specified for the TO parameter in the specified search order.

The CL prompt for this command lists the minimum length for retrieved variables next to the correct parameters. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length, and the second number indicates the minimum number of decimal positions.

Restrictions:

- If the sequence number is reset in the range of the receivers specified, the first occurrence of FROMENTLRG or FROMENT is used, if they are specified. If TOENTLRG or TOENT is specified, the first occurrence after the FROMENTLRG or FROMENT entry is used, if FROMENTLRG or FROMENT is specified. Otherwise the first occurrence is used.
- The FILE, OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, OBJJID, JRNCD, ENTYP, JOB, PGM, USRPRF, CCIDLRG, CMTCYCID, and DEPENT parameters can be used to specify a subset of all available entries within a range of journal entries.
 - If no values are specified using these parameters, all available journal entries are retrieved.
 - If more than one of these parameters are specified, then a journal entry must satisfy all of the values specified on these parameters, except when *IGNFILSLT or *IGNOBSLT is specified on the JRNCD parameter.
 - If a journal code is specified on the JRNCD parameter and *IGNFILSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the FILE parameter.
 - If a journal code is specified on the JRNCD parameter and *IGNOBSLT is the second element of that journal code, then journal entries with the specified journal code are selected if they satisfy all selection criteria except what is specified on the OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, and OBJJID parameters.
- The JOB, PGM, and USRPRF parameters cannot be used to specify selection criteria if one or more journal receivers in the specified receiver range was attached to the journal when a receiver size option (RCVSIZOPT) or a fixed length data option (FIXLENDTA) that would have omitted this data was in effect.
- If more than the maximum number of objects is identified (32767 objects), an error occurs and no entries are retrieved. This restriction is ignored if *ALLFILE is specified or no objects are specified.

- When journal caching is being used, entries that are in the cache are not retrievable.

Top

Parameters

Keyword	Description	Choices	Notes
JRN	Journal	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Journal	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
FILE	Journalled file	Single values: *ALLFILE Other values (up to 300 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Member	<i>Name, *FIRST, *ALL, *NONE</i>	
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Object type	*FILE, *DTAARA, *DTAQ, *LIB	
	Element 3: Member, if data base file	<i>Name, *FIRST, *ALL, *NONE</i>	
OBJPATH	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Name	<i>Path name</i>	
	Element 2: Include or omit	*INCLUDE, *OMIT	
SUBTREE	Directory subtree	*NONE, *ALL	Optional
PATTERN	Name pattern	Values (up to 20 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value, *</i>	
	Element 2: Include or omit	*INCLUDE, *OMIT	
RCVRNG	Range of journal receivers	Single values: *CURRENT, *CURCHAIN Other values: <i>Element list</i>	Optional, Positional 3
	Element 1: Starting journal receiver	<i>Qualified object name</i>	
	Qualifier 1: Starting journal receiver	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
	Element 2: Ending journal receiver	Single values: *CURRENT Other values: <i>Qualified object name</i>	
	Qualifier 1: Ending journal receiver	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
FROMENTLRG	Starting large sequence number	<i>Character value, *FIRST, *LAST</i>	Optional
FROMTIME	Starting date and time	<i>Element list</i>	Optional
	Element 1: Starting date	<i>Date</i>	
	Element 2: Starting time	<i>Time</i>	
TOENTLRG	Ending large sequence number	<i>Character value, *LAST, *FIRST</i>	Optional

Keyword	Description	Choices	Notes
TOTIME	Ending date and time	<i>Element list</i>	Optional
	Element 1: Ending date	<i>Date</i>	
	Element 2: Ending time	<i>Time</i>	
SEARCH	Search	*ASCEND , *DESCEND	Optional
JRNCD	Journal codes	Single values: *ALL , *CTL Other values (up to 16 repetitions): <i>Element list</i>	Optional
	Element 1: Journal code value	A, B, C, D, E, F, J, L, M, P, Q, R, S, T, U, Y	
	Element 2: Journal code selection	*ALLSLT , *IGNFILSLT , *IGNOBSLT	
ENTTYP	Journal entry types	Single values: *ALL , *RCD Other values (up to 300 repetitions): <i>Character value</i>	Optional
JOB	Job name	Single values: *ALL , * Other values: <i>Qualified job name</i>	Optional
	Qualifier 1: Job name	<i>Name</i>	
	Qualifier 2: User	<i>Name</i>	
	Qualifier 3: Number	000000-999999	
PGM	Program	<i>Name</i> , *ALL	Optional
USRPRF	User profile	<i>Name</i> , *ALL	Optional
CCIDLRG	Commit cycle large identifier	<i>Character value</i> , *ALL	Optional
DEPENT	Dependent entries	*ALL , *NONE	Optional
OBJFID	File identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
OBJJID	Object journal identifier	Values (up to 300 repetitions): <i>Hexadecimal value</i>	Optional
ENTFMT	Entry format	*TYPE1 , *TYPE2 , *TYPE3 , *TYPE4 , *TYPE5	Optional
NULLINDLEN	Null value indicators length	1-8000	Optional
FMTMINDTA	Format minimized data	*NO , *YES	Optional
INCENT	Include entries	*CONFIRMED , *ALL	Optional
FROMENT	Starting sequence number	1-999999999, *FIRST , *LAST	Optional
TOENT	Ending sequence number	1-999999999, *LAST , *FIRST	Optional
CMTCYCID	Commit cycle identifier	1-999999999, *ALL	Optional
RTNSEQLRG	CL var for RTNSEQLRG (20)	<i>Character value</i>	Optional
RTNJRNCD	CL var for RTNJRNCD (1)	<i>Character value</i>	Optional
RTNENTTYP	CL var for RTNENTTYP (2)	<i>Character value</i>	Optional
RTNRCV	CL var for RTNRCV (10)	<i>Character value</i>	Optional
RTNRCVLIB	CL var for RTNRCVLIB (10)	<i>Character value</i>	Optional
RTNJRNE	CL var for RTNJRNE (1)	<i>Character value</i>	Optional
RTNSEQNBR	CL var for RTNSEQNBR (10 0)	<i>Decimal number</i>	Optional

Top

Journal (JRN)

Specifies the journal from which the journal entry is retrieved.

This is a required parameter.

Qualifier 1: Journal

journal-name

Specify the name of the journal.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Top

Journalled physical file (FILE)

Specifies a maximum of 300 files whose journal entries are retrieved. This parameter also specifies the file member whose journal entries are to be retrieved.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be retrieved, based on the specified file member name, the following is done:

- If the journal is a local journal, and if the specified file member currently exists on the system, the journal identifier id determined from the specified file member. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified file member does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified file member. All journal entries in the specified receiver range for those journal identifiers are retrieved. Specify the library name or *CURLIB to have entries returned for the file.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved, or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying a database file on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters:
 - Journal code D (database file-level information entries).
 - Journal code F (file member-level information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNFILSLT is specified on that journal code. If *ALLSLT is specified on that journal code, no journal entries with that code are retrieved.

Single values

***ALLFILE**

The search for the entry being retrieved is not limited to a specified file name. All journal entries are retrieved, regardless of which objects, if any, the entries are associated with.

Element 1: File

Qualifier 1: File

***ALL** Journal entries for all physical or logical files in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are retrieved. If ***ALL** is specified and the user does not have the required authority to all of the files, an error occurs, and the command ends.

file-name

Specify the name of the database physical or logical file for which a journal entry is retrieved.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Element 2: Member

***FIRST**

Journal entries for the database file and the first member in the file are retrieved. This value is not valid for remote journals.

***ALL** Journal entries for the database file and all the currently existing members in the file are retrieved.

***NONE**

Only entries for the database file are retrieved. Entries for members of the file are not retrieved.

member-name

Specify the name of the member for which an entries are retrieved.

If ***ALL** is specified for the file-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FIRST is specified on the FILE parameter, the journal entries of the first members of all applicable files in the specified library are retrieved.

Top

Objects (OBJ)

Specifies a maximum of 300 qualified object names whose journal entries are to be retrieved. The possible object types are *FILE, *DTAARA, *DTAQ, and *LIB. If *FILE is specified, this parameter also specifies the name of the file member whose journal entries are to be retrieved.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be retrieved, based on the specified object name, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are retrieved. Specify the library name or *CURLIB to have entries returned for an object.

There may be more than one journal identifier associated with a specified object within the specified receiver range. This can happen when a journaled object is deleted, and then a new object is created with the same name and journaled to the same journal.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the object name specification:
 - Journal code D (database file-level information entries).
 - Journal code E (data area information entries).
 - Journal code F (file member-level information entries).
 - Journal code Q (data queue information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Journal code Y (library information entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

Element 1: Object

Qualifier 1: Object

***ALL** Journal entries for all objects of the specified object type in the specified library (the library name must be specified) whose journaled changes are currently in the journal receiver are retrieved. The library name must be specified. If *ALL is specified and the user does not have the required authority for all objects in the library, a message is sent and the command ends.

object-name

Specify the name of the object whose journaled changes are to be retrieved.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the name of the library to be searched.

Element 2: Object type

Specify the object type of the object whose journaled changes are to be retrieved.

***FILE** Entries for database files and database file members are retrieved.

***DTAARA**

Entries for data areas are retrieved.

***DTAQ**

Entries for data queues are retrieved.

***LIB** Entries for libraries are retrieved.

Element 3: Member, if data base file

Specify the name of the member in the file whose journal entries are to be retrieved. If *ALL is specified for the first part of this parameter, the value specified for the member name is used for all applicable files in the library. For example, if *FIRST is specified, the journal entries of the first member of all applicable files in the specified library are retrieved.

Note: If the specified object type is not *FILE, the member name element value is ignored.

***FIRST**

Journal entries for the database file and the first member in the file are retrieved. This value is not valid for remote journals.

***ALL** Journal entries for the database file and all the currently existing members in the file are retrieved.

***NONE**

Only entries for the database file are retrieved. Entries for members of the file are not retrieved.

member-name

Specify the name of the member for which an entries are retrieved.

If *ALL is specified for the object-name element, this member name is used for all applicable files in the library. For example, if library-name/*ALL *FILE *FIRST is specified on the OBJ parameter, the journal entries of the first members of all applicable files in the specified library are retrieved.

Top

Objects (OBJPATH)

Specifies a maximum of 300 objects whose journal entries are to be retrieved. Only objects whose path name identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" (/), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

This parameter is not valid for remote journals.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

Only objects that are currently linked with the specified path name and have a journal identifier associated with them are used in journal entry selection. If the specified object does exist, the journal identifier associated with that link is used for journal entry selection. If a specified object does not exist or does not have a journal identifier associated with it, that link is not used in selecting journal entries and no error is sent.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the object name specification:
 - Journal code B (integrated file system information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

Element 1: Name

path-name

Entries for objects identified by the path name are retrieved.

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

Additional information about path name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 2: Include or omit

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

*INCLUDE

The objects that match the object name pattern are to be included in determining what journal entries are retrieved, unless overridden by an *OMIT specification.

*OMIT

The objects that match the object name pattern are not to be included in determining what journal entries are retrieved. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Directory subtree (SUBTREE)

Specifies whether the directory subtrees are included in determining the objects for which journal entries are to be retrieved.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

*NONE

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

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

Once the command has begun processing a specific directory subtree, the objects which will be found and processed may be affected by operations that update the organization of objects within the specified directory tree. This includes, but is not limited to, the following:

- Adding, removing, or renaming object links
- Mounting or unmounting file systems
- Updating the effective root directory for the process calling the command
- Updating the contents of a symbolic link

In order to process the directory subtree, the system code may increase the process-scoped maximum number of file descriptors that can be opened during processing. This is done so that the command is not likely to fail due to a lack of descriptors. This process-scoped maximum value is not reset when the command completes.

Top

Name pattern (PATTERN)

Specifies a maximum of 20 patterns to be used to include or omit objects for which journal entries are to be retrieved.

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

If the Name Pattern parameter is not specified the default will be to match all patterns.

Note: This parameter is only valid if one or more path names were specified on the OBJPATH parameter.

Element 1: Pattern

'*' All objects that match the input OBJPATH parameter are to be included.

name-pattern

Specify the pattern to be used to include or omit objects for which journal entries are retrieved. Only the last part of the path name will be considered for the name pattern match. Path name delimiters are not allowed in the name pattern.

If the Name Pattern parameter is not specified the default will be to match all patterns.

Additional information about path name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 2: Include or omit

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

***INCLUDE**

The objects that match the object name pattern are included in the operation, unless overridden by an *OMIT specification.

***OMIT**

The objects that match the object name pattern are not to be included in the operation. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Range of journal receivers (RCVRNG)

Specifies the starting (first) and ending (last) journal receivers used in the search for a journal entry to be retrieved. The system starts the search with the starting journal receiver (as specified by the first value) and proceeds through the receiver chain until the ending journal receiver (as specified by the second value) is processed.

If *ASCEND is specified for the **Search** prompt (SEARCH parameter), journal receivers must be specified in the order of oldest to newest. If *DESCEND is specified for the **Search (SEARCH)** parameter, journal receivers must be specified in the order of newest to oldest.

Note: If the maximum number of receivers (2045) in the range is surpassed, an error occurs and no journal entries are retrieved.

Single values

***CURRENT**

The journal receiver that is currently attached when starting to retrieve journal entries is used.

***CURCHAIN**

The journal receiver chain that includes the journal receiver that is currently attached when starting to retrieve journal entries is used. This receiver chain does not cross a break in the chain. If there is a break in the chain, the receiver range is from the most recent break in the chain through the receiver that is attached when starting to retrieve journal entries.

Element 1: Starting journal receiver

Qualifier 1: Starting journal receiver

starting-journal-receiver-name

Specify the name of the first journal receiver that contains journal entries to be retrieved.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the library where the journal receiver is located.

Element 2: Ending journal receiver

Qualifier 1: Ending journal receiver

***CURRENT**

The journal receiver that is currently attached when starting to retrieve journal entries is used.

ending-journal-receiver

Specify the name of the last journal receiver containing journal entries to be searched. If the end of the receiver chain is reached before a receiver of this name is found, an error message is sent and no journal entry is retrieved.

Qualifier 2: Library

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

***CURLIB**

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

library-name

Specify the library where the journal receiver is located.

Top

Starting large sequence number (FROMENTLRG)

Specifies the first journal entry considered for retrieval.

Note: You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

***FIRST**

The first journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENTLRG(*FIRST) or TOENT(*FIRST) is also specified.

***LAST**

The last journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENTLRG(*LAST) or TOENT(*LAST) is also specified.

starting-sequence-number

The journal entry with the assigned sequence number is the first entry considered for retrieval. The possible range is 1 to 18,446,744,073,709,551,600.

Top

Starting date and time (FROMTIME)

Specifies the date and time of the first journal entry considered for retrieval. The first journal entry found with the specified date and time or the next later journal entry is the starting point for the search.

Element 1: Starting date

starting-date

Specify the date. The starting date and time of the first journal entry occurring at or after the specified date and time becomes the starting point for the search.

Element 2: Starting time

starting-time

Specify the time. The starting date and time of the first journal entry occurring at or after the specified date and time becomes the starting point for the search.

The time can be specified in 24-hour format with or without a time separator:

- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds.

Top

Ending large sequence number (TOENTLRG)

Specifies the last journal entry considered for retrieval.

Note: You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

*LAST

The search continues until the last journal entry in the journal receiver range specified is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is valid only if FROMENTLRG(*LAST) or FROMENT(*LAST) is also specified.

*FIRST

The search continues until the first journal entry in the journal receiver range specified is processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if FROMENTLRG(*LAST) or FROMENT(*FIRST) is also specified.

ending-sequence-number

Specify the sequence number of the final journal entry considered for retrieval. The possible range is 1 to 18,446,744,073,709,551,600.

Note: The values specified for the FROMENT and TOENT parameter can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

Top

Ending date and time (TOTIME)

Specifies the date and time of the last entry considered for retrieval. The first journal entry found with the specified date and time, or the latest earlier journal entry is the ending point for the search.

Element 1: Ending date

ending-date

Specify the ending date. The ending date and time of the journal entry occurring at or before the specified date and time becomes the ending point for the search.

Element 2: Ending time

ending-time

Specify the time. The ending date and time of the journal entry occurring at or before the specified date and time becomes the ending point for the search.

The time can be specified in 24-hour format with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Search (SEARCH)

Specifies the order in which the journal entries are searched to retrieve an entry.

*ASCEND

The journal entries are searched in ascending order (from the oldest entry to the newest entry).

*DESCEND

The journal entries are searched in descending order (from the newest entry to the oldest entry).

Top

Journal codes (JRNCDE)

Specifies the journal codes of the journal entries being considered for retrieval.

Single values

*ALL The search for the entry is not limited to a specified journal code.

*CTL The journal entries considered for retrieval are those used to control the journal functions. The journal codes are J and F.

Element 1: Journal code value

journal-code

Specify the journal code to which journal entries are limited. Only journal entries with the specified journal code are retrieved.

An explanation of the journal codes that can be specified is in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Element 2: Journal code selection

*ALLSLT

The journal entries with the specified journal code are retrieved only if all other selection parameters are satisfied.

*IGNFILSLT

Journal entries having the specified journal code are retrieved only if all selection parameters, except the FILE parameter, are satisfied.

Note: This value is not valid for journal codes D, F, and R. This value is not valid if the OBJ, OBJPATH, OBJFID, or OBJJID parameters are specified.

*IGNOBSLT

Journal entries having the specified journal code are retrieved only if all selection parameters are satisfied except OBJ, OBJPATH, OBJFID, SUBTREE, PATTERN, and OBJJID.

Note: This value is not valid for journal codes B, D, E, F, Q, R, and Y. This value is not valid if the FILE parameter is specified.

Top

Journal entry types (ENTTYP)

Specifies whether to limit the journal entries retrieved to those of a specified journal entry type.

Single values

***ALL** The search for the entry is not limited to a particular entry type.

***RCD** Only entries that have an entry type for record-level operations are retrieved. The following entry types are valid: BR, DL, DR, IL, PT, PX, UB, UP, and UR.

Other values

entry-type

Specify the entry type that limits the search for the entry. Only journal entries that contain the specified entry type are considered for retrieval. Up to 300 valid entry types can be specified. More information on entry types is in the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Top

Job name (JOB)

Specifies that the journal entries searched for retrieval are limited to the journal entries for the specified job.

Single values

***ALL** The search is not limited to entries for a specified job.

***** The search is limited to entries for the current job.

Other values

job-identifier

Specify the job name, the user name, and the job number of the job to use. You can also specify that the job name only, or that the job name and the user name be used.

job-name

Specify the job name of the job.

user-name

Specify the user name of the job.

job-number

Specify the system-assigned job number.

Top

Program (PGM)

Specifies that the journal entries considered for retrieval are limited to the journal entries created by the specified program.

***ALL** The search is not limited to entries for a specified program.

program-name

Specify the name of the program whose journal entries are considered for retrieval.

Top

User profile (USRPRF)

Specifies that the journal entries considered for retrieval are limited to the journal entries for a specified user profile.

***ALL** The retrieval of journal entries is not limited to entries for a specified user profile.

user-profile-name

Specify the name of the user profile whose journal entries are considered for retrieval.

Top

Commit cycle large identifier (CCIDLRG)

Specifies that the journal entries considered for retrieval are limited to the journal entries containing the specified commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

***ALL** The search is not limited to entries for a specified commit cycle identifier.

commit-cycle-identifier

Specify the commit cycle identifier of the journal entries to be retrieved. The possible range is 1 to 18,446,744,073,709,551,600.

Top

Dependent entries (DEPENT)

Specifies whether to retrieve the journal entries recording actions

- that occur as a result of a trigger program
- on records that are part of a referential constraint
- that will be ignored during an Apply Journalized Changes (APYJRNCHG) or Remove Journalized Changes (RMVJRNCHG) operation.

***ALL** The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journalized Changes operations are retrieved.

***NONE**

The journal entries relating to trigger programs, referential constraints and the entries which will be ignored by an Apply or Remove Journalized Changes operations are not retrieved.

Top

File identifier (OBJFID)

Specifies a maximum of 300 file identifiers (FID) for which journal entries are to be retrieved. FIDs are a unique identifier associated with integrated file system related objects. This field is input in hexadecimal format. Only objects whose FID identifies an object of type *STMF, *DIR or *SYMLNK that are in the "root" (/), QOpenSys, and user-defined file systems are supported. All other objects are ignored.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

To determine which journal entries are to be received, based on the specified file identifier, the following is done:

- If the journal is a local journal, and if the specified object currently exists on the system, the journal identifier is determined from the specified object. All journal entries in the specified receiver range for that journal identifier are retrieved.
- If the journal is a remote journal, or if the specified object does not currently exist on the system, the specified receiver range is searched to determine all possible journal identifiers that are associated with the specified object. All journal entries in the specified receiver range for those journal identifiers are retrieved.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.
2. When specifying an object on this parameter, journal entries with the following journal code values are retrieved only if they satisfy the values specified on the other parameters in addition to the FID specification:
 - Journal code B (integrated file system information entries).
 - Journal code U (user-generated entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are retrieved.

file-identifier

Entries for objects identified with the FID are retrieved.

Top

Object journal identifier (OBJJID)

Specifies a maximum of 300 journal identifiers for which journal entries are to be retrieved. This field is input in hexadecimal format. Hexadecimal zero is not valid.

Either the FILE parameter may be specified, or one or more of the object parameters (OBJ, OBJPATH, OBJFID, or OBJJID) may be specified, but not both.

Notes:

1. The journal identifier is the unique identifier associated with the object when journaling is started for that object. The journal identifier stays constant, even if the object is renamed, moved or restored. See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information.

2. When specifying a journal identifier on this parameter, journal entries with the following journal code values are converted for output only if they satisfy the values specified on the other parameters in addition to the journal identifier specification:
 - Journal code B (integrated file system information entries).
 - Journal code D (database file-level information entries).
 - Journal code E (data area information entries).
 - Journal code F (file member-level information entries).
 - Journal code J (journal receiver information entries).
 - Journal code Q (data queue information entries).
 - Journal code R (record-level information entries).
 - Journal code U (user-generated entries).
 - Journal code Y (library information entries).
 - Other journal codes, if *IGNOBSLT is the second element of the journal code. If *ALLSLT is the second element of the journal code, no journal entries with that code are converted for output.

journal-identifier

Entries for objects associated with the specified journal identifier are retrieved.

Top

Entry format (ENTFMT)

Specifies the format of the retrieved journal entry. For a description of what is represented by each of the fields in the journal entry, see the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. Search for "journal entry finder".

Lists showing detailed information on the format of the retrieved journal entries are in the RTNJRNE parameter description.

Note: If ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is not specified, the NULLINDLEN parameter must be specified.

Note: If the **Receiver size options** field (RCVSIZOPT) for the journal was specified as *MAXOPT3, the sequence number, commit cycle identifier, the count of entries applied or removed or relative record number fields can reach a maximum value of 18,446,744,073,709,551,600. The length of these fields for the ENTFMT(*TYPE1), ENTFMT(*TYPE2), ENTFMT(*TYPE3), and ENTFMT(*TYPE4) formats is defined to hold a 10 digit number. Only the *TYPE5 format has fields large enough to hold this maximum value. When returning these fields for a format other than *TYPE5, if a sequence number, commit cycle identifier, or count of entries applied or removed or relative record number larger than 10 digits is found, the field is set to -1 for that entry.

*TYPE1

The retrieved journal entries are formatted to include the minimum information that can be specified.

*TYPE2

The retrieved journal entries include the information returned when ENTFMT(*TYPE1) is specified, and the user profile field, which gives the name of the user who logged the retrieved journal entries, and the name of the system on which the entry was sent.

*TYPE3

The retrieved journal entries include the information returned when ENTFMT(*TYPE2) is specified, and the null value indicators.

***TYPE4**

The retrieved journal entries include the information returned when ENTFMT(*TYPE3) is specified, the journal identifier, the physical file trigger indicator, and the referential constraint indicator.

***TYPE5**

The retrieved journal entries include the information returned when OUTFILFMT(*TYPE4) is specified, in addition to the program library name, the program library ASP device name, the program library ASP number, the system sequence number, the logical unit of work, the transaction identifier, the thread identifier, the remote address, the address family, the remote port, the arm number, the receiver name, the receiver library name, the receiver library ASP device name and the receiver library ASP number.

Top

Null value indicators length (NULLINDLEN)

Specify the length used for the null value indicators portion of the retrieved entry. This parameter is not valid if ENTFMT(*TYPE1) or ENTFMT(*TYPE2) is specified. Valid values range from 1 to 8000 bytes. If the retrieved journal entry has fewer null value indicators than the specified field length, the trailing bytes in the null value indicators field is set to 'F0'X.

Null value indicators are present in journal entries for record level operations as follows:

1. The corresponding physical file has null capable fields.
2. The record image has been minimized in the entry specific data.

If the record image has not been minimized in the entry specific data, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

- 'F0'X = Corresponding field is not null.
- 'F1'X = Corresponding field is null.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*YES) was specified on the RCVJRNE command, then there is one null value indicator per field in the physical file. Each indicator is one character long and can be either:

- 'F0'X = Corresponding field is not null.
- 'F1'X = Corresponding field is null.
- 'F9'X = Corresponding field was not changed and the default value for the field is returned.

If the record image has been minimized on file field boundaries in the entry specific data and FMTMINDTA(*NO) was specified on the RTVJRNE command, then an internal value is returned for the null value indicator.

Top

Format minimized data (FMTMINDTA)

Specifies whether entry specific data which has been minimized on field boundaries will be returned in a readable format.

***NO** The journal entries which have entry specific data that has been minimized on field boundaries will not be returned in a readable format. Therefore, the entry specific data may not be viewable.

***YES** The journal entries which have entry specific data that has been minimized on field boundaries will be returned in a readable format. Therefore, the entry specific data is viewable and may be

used for auditing purposes. The fields that were changed are accurately reflected. The fields that were not changed and were not recorded display default data and are indicated by a value of 'F9'X in the null value indicators field.

Top

Include entries (INCENT)

Specifies whether only the confirmed, or both the confirmed and unconfirmed, journal entries are retrieved. This parameter only applies when converting journal entries for output from a remote journal.

Confirmed entries are those journal entries which have been sent to this remote journal and the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is known.

Unconfirmed entries are those journal entries which have been sent to this remote journal, but the state of the Input/Output (I/O) to auxiliary storage for the same journal entries on the local journal is not known, or the object name information for those journal entries is not yet known to the remote journal. Unconfirmed journal entries can only exist within the attached receiver of a remote journal. This only applies if synchronous delivery mode is being used for a particular remote journal.

*CONFIRMED

Only those journal entries which have been confirmed are retrieved.

***ALL** All confirmed and unconfirmed journal entries are retrieved.

Top

Starting sequence number (FROMENT)

Specifies the first journal entry considered for retrieval.

Note: You can specify a value for either the **Starting sequence number (FROMENT)** parameter or the **Starting large sequence number (FROMENTLRG)** parameter, but not for both.

*FIRST

The first journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*DESCEND) is specified, FROMENT(*FIRST) is valid only if TOENTLRG(*FIRST) or TOENT(*FIRST) is also specified.

***LAST**

The last journal entry in the specified journal receiver range is the first entry considered for retrieval. If SEARCH(*ASCEND) is specified, FROMENT(*LAST) is valid only if TOENTLRG(*LAST) or TOENT(*LAST) is also specified.

starting-sequence-number

The journal entry with the assigned sequence number is the first entry considered for retrieval. The possible range is 1 to 9,999,999,999.

Top

Ending sequence number (TOENT)

Specifies the last journal entry considered for retrieval.

Note: You can specify a value for either the **Ending sequence number (TOENT)** parameter or the **Ending large sequence number (TOENTLRG)** parameter, but not for both.

***LAST**

The search continues until the last journal entry in the journal receiver range specified is processed. If SEARCH(*DESCEND) is specified, TOENT(*LAST) is valid only if FROMENTLRG(*LAST) or FROMENT(*LAST) is also specified.

***FIRST**

The search continues until the first journal entry in the journal receiver range specified is processed. If SEARCH(*ASCEND) is specified, TOENT(*FIRST) is only valid if FROMENTLRG(*LAST) or FROMENT(*FIRST) is also specified.

ending-sequence-number

Specify the sequence number of the final journal entry considered for retrieval. The possible range is 1 to 9,999,999,999.

Note: The values specified for the FROMENT and TOENT parameter can be the same. For example, FROMENT(234) and TOENT(234) can be specified.

Top

Commit cycle identifier (CMTCYCID)

Specifies that the journal entries considered for retrieval are limited to the journal entries containing the specified commit cycle identifier. A commit cycle consists of all journal entries sharing the same commit cycle identifier. A journal entry's commit cycle identifier can be displayed by using the Display Journal (DSPJRN) command and entering option five.

Note: You can input a value for either the **Commit cycle identifier** field (CMTCYCID) or the **Commit cycle large identifier** field (CCIDLRG) but not for both.

***ALL** The search is not limited to entries for a specified commit cycle identifier.

commit-cycle-identifier

Specify the commit cycle identifier of the journal entries to be retrieved. The possible range is 1 to 9,999,999,999.

Top

CL var for RTNSEQLRG (20) (RTNSEQLRG)

Specifies the name of the program CL character variable into which the journal entry sequence number of the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence number is not copied into the program. The specified variable must be a character variable that has a length of twenty positions. If the retrieved sequence number is shorter than the length of the field, the number is padded on the right with blanks.

Top

CL var for RTNJRNCD (1) (RTNJRNCD)

Specifies the name of the program CL character variable into which the journal code of the retrieved journal entry is copied. If a CL variable name is not specified, the journal code of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 1 character. If the length of the variable is longer than 1 character, it is padded on the right with blanks.

Top

CL var for RTNENTTYP (2) (RTNENTTYP)

Specifies the name of the program CL character variable into which the entry type of the retrieved journal entry is copied. If a CL variable name is not specified, the entry type of the retrieved journal entry is not copied into the program. The specified variable must be a character variable with a minimum length of 2 characters. If the length of the variable is longer than 2 characters, it is padded on the right with blanks.

Top

CL var for RTNRCV (10) (RTNRCV)

Specifies the name of the program CL character variable into which the journal receiver name from which the returned journal entry was retrieved is copied. If the CL variable name is not specified, the journal receiver name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

Top

CL var for RTNRCVLIB (10) (RTNRCVLIB)

Specifies the name of the CL character variable into which the name of the library containing the receiver of the retrieved journal entry is copied. If the CL variable name is not specified, the journal receiver library name is not copied into the program. The specified variable must be a character variable with a minimum length of 10 characters. If the length of the variable is longer than 10 characters, it is padded on the right with blanks.

Top

CL var for RTNJRNE (1) (RTNJRNE)

Specifies the name of the program CL character variable into which the retrieved journal entry is copied. If a CL variable name is not specified, the retrieved journal entry is not copied into the program. The specified variable must be a character variable. If the retrieved journal entry is longer than the variable's field length, the entry is truncated. If the entry is shorter, it is padded on the right with blanks.

The following lists show detailed information on the format of the retrieved journal entries.

The journal entry can be retrieved in one of the following possible formats:

If ENTFMT(*TYPE1) is specified, then the format of the fields in the retrieved entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (1)	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
DATE	TYPE(*CHAR) LEN(6)
TIME	TYPE(*DEC) LEN(6 0)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN (2)	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (3)	TYPE(*DEC) LEN(10 0)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(6)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) LEN(up to 32642)

Notes:

(1) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.

(2) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.

(3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

If ENTFRMT(*TYPE2) is specified, then the format of the fields in the retrieved entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (1)	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
DATE	TYPE(*CHAR) LEN(6)
TIME	TYPE(*DEC) LEN(6 0)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN (2)	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (3)	TYPE(*DEC) LEN(10 0)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(18)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) LEN(up to 32612)
Notes:	
(1) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.	
(2) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.	
(3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.	

Note: If ENTFRMT(*TYPE3) is specified, the following information is not available in this format:

- Incomplete Data indicating if the journal entry data is incomplete due to either LOB fields or Byte Stream File operations.
- Minimized Entry Specific Data indicating if the journal entry has minimized entry specific data because the journal had MINENTDTA specified for the object type of the journal entry.

See the Journal management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information on the incomplete data indicator, the minimized entry specific data indicator, and these journal entries.

If ENTFRMT(*TYPE3) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (3)	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN (4)	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (5)	TYPE(*DEC) LEN(10)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 32618 minus (field length)) (2)

Notes:

- (1) The length of this field is the length specified on the NULLINDLEN parameter.
- (2) The length of this portion of the entry depends on the length specified on the RTNJRNE parameter and the length specified on the NULLINDLEN parameter.
- (3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

If ENTFMT(*TYPE4) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER (3)	TYPE(*DEC) LEN(10 0)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN (4)	TYPE(*DEC) LEN(10 0)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID (5)	TYPE(*DEC) LEN(10)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
JOURNAL IDENTIFIER	TYPE(*CHAR) LEN(10)
REF CONSTRAINT	TYPE(*CHAR) LEN(1)
TRIGGER	TYPE(*CHAR) LEN(1)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
IGNORE DURING	TYPE(*CHAR) LEN(1)
APYJRNCHG OR RMVJRNCHG	
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
RESERVED	TYPE(*CHAR) LEN(5)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 32598) minus (field length)) (2)

Notes:

- (1) The length of this field is the length specified on the NULLINDLEN parameter.
- (2) The length of this portion of the entry depends on the length specified on the RTNJRNE parameter and the length specified on the NULLINDLEN parameter.
- (3) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the sequence number is larger than 10 digits.
- (4) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the count of entries applied or removed or relative record number is larger than 10 digits.
- (5) When the RCVSIZOPT of the journal is *MAXOPT3, this field will be set to -1 if the commit cycle identifier is larger than 10 digits.

If ENTFMT(*TYPE5) is specified and a value is specified on the NULLINDLEN parameter, the format of the retrieved journal entry is as follows:

Field Name	Field Attributes
ENTRY LENGTH	TYPE(*DEC) LEN(5 0)
SEQUENCE NUMBER	TYPE(*CHAR) LEN(20)
JOURNAL CODE	TYPE(*CHAR) LEN(1)
JOURNAL ENTRY TYPE	TYPE(*CHAR) LEN(2)
TIMESTAMP	TYPE(*TIMESTAMP) LEN(26)
JOB NAME	TYPE(*CHAR) LEN(10)
USER NAME	TYPE(*CHAR) LEN(10)
JOB NUMBER	TYPE(*DEC) LEN(6 0)
PROGRAM NAME	TYPE(*CHAR) LEN(10)
PROGRAM LIBRARY NAME	TYPE(*CHAR) LEN(10)
PROGRAM ASP DEVICE NAME	TYPE(*CHAR) LEN(10)
PROGRAM ASP NUMBER	TYPE(*DEC) LEN(5 0)
OBJECT NAME	TYPE(*CHAR) LEN(10)
OBJECT LIBRARY	TYPE(*CHAR) LEN(10)
MEMBER NAME	TYPE(*CHAR) LEN(10)
COUNT/RRN	TYPE(*CHAR) LEN(20)
FLAG	TYPE(*CHAR) LEN(1)
COMMIT CYCLE ID	TYPE(*CHAR) LEN(20)
USER PROFILE	TYPE(*CHAR) LEN(10)
SYSTEM NAME	TYPE(*CHAR) LEN(8)
JOURNAL IDENTIFIER	TYPE(*CHAR) LEN(10)
REF CONSTRAINT	TYPE(*CHAR) LEN(1)
TRIGGER	TYPE(*CHAR) LEN(1)
INCOMPLETE DATA	TYPE(*CHAR) LEN(1)
IGNORE DURING APYJRNCHG OR RMVJRNCHG	TYPE(*CHAR) LEN(1)
MINIMIZED ENTRY DATA	TYPE(*CHAR) LEN(1)
OBJECT INDICATOR	TYPE(*CHAR) LEN(1)
SYSTEM SEQUENCE NUMBER	TYPE(*CHAR) LEN(20)
RECEIVER NAME	TYPE(*CHAR) LEN(10)
RECEIVER LIBRARY NAME	TYPE(*CHAR) LEN(10)
RECEIVER ASP DEVICE NAME	TYPE(*CHAR) LEN(10)
RECEIVER ASP NUMBER	TYPE(*DEC) LEN(5 0)
ARM NUMBER	TYPE(*DEC) LEN(5 0)
THREAD IDENTIFIER	TYPE(*CHAR) LEN(8)
THREAD IDENTIFIER HEX	TYPE(*CHAR) LEN(16)
ADDRESS FAMILY	TYPE(*CHAR) LEN(1)
REMOTE PORT	TYPE(*DEC) LEN(5 0)
REMOTE ADDRESS	TYPE(*CHAR) LEN(46)
LOGICAL UNIT OF WORK	TYPE(*CHAR) LEN(39)
TRANSACTION IDENTIFIER	TYPE(*CHAR) LEN(140)
OBJECT TYPE	TYPE(*CHAR) LEN(7)
FILE TYPE INDICATOR	TYPE(*CHAR) LEN(1)
NESTED COMMIT LEVEL	TYPE(*DEC) LEN(7 0)
RESERVED	TYPE(*CHAR) LEN(5)
NULL VALUE INDICATORS	TYPE(*CHAR) field-length (1)
ENTRY-SPECIFIC DATA	TYPE(*CHAR) ((up to 32212) minus (field length)) (2)

Notes:

(1) The length of this field is the length specified on the NULLINDLEN parameter.

(2) The length of this portion of the entry depends on the length specified on the RTNJRNE parameter and the length specified on the NULLINDLEN parameter.

Top

CL var for RTNSEQNBR (10 0) (RTNSEQNBR)

Specifies the name of the program CL decimal variable into which the journal entry sequence number of the retrieved journal entry is copied. If a CL variable name is not specified, the journal entry sequence number is not copied into the program. The specified variable must be a decimal variable that has a length of ten positions with no decimal positions. If the sequence number is larger than 9,999,999,999, this value is set to -1, left-aligned and padded on the right with blanks. Use the **CL var for RTNSEQLRG (20) (RTNSEQLRG)** parameter to retrieve a value larger than 9,999,999,999.

Top

Examples

Example 1

Assume the following variables are specified:

```
DCL &SEQ      TYPE(*DEC)  LEN(10 0)
DCL &JRNENT   TYPE(*CHAR) LEN(200)
DCL &RCVNAME  TYPE(*CHAR) LEN(10)
DCL &RCVLIB   TYPE(*CHAR) LEN(10)
```

and this command is run:

```
RTVJRNE  JRN(MYLIB/JRNA) ENTYP(PR) +
         RTNSEQNBR(&SEQ) RTNJRNE(&JRNENT)
```

Since no starting journal entry is specified in this command, the first entry from the journal receiver that is currently attached to the journal JRNA in the library MYLIB, when starting to retrieve entries, is considered for retrieval. The first entry in any receiver is always an identifier for the previously-attached receiver. This first receiver entry is known as a type PR entry, and it contains the name of the previously attached receiver in its entry-specific data. The PR entry is the first entry in ascending order in the currently attached receiver; when it is found, the entry is placed into a CL variable named &JRNENT.

Change Variable (CHGVAR) can then be used to separate the name and library of the previous journal receiver, found in the entry specific data, as follows:

```
CHGVAR  &RCVNAME  (%SST(&JRNENT 126 10))
CHGVAR  &RCVLIB   (%SST(&JRNENT 136 10))
```

Example 2

Assume the following variables are specified:

```
DCL &ENTNO     TYPE(*DEC)  LEN(10 0)
DCL &JCODE     TYPE(*CHAR) LEN(1)
DCL &ETYP      TYPE(*CHAR) LEN(2)
DCL &RCVNAME   TYPE(*CHAR) LEN(10)
DCL &RCVLIB    TYPE(*CHAR) LEN(10)
DCL &JENTRY    TYPE(*CHAR) LEN(205)
```

and this command is run:

```

RTVJRNE  JRN(MYLIB/JRNLA) OBJ(LIB1/A *FILE MBR3) +
          RCVRNG(RCVLIB/RCV30 RCVLIB/RCV27) ORDER(*DESCEND) +
          JRNCDE(R) ENTTYP(UP DL) JOB(000666/QPGMR/PRESTRT) +
          PGM(WAKEUP) USRPRF(MAC7) ENTFMT(*TYPE2) +
          RTNSEQNBR(&ENTNO) RTNJRNCDE(&JCODE) +
          RTNENTYP(&ETYP) RTNRCV(&RCVNAME) +
          RTNRCVLIB(&RCVLIB) RTNJRNE(&JENTRY)

```

This command gets a journal entry, searching in descending order the journal receiver chain from receiver RCV30 in library RCVLIB to receiver RCV27 in library RCVLIB, journaled through journal JRNLA in library MYLIB, and copies the entry into the specified CL variables. The retrieved entry is an UPDATE or DELETE entry with journal code R from member MBR3 in file A in library LIB1, created in job 000666/QPGMR/PRESTRT in program WAKEUP by user profile MAC7. The retrieved journal entry includes the user profile field. The sequence number of the retrieved entry is copied into CL variable &ENTNO. The journal code of the retrieved entry is copied into CL variable &JCODE. The entry type of the retrieved entry is copied into CL variable &ETYP. The name of the journal receiver from which the returned entry was retrieved is copied into &RCVNAME. The library name of the journal receiver from which the returned entry was retrieved is copied into &RCVLIB.

Top

Error messages

*ESCAPE Messages

CPF7002

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

CPF7006

Member &3 not found in file &1 in &2.

CPF7007

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

CPF701B

Journal recovery of an interrupted operation failed.

CPF705C

INCENT(*ALL) not allowed for a local journal.

CPF7053

Values for RCVRNG parameter not correct; reason code &1.

CPF7054

FROM and TO values not valid.

CPF7055

Maximum number of objects exceeded.

CPF7057

*LIBL not allowed with *ALL or *ALLLIB or a generic name.

CPF7060

Object not journaled in specified receiver range.

CPF7061

Conversion of journal entries failed.

CPF7062

No entries converted or received from journal &1.

CPF7065
Entry type (ENTTYP) not valid for journal code (JRNCDE).

CPF7072
Retrieval of journal entry failed.

CPF7073
No entry retrieved from journal &1 in &2.

CPF7074
RCVRNG for specified SEARCH not valid.

CPF708D
Journal receiver found logically damaged.

CPF709C
JOB, PGM, and USRPRF not valid for receiver range.

CPF70A9
OBJPATH parameter not valid for a remote journal.

CPF70AC
Object for file identifier &1 not found.

CPF70AE
Member *FIRST not allowed for a remote journal.

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9803
Cannot allocate object &2 in library &3.

CPF9809
Library &1 cannot be accessed.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9825
Not authorized to device &1.

Top

Retrieve Library Description (RTVLIBD)

Where allowed to run: Compiled CL program or interpreted
 REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

Parameters
 Examples
 Error messages

The Retrieve Library Description (RTVLIBD) command retrieves the description of a library and journaling information for the library. The values are returned (copied) to the specified variables in the program.

For parameters that are returned into CL variables by this command, the parameter descriptive title/and prompt text lists the minimum length for the CL variable. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. You must have some authority other than *EXCLUDE authority for the library to retrieve the attributes of the library.
2. The actual default auditing value for objects created into the library is shown only if you have all object (*ALLOBJ) or audit (*AUDIT) special authority. If you do not have at least one of these special authorities, the default auditing value is shown as not available (*NOTAVL).

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name	Required, Positional 1
TYPE	CL var for TYPE (10)	Character value	Optional
ASP	CL var for ASP (2 0)	Decimal number	Optional
ASPDEV	CL var for ASPDEV (10)	Character value	Optional
ASPGRP	CL var for ASPGRP (10)	Character value	Optional
CRTAUT	CL var for CRTAUT (10)	Character value	Optional
CRTOBJAUD	CL var for CRTOBJAUD (10)	Character value	Optional
JRNSTS	CL var for JRNSTS (1)	Character value	Optional
JRN	CL var for JRN (10)	Character value	Optional
JRNLIB	CL var for JRNLIB (10)	Character value	Optional
JRNIMG	CL var for JRNIMG (1)	Character value	Optional
JRNOMTE	CL var for JRNOMTE (1)	Character value	Optional
INHERIT	CL var for INHERIT (1)	Character value	Optional
JRNSTRDATE	CL var for JRNSTRDATE (13)	Character value	Optional
STRJRNRCV	CL var for STRJRNRCV (10)	Character value	Optional
JRNRCVLIB	CL var for JRNRCVLIB (10)	Character value	Optional
RCVLIBASP	CL var for RCVLIBASP (10)	Character value	Optional
RCVLIBGRP	CL var for RCVLIBGRP (10)	Character value	Optional

Keyword	Description	Choices	Notes
INHRULES	CL var for INHRULES (1280)	Character value	Optional
TEXT	CL var for TEXT (50)	Character value	Optional

Top

Library (LIB)

Specifies the library for which attributes are to be retrieved. If a variable is specified, it must be 10 characters in length and contain a library name.

This is a required parameter.

name Specify the name of the library for which attributes are to be retrieved.

Top

CL var for TYPE (10) (TYPE)

Specifies a 10-character variable used to return the library type. The value PROD or the value TEST is returned.

Top

CL var for ASP (2 0) (ASP)

Specifies a decimal (2 0) variable used to return the number of the auxiliary storage pool (ASP) from which the system allocates storage for the library. The following values can be returned:

- 1 The library is in the system auxiliary storage pool.
- 2-32 The library is in a user auxiliary storage pool.
- 1 The library is in a primary or secondary auxiliary storage pool (ASP) with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the ASPDEV parameter.

Top

CL var for ASPDEV (10) (ASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library. One of the following special values can be returned:

- *N The name of the ASP device cannot be determined.
- *SYSBAS The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Top

CL var for ASPGRP (10) (ASPGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the library. The ASP group name is the name of the primary ASP within the ASP group. One of the following special values can be returned:

*N The name of the ASP device cannot be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Top

CL var for CRTAUT (10) (CRTAUT)

Specifies a 10-character variable used to return the create authority value of the library. The value *SYSVAL, *CHANGE, *ALL, *USE, or *EXCLUDE, or the name of an authorization list is returned.

Top

CL var for CRTOBJAUD (10) (CRTOBJAUD)

Specifies a 10-character variable used to return the auditing value of the library. A value of *NOTAVL will be returned if you do not have either all object (*ALLOBJ) or audit (*AUDIT) special authority. The values that can be returned include *SYSVAL, *NONE, *USRPRF, *CHANGE, *ALL, and *NOTAVL. See the **Create object auditing (CRTOBJAUD)** parameter on the Create Library (CRTLIB) command for more information.

Top

CL var for JRNSTS (1) (JRNSTS)

Specifies a 1-character variable used to return the current journaling status of the library. See the Start Journal Library (STRJRNLIB) command for more information about starting journaling for a library. The value returned can be one of the following:

'0' The library is not currently journaled.

'1' The library is currently journaled.

Other journaling-related fields may contain data even though the library is not currently journaled.

Top

CL var for JRN (10) (JRN)

Specifies a 10-character variable used to return the name of the journal that receives the journaled changes to the library, if the library is currently journaled. If the library was previously journaled but is not currently journaled, this field shows the name of the last journal to which the library was journaled. This field is blank if journaling has never been started for this library.

Top

CL var for JRNLIB (10) (JRNLIB)

Specifies a 10-character variable used to return the name of the library that contains the current or last journal that receives the journaled changes to the library, if the library is currently journaled. If the library was previously journaled but is not currently journaled, this field shows the name of the library that contains the last journal to which the library was journaled. This field is blank if journaling has never been started for this library.

Top

CL var for JRNIMG (1) (JRNIMG)

Specifies a 1-character variable used to return the journal image information. The value returned can be one of the following:

- ' ' A blank if journaling has never been started for the library.
- '0' Only *after* images are written to the journal for changes to the library.

Top

CL var for JRNOMTE (1) (JRNOMTE)

Specifies a 1-character variable used to return information regarding journal entries to be omitted. The value returned can be one of the following:

- ' ' A blank if journaling has never been started for the library.
- '0' No journal entries are omitted.

Journal entries cannot be omitted for libraries.

Top

CL var for INHERIT (1) (INHERIT)

Specifies 1-character variable used to return information regarding whether or not new journal-eligible objects created into, moved into, or restored into the library should inherit journaling from the library according to the library's journal inherit rules. The journal inherit rules can be retrieved using the **INHRULES** parameter.

Note: You should examine the journal inherit rules to determine whether or not they are overridden by the existence of a data area with the name QDFTJRN in the library, regardless of the value of this field.

The value returned can be one of the following:

- '0' New journal-eligible objects do not inherit journaling from the library.
- '1' New journal-eligible objects inherit journaling from the library according to the journal inherit rules.

Top

CL var for JRNSTRDATE (13) (JRNSTRDATE)

Specifies a 13-character variable used to return the date and time journaling was last started for the library. A value is returned in the form CYYMMDDHHMMSS where C = century; '0' indicates years 19xx and '1' indicates years 20xx; YY = year, MM = month, DD = day, HH = hour, MM = minutes and SS = seconds. This field is blank if journaling has never been started for the library.

Top

CL var for STRJRNRVCV (10) (STRJRNRVCV)

Specifies a 10-character variable used to return the name of the oldest journal receiver needed to successfully use the Apply Journalized Changes (APYJRNRCHG) command. This field is blank if either the library has never been journaled or it has not been saved and restored since journaling was started.

Top

CL var for JRNRCVLIB (10) (JRNRCVLIB)

Specifies a 10-character variable used to return the name of the library that contains the starting journal receiver for apply. This field is blank if either the library has never been journaled or it has not been saved and restored since journaling was started.

Top

CL var for RCVLIBASP (10) (RCVLIBASP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library that contains the starting journal receiver for apply. This field is blank if either the library has never been journaled or it has never been saved and restored since journaling was started. The following special values can be returned:

*N The name of the ASP device cannot be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Top

CL var for RCVLIBGRP (10) (RCVLIBGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the library that contains the starting journal receiver for apply. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the starting journal receiver library ASP device name field. This field is blank if either the library has never been journaled or it has not been saved and restored since journaling was started. The following special values can be returned:

*N The name of the ASP device cannot be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Top

CL var for INHRULES (1280) (INHRULES)

Specifies a 1280-character variable used to return the information specified for the INHRULES keyword on the Start Journal Library (STRJRNLIB) command (or the Change Journal Object (CHGJRNOBJ) command) if journaling has ever been started for this library. If journaling has never been started for this library, the default journal inherit rules will be returned. The journal inherit rules specify which journal-eligible objects created into, moved into, or restored into the library should inherit journaling from the library.

The journal inherit rules are applied when both:

- the value returned for the **INHERIT** parameter is a '1', and
- the journal inherit rules overridden field (rulesOverridden) in the journal inherit rules below is a '0'.

Each rule defines the object types and operations that determine the objects to which the rule applies. Multiple rules can be defined for the same set of objects. If multiple rules are defined that would apply to the same object type and operation, the last rule defined for that object will be applied.

The fields are returned in the order described below. Note that some of the fields must be accessed using an offset or displacement field.

1. Version: A 4-byte unsigned integer field indicating the version of this structure. The value returned can only be the following:
 - 1 This is the first release of this structure.
2. dispEntry: A 4-byte unsigned integer field containing the offset to the first journal inherit rule entry from the beginning of the journal inherit rules information.
3. lengthRules: A 4-byte unsigned integer field containing the total length of all journal inherit rules information.
4. numRules: A 4-byte unsigned integer field containing the number of journal inherit rule entries.
5. rulesOverridden: A 1-character field indicating whether or not the library's journal inherit rules will be overridden by a QDFTJRN data area that exists in the library. That is, the information in the QDFTJRN data area takes precedence over the journal inherit rules returned for the library. The value returned can be one of the following:
 - '0' The QDFTJRN data area does not override the library's journal inherit rules.
 - '1' The QDFTJRN data area overrides the library's journal inherit rules.
6. A 3-character reserved field containing hexadecimal zeros.
7. Inherit_Rules_Entry: The following structure repeats for the number of journal inherit rule entries. The displacement to the first journal inherit rule is returned in the dispEntry field above.
 - a. nextEntryDisp: A 4-byte unsigned integer field containing the displacement to the next journal inherit rule entry from the beginning of this journal inherit rule entry.
 - b. A 20-character reserved field containing hexadecimal zeros.
 - c. objectType: A 10-character field indicating the journal-eligible object type to which the rule applies. The value returned can be one of the following:
 - *ALL All object types for which journaling is supported.
 - *DTAARA
Data areas
 - *DTAQ
Data queues
 - *FILE Data base physical files
 - d. Operation: A 1-character field indicating the operations on the object type for which journaling attributes are to be inherited from the library. The value returned can be one of the following:

- '0' Journaling is inherited for the specified object types that are created into ('1'), moved into ('2'), or restored into ('3') this library. See the descriptions of each of the operations below for further information. Note that operation '0' includes operations '1', '2', and '3' but does not include operation '4'.
 - '1' Journaling is inherited for the specified object types that are created into this library if other criteria specified for this rule are satisfied.
 - '2' Journaling is inherited for the specified object types that are moved into this library if the object is not already journaled and other criteria specified for this rule are satisfied.
 - '3' Journaling is inherited for the specified object types that are restored into in this library if other criteria specified for this rule are satisfied unless the object already exists in the library or the object was journaled when it was saved. If the object already exists in the library, the journaling attributes of the existing object are not changed. If the object was journaled when it was saved, it will retain the journaling attributes it had when it was saved, if possible. If not possible (for example, the journal to which it was journaled when it was saved is not found), the restored object will inherit journaling from the library.
 - '4' Journaling is inherited for the specified object types that are restored into this library if other criteria specified for this rule are satisfied regardless of the journaling attributes of the object when it was saved unless the object already exists in the library. If the object already exists in the library, the journaling attributes of the existing object are not changed.
- e. ruleAction: A 1-character field indicating whether or not the objects that match the object type and operation in this rule will be included or omitted from the list of objects that inherit journaling from the library. The value returned can be one of the following:
- '0' All objects that match the object type and operation of this rule will be included in the list of objects that will inherit journaling from the library according to the journal inherit rules.
 - '1' All objects that match the object type and operation of this rule will be omitted from the list of objects that will inherit journaling from the library.
- f. Images: A 1-character field indicating the kinds of images that are written to the journal receiver for changes to objects that inherit journaling from the library because of this rule. The value returned can be one of the following:
- '0' The default value for each object type will be used for this journaling attribute when an object inherits journaling from the library. Database files (*FILE) will have both before and after images generated by the system ('2'). Data areas (*DTAARA) and data queues (*DTAQ) will have only after images generated by the system ('1').
 - '1' Only after images are journaled for an object that inherits journaling from the library.
 - '2' Both before and after images are journaled for an object that inherits journaling from the library. The value '2' is only valid for data areas (*DTAARA) and database files (*FILE). If '2' is specified and *ALL is specified for Object type, the system will generate both before and after images for data areas and database files and the system will only generate after images for data queues (*DTAQ).
- g. OmtJrnE: A 1-character field indicating the journal entries to be omitted (not written to the journal receiver) for changes to objects that inherit journaling from the library because of this rule. The value returned can be one of the following:
- '0' The default value for each object type will be used for this journaling attribute when an object inherits journaling from the library because of this rule. No journal entries ('1') will be omitted for data areas (*DTAARA) and data queues (*DTAQ). Open and close journal entries ('2') will be omitted for database files (*FILE).
 - '1' No journal entries will be omitted for objects that inherit journaling from the library because of this rule.
 - '2' Open and close journal entries are not written for database files (*FILE) that inherit

journaling from the library because of this rule. Specifying '2' prevents the use of TOJOB0 and TOJOB1 entries on the Apply Journalled Changes (APYJRNCHG) and Remove Journalled Changes (RMVJRNCHG) commands, but it saves some storage space in the journal receivers. The value '2' is only valid for database files (*FILE). If '2' is specified and *ALL is specified for Object type, database files will omit open and close journal entries and data areas (*DTAARA) and data queues (*DTAQ) will not omit any journal entries.

h. A 2-character reserved field containing hexadecimal zeros.

Top

CL var for TEXT (50) (TEXT)

Specifies a 50-character CL variable used to return the text description of the library.

Top

Examples

Assume that the library named TESTLIB was created as follows:

```
CRTLIB LIB(TESTLIB) CRTAUT(*ALL) TEXT('John Smith library')
```

Also assume that journaling was started for the library named TESTLIB using the following Start Journal Library (STRJRNLIB) command:

```
STRJRNLIB LIB(TESTLIB) JRN(TESTLIB/LIBJRN)
```

PGM Example

```
DCL VAR(&CRTAUT) TYPE(*CHAR) LEN(10)
DCL VAR(&JRNSTS) TYPE(*CHAR) LEN(1)
DCL VAR(&JRN) TYPE(*CHAR) LEN(10)
DCL VAR(&JRNLIB) TYPE(*CHAR) LEN(10)
RTVLIBD LIB(TESTLIB) CRTAUT(&CRTAUT) JRNSTS(&JRNSTS)
        JRN(&JRN) JRNLIB(&JRNLIB)
```

The RTVLIBD command will return:

- a value of '*ALL' in program variable &CRTAUT,
- a value of '1' in program variable &JRNSTS,
- a value of 'LIBJRN' in program variable &JRN, and
- a value of 'TESTLIB' in program variable &JRNLIB.

Top

Error messages

*ESCAPE Messages

CPF21AD

INHRULES parameter size is too small.

CPF210E
Library &1 not available for reason code &2.

CPF2115
Object &1 in &2 type *&3 damaged.

CPF2150
Object information function failed.

CPF2151
Operation failed for &2 in &1 type *&3.

CPF980B
Object &1 in library &2 not available.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9830
Cannot assign library &1.

Top

Retrieve Member Description (RTVMBRD)

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

Parameters
Examples
Error messages

The Retrieve Member Description (RTVMBRD) command is used in a CL program or REXX procedure to retrieve (return) the member-level information (in CL variables) from a database file.

The values are returned (copied) to the specified CL variables. The following kinds of member information can be retrieved:

- The library name.
- The member name.
- The file attribute.
- The file type.
- The source type.
- The source date.
- The date created.
- The expiration date.
- The member text.
- The number of nondeleted records.
- The number of deleted records.
- The open data path status (shared or not shared).
- The data space size.
- The access path size.
- The date changed.
- The date saved.
- The date restored.
- The number of data members.
- The last date used.
- The days count used.
- The date and days count was reset.

Top

Parameters

Keyword	Description	Choices	Notes
FILE	File	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: File	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MBR	Member	Single values: *FIRST , *LAST Other values: <i>Element list</i>	Optional
	Element 1: Reference member	<i>Generic name, name, *FIRSTMBR, *LASTMBR</i>	
	Element 2: Relationship	*SAME , *NEXT , *PRV	

Keyword	Description	Choices	Notes
RTNSYSTEM	CL var for RTNSYSTEM (4)	Character value	Optional
RTNLIB	CL var for RTNLIB (10)	Character value	Optional
RTNMBR	CL var for RTNMBR (10)	Character value	Optional
FILEATR	CL var for FILEATR (3)	Character value	Optional
FILETYPE	CL var for FILETYPE (5)	Character value	Optional
SRCTYPE	CL var for SRCTYPE (10)	Character value	Optional
SRCCHGDATE	CL var for SRCCHGDATE (13)	Character value	Optional
CRTDATE	CL var for CRTDATE (13)	Character value	Optional
EXPDATE	CL var for EXPDATE (7)	Character value	Optional
TEXT	CL var for TEXT (50)	Character value	Optional
NBRCURRCD	CL var for NBRCURRCD (10 0)	Decimal number	Optional
NBRDLTRCD	CL var for NBRDLTRCD (10 0)	Decimal number	Optional
SHARE	CL var for SHARE (4)	Character value	Optional
DTASPCISZ	CL var for DTASPCISZ (15 0)	Not restricted	Optional
ACCPHSIZ	CL var for ACCPHSIZ (12 0)	Not restricted	Optional
CHGDATE	CL var for CHGDATE (13)	Character value	Optional
SAVDATE	CL var for SAVDATE (13)	Character value	Optional
RSTDATE	CL var for RSTDATE (13)	Character value	Optional
NBRDTAMBR	CL var for NBRDTAMBR (2 0)	Decimal number	Optional
USEDATE	CL var for USEDATE (7)	Character value	Optional
USECOUNT	CL var for USECOUNT (5 0)	Decimal number	Optional
RESETDATE	CL var for RESETDATE (7)	Character value	Optional

Top

File (FILE)

Specifies the file that contains the member description that is retrieved.

This is a required parameter.

Note: You must have use (*USE) authority to the file and execute (*EXECUTE) authority to the library before the member description information can be retrieved.

Qualifier 1: File

name Specify the name of the file that contains the file member.

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 current entry exists in the library list, QGPL is used.

name Specify the library name to search.

Top

Member (MBR)

Specifies the file member whose description is retrieved. Either a single value (*FIRST or *LAST) or a double value (reference member and relationship) can be specified.

Single values

*FIRST

The first member in a date-ordered list is retrieved.

*LAST

The last member in a date-ordered list is retrieved.

Element 1: Reference member

*FIRSTMBR

The first member in a name ordered list is retrieved. The relationship value *SAME is required.

*LASTMBR

The last member in a name ordered list is retrieved. The relationship value *SAME is required.

name Specify the name of the reference member. The relationship of the retrieved member to the reference member is specified on the second element of this parameter (*SAME, *NEXT, or *PRV). If a variable is specified, it must be a 10-character field that contains the name of the reference member.

generic-name

Specify the starting characters of the member name followed by an asterisk. This retrieves the first member in the name ordered list that starts with the specified characters. The relationship value is required to be *SAME.

Element 2: Relationship

*SAME

The reference member is retrieved.

*NEXT

The member immediately after the reference member in a name ordered list is retrieved.

*PRV The member immediately previous to the reference member in a name ordered list is retrieved.

Top

CL var for RTNSYSTEM (4) (RTNSYSTEM)

Specifies the name of a variable used to retrieve the name of the system from which the file was retrieved. In CL programs, this should be a 4-character variable.

The values that can be returned are *LCL (file found on the local system) and *RMT (file found on a remote system).

Top

CL var for RTNLIB (10) (RTNLIB)

Specifies the name of a variable used to retrieve the name of the library in which the file containing the specified file member is located. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for RTNMBR (10) (RTNMBR)

Specifies the name of a variable used to retrieve the name of the file member whose description is being retrieved. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for FILEATR (3) (FILEATR)

Specifies the name of a variable used to retrieve the file attribute. In CL programs, this should be a 3-character variable.

The values that can be returned are *PF (physical file member) and *LF (logical file member).

[Top](#)

CL var for FILETYPE (5) (FILETYPE)

Specifies the name of a variable used to retrieve the file type. In CL programs, this should be a 5-character variable.

The values that are returned are *DATA (data file member) and *SRC (source file member).

[Top](#)

CL var for SRCTYPE (10) (SRCTYPE)

Specifies the name of a variable used to retrieve the source file member type if this is a source file member. Blanks are returned if this is not a source file member. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for SRCCHGDATE (13) (SRCCHGDATE)

Specifies the name of a 13-character CL variable used to retrieve the century, date, and time the last source file member was changed. The format is CYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Blanks are returned if no date is available. Remote non-System i5 and non-System/38 files or non-source physical file return blanks.

[Top](#)

CL var for CRTDATE (13) (CRTDATE)

Specifies the name of a variable used to retrieve the file member creation century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Top

CL var for EXPDATE (7) (EXPDATE)

Specifies the name of a variable used to retrieve the file member expiration century and date. In CL programs, this should be a 7-character variable. The format is CYYMMDD where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, and D = Day.

*NONE is returned if no date is available.

Top

CL var for TEXT (50) (TEXT)

Specifies the name of a variable used to retrieve the file member text. In CL programs, this should be a 50-character variable.

Top

CL var for NBRCURRCD (10 0) (NBRCURRCD)

Specifies the name of a variable used to retrieve the current number of nondeleted records in this file member. In CL programs, this should be a 10-position decimal variable.

If the member is a keyed logical member, the number of index entries is returned. For nonkeyed logical members, the number of nondeleted records in the based-on physical file member is returned.

For a join logical file, the number of records returned is a total of all the nondeleted records in the files being joined. This number includes those records that exist in the secondary file which do not have matching records in the primary file. If omit or select criteria is used, the number of records returned will be reduced or increased by the number of records meeting the selection criteria in the specified files.

Top

CL var for NBRDLTRCD (10 0) (NBRDLTRCD)

Specifies the name of a variable used to retrieve the current number of deleted records in this file member. In CL programs, this should be a 10-position decimal variable. Zero (0) is returned for keyed logical files. Remote non-System i5 and non-S/38 files return a value of 0. For nonkeyed logical files, the number of deleted records in the based-on physical file member is returned.

Top

CL var for SHARE (4) (SHARE)

Specifies the name of a variable used to retrieve a value indicating whether the open data path (ODP) allows sharing with other programs in the same job. In CL programs, this should be a 4-character variable.

Values for ODP sharing are *YES (ODP sharing is allowed) and *NO (ODP sharing is not allowed).

Remote non-System i5 and non-S/38 files return *NO.

Top

CL var for DTASPCSI (15 0) (DTASPCSI)

Specifies the name of a variable used to retrieve the data space size (in bytes) of this file member. In CL programs, this should be a 15-position decimal variable. Zero (0) is returned if this is a logical file member.

Top

CL var for ACCPTHSI (12 0) (ACCPTHSI)

Specifies the name of a variable used to retrieve the access path size (in bytes) for this file member. In CL programs, this should be a 12-position decimal variable. Zero (0) is returned if the file member is non-keyed. Remote non-System i5 and non-S/38 files return a value of 0.

Top

CL var for CHGDATE (13) (CHGDATE)

Specifies the name of a variable used to retrieve the file change century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Top

CL var for SAVDATE (13) (SAVDATE)

Specifies the name of a variable used to retrieve the file member and the save century, date, and time. In CL programs, this should be a 13-character variable. The format is CYYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Blanks are returned if no date is available. Remote non-System i5 and non-System/38 files return blanks.

Top

CL var for RSTDATE (13) (RSTDATE)

Specifies the name of a variable used to retrieve the file member and restore century, date, and time. In CL programs, this should be a 13-character variable. The format is CYMMDDHHMMSS where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, D = Day, H = Hour, M = Minutes, and S = Seconds.

Blanks are returned if there is no date available. Remote non-System i5 and non-System/38 files return blanks.

[Top](#)

CL var for NBRDTAMBR5 (2 0) (NBRDTAMBR5)

Specifies the name of a variable used to retrieve the number of data file members for this logical file member. In CL programs, this should be a 2-position decimal variable. If the member is a physical file member, a value of 0 is returned.

[Top](#)

CL var for USEDATE (7) (USEDATE)

Specifies the name of a variable used to return the member last used century and date. In CL programs, this should be a 7-character variable. The format is CYMMDD where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, and D = Day.

[Top](#)

CL var for USECOUNT (5 0) (USECOUNT)

Specifies the name of a variable that is used to return the number of days the member has been used. In CL programs, this should be a 5 decimal variable. If the member does not have a last used date, 0 is returned.

[Top](#)

CL var for RESETDATE (7) (RESETDATE)

Specifies the name of a variable that is used to return the century and date the days used count was last reset to 0. In CL programs, this should be a 7-character variable. The format is CYMMDD where C = Century (0 = 1940 through 1999 and 1 = 2000 through 2039), Y = Year, M = Month, and D = Day. If the days used count has not been reset, blanks are returned.

[Top](#)

Examples

Assume the user has a file named MYFILE in library MYLIB (which is the current library) with members QMEMBER, BMEMBER, ZMEMBER, and JMEMBER (created in that order).

Also assume the following variables are specified in the CL program:

```

DCL &LIB      TYPE(*CHAR) LEN(10)
DCL &MBR      TYPE(*CHAR) LEN(10)
DCL &SYS      TYPE(*CHAR) LEN(4)
DCL &MTYPE    TYPE(*CHAR) LEN(5)
DCL &CRTDATE  TYPE(*CHAR) LEN(13)
DCL &CHGDATE  TYPE(*CHAR) LEN(13)
DCL &TEXT     TYPE(*CHAR) LEN(50)
DCL &NBRRCD  TYPE(*DEC) LEN(10 0)
DCL &SIZE     TYPE(*DEC) LEN(10 0)

```

Example 1: Retrieving Member Description Values

```

RTVMBRD FILE(*CURLIB/MYFILE) MBR(BMEMBER *SAME) +
RTNLIB(&LIB) RTNSYSTEM(&SYS) +
RTNMBR(&MBR) FILEATR(&MTYPE) +
CRTDATE(&CRTDATE) TEXT(&TEXT) +
NBRCURRCD(&NBRRCD) DTASPCSI(&SIZE)

```

This command retrieves the member description for member BMEMBER of file MYFILE located using the library list. The requested information is placed in the CL variables as follows:

- The current library name (MYLIB) is placed in the CL variable named &LIB.
- The system on which MYFILE was found is placed in the CL variable named &SYS. (*LCL means the file was found on the local system, and *RMT means the file was found on a remote system.)
- The member name (BMEMBER) is placed in the CL variable named &MBR.
- The file attribute of MYFILE is placed in the CL variable named &MTYPE. (*DATA means the member is a data member, and *SRC means the file is a source member.)
- The creation date of BMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with BMEMBER is placed in the CL variable called &TEXT.
- The current number of records in BMEMBER is placed in the CL variable called &NBRRCD.
- The size of BMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

Example 1: Retrieving the Next Member Description

```

RTVMBRD FILE(&LIB/MYFILE) MBR(&MBR *NEXT) +
RTNMBR(&MBR) CRTDATE(&CRTDATE) +
TEXT(&TEXT) NBRCURRCD(&NBRRCD) +
DTASPCSI(&SIZE)

```

This command retrieves the member description for the member of file MYFILE which is "next" (in name order). The requested information is placed in the CL variables as follows:

- The next member's name after BMEMBER (JMEMBER since the file is searched in name order) in MYFILE is placed in the CL variable named &MBR.
- The creation date of JMEMBER is placed in the CL variable named &CRTDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.
- The current number of records in JMEMBER is placed in the CL variable called &NBRRCD.
- The size of JMEMBER's data space (in bytes) is placed in the CL variable called &SIZE.

The file can also be searched backwards. An example is:

```

RTVMBRD FILE(*CURLIB/MYFILE) MBR(ZMEMBER *PRV) +
RTNMBR(&MBR) CHGDATE(&CHGDATE) TEXT(&TEXT)

```

The requested information is placed in the CL variables as follows:

- The member name (QMEMBER since it is the member just previous to ZMEMBER in a name-ordered list) is placed in the CL variable named &MBR.
- The date QMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with QMEMBER is placed in the CL variable called &TEXT.

If only the first part of the member name is known, you can use a generic name (or partial name) search of the list of members, as follows:

```
RTVMBRD FILE(*LIBL/MYFILE) MBR(JM*) RTNMBR(&MBR) +
        CHGDATE(&CHGDATE) TEXT(&TEXT)
```

The requested information is placed in the CL variables as follows:

- The member name (JMEMBER since it is the first member starting with the characters JM in a name ordered list) is placed in the CL variable named &MBR.
- The date JMEMBER was last changed is placed in the CL variable named &CHGDATE.
- The text associated with JMEMBER is placed in the CL variable called &TEXT.

Top

Error messages

*ESCAPE Messages

CPF3018

Member &3 for file &1 in &2 not available.

CPF3019

File &1 in library &2 has no members.

CPF3027

File &1 in &2 not a database file.

CPF3038

Attributes for return variable &1 not valid.

CPF3039

Return variable &1 too small to hold result.

CPF3049

*NEXT or *PRV member does not exist.

CPF3051

File &1 in library &2 not available.

CPF325F

Conversion of the text failed.

CPF327B

File &1 information cannot be retrieved.

CPF8109

&8 damage on physical database file &4.

CPF8110

&8 damage on logical data base file &4 in &9. VLOG-&7.

CPF8111

&8 damage on member &9 file &4.

CPF9803

Cannot allocate object &2 in library &3.

CPF9806

Cannot perform function for object &2 in library &3.

CPF9810

Library &1 not found.

CPF9812

File &1 in library &2 not found.

CPF9815

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

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

Top

Retrieve Message (RTVMSG)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

Parameters
Examples
Error messages

The Retrieve Message (RTVMSG) command is used in a CL program or REXX procedure to retrieve a specified predefined message from a message file and to copy it into CL variables. Substitution values can be specified in the MSGDTA parameter (as a single character string containing one or more concatenated message data fields) to replace the substitution variables in the predefined message text. The program can later write the message to an output device file to be printed, for example.

The CL prompt for this command lists the minimum length for retrieved variables next to the parameters that have a minimum length. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions: The user of this command must have use (*USE) authority for the message file and *USE authority for the library in which the message file is located.

Top

Parameters

Keyword	Description	Choices	Notes
MSGID	Message identifier	<i>Name</i>	Required, Positional 1
MSGF	Message file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Message file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
MSGDTA	Message data field values	<i>Character value</i>	Optional
MSG	CL var for 1st level text	<i>Character value</i>	Optional
MSGLEN	CL var for MSGLEN (5 0)	<i>Decimal number</i>	Optional
SECLVL	CL var for 2nd level text	<i>Character value</i>	Optional
SECLVLEN	CL var for SECLVLEN (5 0)	<i>Decimal number</i>	Optional
SEV	CL var for SEV (2 0)	<i>Decimal number</i>	Optional
ALROPT	CL var for ALROPT (9)	<i>Character value</i>	Optional
LOGPRB	CL var for LOGPRB (1)	<i>Character value</i>	Optional
CCSID	Convert to CCSID	1-65535, *HEX, *JOB	Optional
MDTACCSID	Message data CCSID	1-65535, *HEX, *JOB	Optional
TXTCSSID	CL var for text CCSID (5 0)	<i>Decimal number</i>	Optional
DTACCSID	CL var for data CCSID (5 0)	<i>Decimal number</i>	Optional

Top

Message identifier (MSGID)

Specifies the message identifier of the predefined message that is being retrieved from the specified message file.

This is a required parameter.

Top

Message file (MSGF)

Specifies the message file that contains the predefined message to be retrieved.

This is a required parameter.

Qualifier 1: Message file

name Specify the name of the message file containing the message to be retrieved.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the library where the message file is located.

Top

Message data field values (MSGDTA)

Specifies the substitution values that are used in the retrieved message if the predefined message contains substitution variables. Either a character string or a CL variable containing the character string can be specified.

Top

CL var for 1st level text (MSG)

Specifies the name of the CL character variable in the program into which the text of the retrieved message is copied. If a CL variable name is not specified, the message text is not copied into the program. This is a variable length field, but most messages are designed to be less than 132 characters long.

Top

CL var for MSGLEN (5 0) (MSGLEN)

Specifies the name of the CL decimal variable in the program into which the total length of the message text available to be retrieved is copied.

The specified variable must be a decimal variable that has a length of five digits.

Top

CL var for 2nd level text (SECLVL)

Specifies the name of the CL character variable in the program into which the second level message, or message help, of the retrieved message is copied. If a variable name is not specified, the message help is not copied into the program. This is a variable length field, but most message help is designed to be less than 3000 characters long.

Top

CL var for SECLVLEN (5 0) (SECLVLEN)

Specifies the name of the CL decimal variable in the program into which the total length of the message help being retrieved is copied.

The specified variable must be a decimal variable that has a length of five positions.

Top

CL var for SEV (2 0) (SEV)

Specifies the name of the CL decimal variable into which the severity code of the retrieved message is copied. The specified variable must be a decimal variable that has a length of two positions. If a variable name is not specified, the severity code of the retrieved message is not copied into the program.

Top

CL var for ALROPT (9) (ALROPT)

Specifies the name of the CL variable into which the alert option of the retrieved message is copied. The variable must be a character variable nine positions long. If a character variable is not specified, the alert option of the retrieved message is not copied into the program.

Top

CL var for LOGPRB (1) (LOGPRB)

Specifies whether the message will be logged in the problem log. The variable must be a character variable one position long.

Top

Convert to CCSID (CCSID)

Specifies the coded character set identifier (CCSID) in which you want your message text returned. This applies only to text returned in the MSG and SECLVL parameters. When replacement data is substituted into the text returned in the MSG or SECLVL parameters, only the part of the replacement data that is defined as a character that can be converted (*CCHAR) is converted. The rest of the replacement data will not be converted. For more information about the *CCHAR field, see the Add Message Description (ADDMSGD) command.

***JOB** The retrieved message description is converted to the CCSID of the job before being returned.

***HEX** The retrieved message description is not converted before being returned.

coded-character-set-identifier

Specify the CCSID that you want your message description converted to before it is returned.

Note: The valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter> for a list of valid CCSID values. Only CCSIDs that you can change your job to are accepted.

For more information on the message handler and its use of CCSIDs, see the i5/OS globalization topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Message data CCSID (MDTACCSID)

Specifies the CCSID that the supplied message data is assumed to be in. This only applies to the parts of the replacement data that are defined as *CCHAR. The rest of the replacement data will never be converted and is assumed to have a CCSID of 65535.

***JOB** The message data supplied is assumed to be in the CCSID of the job running this command.

***HEX** The message data supplied is assumed to be 65535 and is never converted.

coded-character-set-identifier

The message data supplied is assumed to be in the CCSID specified. Valid values range from 1 through 65535. See the Globalization information in the iSeries Information Center at <http://www.ibm.com/eserver/iseres/infocenter/> for a list of valid CCSID values.

Top

CL var for text CCSID (5 0) (TXTCCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the text returned by the MSG and SECLVL parameters. The CCSID that the message description is stored in is returned if the one of the following occurs:

- If a conversion error occurs.
- If the job has a CCSID of 65535 and you did not specify the CCSID parameter.
- If you specify *JOB for the CCSID parameter.
- If the CCSID you requested the text to be converted to is 65535.

Otherwise, the CCSID you wanted the text converted to is returned. If you do not want the text converted before it is returned to you but you do want to know the CCSID that the message description is stored in, specify 65535 for the CCSID parameter. The CCSID that the message description is stored in is returned in the TXTCCSID parameter. You can also check for a conversion error by comparing the CCSID you passed in against the TXTCCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

Top

CL var for data CCSID (5 0) (DTACCSID)

Specifies the name of the CL variable, if any, used to return the coded character set identifier (CCSID) associated with the replacement data defined as *CCHAR. All other replacement data is not converted before it is returned. The CCSID specified for the MDTACCSID parameter is returned if the one of the following occurs:

- If a conversion error occurs.

- If the job has a CCSID of 65535 and you did not specify the CCSID parameter.
- If you specify *JOB for the CCSID parameter.
- If the CCSID you requested the text to be converted to is 65535.

Otherwise the CCSID you wanted the text converted to is returned. When there is no *CCHAR replacement data in the text, 65535 is returned. You can check for a conversion error by comparing the CCSID you passed in against the DTACCSID returned. If they are not equal and they are not 65535, a conversion error occurred.

Top

Examples

Example 1: Replacing Substitution Variables

```
RTVMSG  MSGID(UIN0145) MSGF(INVN) MSG(&WORK)
        MSGDTA('any old time')
```

This command retrieves the message text of the message UIN0145 stored in the INVN message file. The retrieved text is copied into the CL variable &WORK after the substitution variables are replaced with the values *any*, *old*, and *time*. This example assumes that the substitution variables &1, &2, and &3 have been defined in the message as character variables, each 4 characters long.

Example 2: Retrieving First-Level and Second-Level Message Text

```
RTVMSG  MSGID(UIN0150) MSGF(INV) MSG(&MSG)
        SECLVL(&SECLVL)
```

This command retrieves the first-level message text and second-level message text of the message UIN0150, which is stored in message file INV, and moves it into the CL variables &MSG and &SECLVL.

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.

CPF247E

CCSID &1 is not valid.

CPF2471

Length of field not valid.

CPF2499

Message identifier &1 not valid.

CPF2531

Message file &1 in &2 damaged for &3.

CPF2547

Damage to message file QCPFMSG.

CPF2548

Damage to message file &1 in &2.

CPF8126

Message file &4 in &9 damaged.

CPF9830

Cannot assign library &1.

STATUS Messages*CPF2419**

Message identifier &1 not found in message file &2 in &3.

NOTIFY Messages*CPF2465**

Replacement text of message &1 in &2 in &3 not valid for format specified.

Top

Retrieve Network Attributes (RTVNETA)

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

Parameters
 Examples
 Error messages

The Retrieve Network Attributes (RTVNETA) command is used in a CL program or REXX procedure to retrieve the network attributes of the system. The values are returned (copied) to the specified variables in the program.

Restrictions: The attributes of the network attribute and the receiving variable must be compatible.

Top

Parameters

Keyword	Description	Choices	Notes
SYSNAME	CL var for SYSNAME (8)	Character value	Optional
PNDSYSNAME	CL var for PNDSYSNAME (8)	Character value	Optional
LCLNETID	CL var for LCLNETID (8)	Character value	Optional
LCLCPNAME	CL var for LCLCPNAME (8)	Character value	Optional
LCLLOCNAME	CL var for LCLLOCNAME (8)	Character value	Optional
DFTMODE	CL var for DFTMODE (8)	Character value	Optional
NODETYPE	CL var for NODETYPE (8)	Character value	Optional
DTACPR	CL var for DTACPR (10 0)	Decimal number	Optional
DTACPRINM	CL var for DTACPRINM (10 0)	Decimal number	Optional
MAXINTSSN	CL var for MAXINTSSN (5 0)	Decimal number	Optional
RAR	CL var for RAR (5 0)	Decimal number	Optional
NETSERVER	CL var for NETSERVER (85)	Character value	Optional
ALRSTS	CL var for ALRSTS (10)	Character value	Optional
ALRPRIFP	CL var for ALRPRIFP (10)	Character value	Optional
ALRDFTFP	CL var for ALRDFTFP (10)	Character value	Optional
ALRLOGSTS	CL var for ALRLOGSTS (7)	Character value	Optional
ALRBCKFP	CL var for ALRBCKFP (16)	Character value	Optional
ALRRQSFP	CL var for ALRRQSFP (16)	Character value	Optional
ALRCTLD	CL var for ALRCTLD (10)	Character value	Optional
ALRHLCNT	CL var for ALRHLCNT (5 0)	Decimal number	Optional
ALRFTR	CL var for ALRFTR (10)	Character value	Optional
ALRFTRLIB	CL var for ALRFTRLIB (10)	Character value	Optional
MSGQ	CL var for MSGQ (10)	Character value	Optional
MSGQLIB	CL var for MSGQLIB (10)	Character value	Optional
OUTQ	CL var for OUTQ (10)	Character value	Optional
OUTQLIB	CL var for OUTQLIB (10)	Character value	Optional

Keyword	Description	Choices	Notes
JOBACN	CL var for JOBACN (10)	Character value	Optional
MAXHOP	CL var for MAXHOP (5 0)	Decimal number	Optional
DDMACC	CL var for DDMACC (10)	Character value	Optional
DDMACCLIB	CL var for DDMACCLIB (10)	Character value	Optional
PCSACC	CL var for PCSACC (10)	Character value	Optional
PCSACCLIB	CL var for PCSACCLIB (10)	Character value	Optional
DFTNETTYPE	CL var for DFTNETTYPE (10)	Character value	Optional
DFTCNNLST	CL var for DFTCNNLST (10)	Character value	Optional
ALWANYNET	CL var for ALWANYNET (10)	Character value	Optional
NWSDOMAIN	CL var for NWSDOMAIN (8)	Character value	Optional
ALWVRTAPPN	CL var for ALWVRTAPPN (10)	Character value	Optional
ALWHPRTWR	CL var for ALWHPRTWR (10)	Character value	Optional
VRTAUTODEV	CL var for VRTAUTODEV (5 0)	Decimal number	Optional
HPRPTHMR	CL var for HPRPTHMR (40)	Character value	Optional
ALWADDCLU	CL var for ALWADDCLU (10)	Character value	Optional
MDMCNTRYID	CL var for MDMCNTRYID (2)	Character value	Optional

Top

CL var for SYSNAME (8) (SYSNAME)

Specifies the name of the CL variable that receives the current system name. The variable must be a character variable with a minimum length of 8 characters.

Top

CL var for PNDSYSNAME (8) (PNDSYSNAME)

Specifies the name of the CL variable that receives the pending system name. The variable must be a character variable with a minimum length of 8 characters. If there is no pending system name, the value returned is blanks.

Top

CL var for LCLNETID (8) (LCLNETID)

Specifies the name of the CL variable that receives the local network ID. The variable must be a character variable with a minimum length of 8 characters.

Top

CL var for LCLCPNAME (8) (LCLCPNAME)

Specifies the name of the CL variable that receives the local control point name. The variable must be a character variable with a minimum length of 8 characters.

Top

CL var for LCLLOCNAME (8) (LCLLOCNAME)

Specifies the name of the CL variable that receives the default local location name. The variable must be a character variable with a minimum length of 8 characters.

Top

CL var for DFTMODE (8) (DFTMODE)

Specifies the name of the CL variable that receives the default mode name. The variable must be a character variable with a minimum length of 8 characters.

Top

CL var for NODETYPE (8) (NODETYPE)

Specifies the name of the CL variable that receives the APPN node type. The variable must be a character variable with a minimum length of 8 characters.

The following values can be returned in the CL variable:

***ENDNODE**

The node does not provide network services to other nodes but can participate in the APPN network by using the services of an attached network server or can operate in a peer environment similar to low entry networking nodes.

***NETNODE**

The node provides intermediate routing, route selection services, and distributed directory services for local users and to end nodes and low entry networking nodes that it is serving.

***BEXNODE**

The node performs as a branch extender node. The node performs as an end node in the backbone APPN network, and performs as a network node server to end nodes within its local domain.

Top

CL var for DTACPR (10 0) (DTACPR)

Specifies the name of the CL variable that receives the current level of data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the data compression level are:

- 0 *NONE - Data compression is not allowed on the session.
- 1 *REQUEST - Data compression is requested on the session by the local system. However, the request can be refused or changed to a lower compression level by the remote system. Data compression is allowed on the session if requested by the remote system.

- 2 *ALLOW - Data compression is allowed on the session by the local system if requested by a remote system. The local system does not request compression.
- 3 *REQUIRE - Data compression is required on the session. If the remote system does not change the levels of compression to the local system's exact requested levels, the session is not established. The data compression levels that the local system requires are the specified levels.

Top

CL var for DTACPRINM (10 0) (DTACPRINM)

Specifies the name of the CL variable that receives the current level of intermediate node data compression. Specify the name of the decimal variable with a minimum length of 10 digits without decimal positions.

The values that can be returned in the variable as the intermediate node data compression levels are:

- 0 *NONE - The remote systems are not notified of a need to compress data when the system is an SNA intermediate node.
- 1 *REQUEST - The remote systems are requested to compress data when the system is an SNA intermediate node.

Top

CL var for MAXINTSSN (5 0) (MAXINTSSN)

Specifies the name of the CL variable that receives the maximum number of intermediate sessions. Specify the name of a decimal variable with a minimum length of 5 digits and no decimal positions.

Top

CL var for RAR (5 0) (RAR)

Specifies the name of the CL variable that receives the route addition resistance. Specify the name of a decimal variable with a minimum length of 5 digits and no decimal positions.

Top

CL var for NETSERVER (85) (NETSERVER)

Specifies the name of the CL variable that receives the list of network node servers. Specify the name of a character variable with a minimum length of 85 characters. If the server name or network ID has fewer characters than the variable allows, the value is padded on the right with blanks. The list contains five node servers. Each server has the form: Network ID (9 characters) followed by the server name (8 characters). There are no separators. The network ID can contain the value *LCLNETID, which specifies that the current network ID is used. If there are less than five node servers specified, the remaining ones contain blanks for a name. As soon as the first blank name is encountered in the list, it is safe to assume that the remaining names are also blank.

Top

CL var for ALRSTS (10) (ALRSTS)

Specifies the name of the CL variable that receives the alert status. Specify the name of a character variable with a minimum length of 10 characters. (If the alert status value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

***ON** Alerts are created by the system for all alert conditions, except unattended conditions.

***OFF** Alerts are not created by the system.

***UNATTEND**

Alerts are created by the system for all alert conditions, including messages for which ***UNATTEND** is specified for the **Alert options (ALROPT)** parameter of the Add Message Description (ADDMSGD) or Change Message Description (CHGMSGD) command.

Top

CL var for ALRPRIFP (10) (ALRPRIFP)

Specifies the name of the CL variable that receives the alert primary focal point. Specify the name of the CL variable with a minimum length of 10 characters. (If the alert primary focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

***NO** The system is not an alert primary focal point.

***YES** The system is an alert primary focal point.

Top

CL var for ALRDFTFP (10) (ALRDFTFP)

Specifies the name of the CL variable that receives the value for the alert default focal point. Specify a CL variable with a minimum length of 10 characters. (If the alert default focal point value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

***NO** The system is not an alert default focal point.

***YES** The system is an alert default focal point.

Top

CL var for ALRLOGSTS (7) (ALRLOGSTS)

Specifies the name of the CL variable that receives the alert logging status. The variable must be a character variable with a minimum length of 7 characters.

The following values can be returned in the CL variable:

***NONE**

No alerts are logged.

***LOCAL**

Only locally created alerts are logged.

*RCV Only alerts received from other nodes are logged.

*ALL Both locally created alerts and alerts received from other nodes are logged.

Top

CL var for ALRBCKFP (16) (ALRBCKFP)

Specifies the name of the CL variable that receives the name of the system that provides alert focal point services if the primary focal point is unavailable. Specify the name of a character variable with a minimum length of 16 characters. (If the back up system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for ALRRQSFP (16) (ALRRQSFP)

Specifies the name of the CL variable that receives the name of the system that is requested to provide alert focal point services. Specify the name of a character variable with a minimum length of 16 characters. (If the requesting system name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for ALRCTLN (10) (ALRCTLN)

Specifies the name of the CL variable that receives the name of the controller through which alert messages are sent to another system when alert processing is active. Specify a CL variable with a minimum length of 10 characters. (If the alert controller name has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the variable:

***NONE**

There is no controller for alerts.

name Specifies the name of the controller being used for alerts in an alert controller session. This controller is ignored if the system has a primary or default alert focal point (if, for example, the node is in another system's sphere of control).

Top

CL var for ALRHLDCNT (5 0) (ALRHLDCNT)

Specifies the name of the CL variable that receives the maximum number of alerts that are created before the alerts are sent over the alert controller session (ALRCTLN network attribute). The alerts are held (queued) by the system until the specified number of alerts have been created. This parameter can be used to manage alerts that are sent over a limited resource by reducing the number of times alerts are sent.

Note: The ALRHLDCNT network attribute only applies when the ALRCTLN network attribute is used. When management services sessions, APPN, and sphere of control support are used, the ALRHLDCNT value is ignored.

The maximum number of alerts that can be created before the alerts are sent is 32,767. Specify the name of a decimal variable with a total length of 5 digits without decimal positions.

The following values can be returned in the variable:

-2 This value represents *NOMAX. The alerts are held indefinitely. The current alert hold count is the maximum value. The alerts can be sent at a later time by changing the ALRHLDCNT value to a lower value.

0-32767

Specifies the maximum number of alerts that can be created before being sent. Alerts have a "held" status until the maximum is reached. If the value 0 is specified, alerts are sent as soon as they are created.

Top

CL var for ALRFTR (10) (ALRFTR)

Specifies the name of the CL variable that receives the name of the active alert filter. Specify the name of a character variable with a minimum length of 10 characters. (If the alert filter name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for ALFTRLIB (10) (ALFTRLIB)

Specifies the name of the CL variable that receives the name of the library that contains the alert filter definition. Specify the name of a character variable with a minimum length of 10 characters. If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.

Top

CL var for MSGQ (10) (MSGQ)

Specifies the name of the CL variable that receives the system default network message queue name. Specify the name of a character variable with a minimum length of 10 characters. (If the message queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the system-default network message queue. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for OUTQ (10) (OUTQ)

Specifies the name of the CL variable that receives the system default network output queue name. Specify the name of a character variable with a minimum length of 10 characters. (If the output queue name has fewer characters than the variable allows, the value is padded on the right with blanks.)

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of the CL variable that receives the name of the library that contains the system-default network output queue. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.)

Top

CL var for JOBACN (10) (JOBACN)

Specifies the name of the CL variable that receives the current job action for job streams received through the network. Specify the variable with a minimum length of 10 characters. (If the job action value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

***REJECT**

The input stream is rejected by the system. This action allows you to secure your system from job streams received through the network.

***FILE** The input stream is filed in the queue of network files received by the user to whom it was sent. That user can then view, end, or receive the input stream, or submit the input stream to a job queue.

***SEARCH**

The table of network job entries is searched to determine the action taken for the input stream.

Top

CL var for MAXHOP (5 0) (MAXHOP)

Specifies the name of the CL variable that receives the maximum number of times in the SNADS network that a distribution queue originating at this node can be received and rerouted on the path to its final destination. Specify the name of a decimal variable with a total length of 5 digits, and no decimal positions.

Top

CL var for DDMACC (10) (DDMACC)

Specifies the name of the CL variable that receives the current system action for DDM or DRDA requests from other systems. Specify the CL variable with a minimum length of 10 characters. (If the DDM access value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

***REJECT**

This system does not allow DDM and DRDA requests from remote systems. However, this system can still use DDM or DRDA to access files or SQL tables on remote systems. Source (client) systems cannot access files or SQL tables on any other system that specifies *REJECT.

***OBJAUT**

If the user profile associated with the DDM or DRDA job is authorized to the files, all file or remote SQL requests are accepted. Object authorities such as read, write, or update must also exist for the files.

name Specifies the name of the customer validation program that can supplement object level security. This user-exit program can restrict user access to *PUBLIC and private files. The target DDM support calls the user program for each reference to a file. The user-exit program indicates to DDM if the request should proceed or end. In the case of DRDA use, the validation program is called only for connection requests, not for individual file access.

Top

CL var for DDMACCLIB (10) (DDMACCLIB)

Specifies the name of the CL variable that receives the name of the library that contains the DDM access program. Specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT or *OBJAUT is returned for the **CL var for DDMACC (10) (DDMACC)** parameter, the value for this parameter is all blanks.

Top

CL var for PCSACC (10) (PCSACC)

Specifies the name of the CL variable that receives the current system action for IBM System i Access for Windows requests. Specify a CL variable with a minimum length of 10 characters. (If the IBM System i Access for Windows access value has fewer characters than the variable allows, the value is padded on the right with blanks.)

The following values can be returned in the CL variable:

***REJECT**

The system does not allow any IBM System i Access for Windows requests.

***OBJAUT**

All IBM System i Access for Windows requests are allowed and controlled by the object authorizations on the system.

***REGFAC**

The registration facility is used to determine exit programs for the different servers. If no program is defined in the registration facility, *OBJAUT is used.

name The name of the customer supplied IBM System i Access for Windows host system application exit program that can supplement system object level security.

Top

CL var for PCSACCLIB (10) (PCSACCLIB)

Specifies the name of the CL variable that receives the name of the library that contains the IBM System i Access for Windows access program. You must specify the name of a character variable with a minimum length of 10 characters. (If the library name has fewer characters than the variable allows, the value is padded on the right with blanks.) If *REJECT, *REGFAC, or *OBJAUT is returned for the **CL var for PCSACC (10) (PCSACC)** parameter, the value for this parameter will be all blanks.

Top

CL var for DFTNETTYPE (10) (DFTNETTYPE)

Specifies the name of the CL variable that receives the system default value for the Integrated Services Digital Network (ISDN) network type. The operating system no longer uses this network attribute. Changes made to this network attribute have no effect. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for DFTCNLST (10) (DFTCNLST)

Specifies the name of the CL variable that receives the system default value for the ISDN connection list. The operating system no longer uses this network attribute. Changes made to this network attribute have no effect. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for ALWANYNET (10) (ALWANYNET)

Specifies the name of the CL variable that receives the network attribute that allows the Communications APIs to use other communication transports that are not native for that API. Examples include ICF over TCP/IP or Sockets over SNA. The variable must be a character variable with a minimum length of 10 characters.

[Top](#)

CL var for NWSDOMAIN (8) (NWSDOMAIN)

Specifies the name of the CL variable that receives the LAN Server domain to which all integrated PC Servers, also known as File Server Input/Output Processors(FSIOP), on the system belong. The variable must be a character variable with a minimum length of 8 characters.

[Top](#)

CL var for ALWVRTAPPN (10) (ALWVRTAPPN)

Specifies the name of the CL variable that receives the current setting for the virtual APPN support. The character variable must have a minimum length of 10 characters.

[Top](#)

CL var for ALWHPRTWR (10) (ALWHPRTWR)

Specifies the name of the CL variable that receives the current setting for the HPR tower transport support. The character variable must have a minimum length of 10 characters.

[Top](#)

CL var for VRTAUTODEV (5 0) (VRTAUTODEV)

Specifies the name of the CL variable that receives the current setting for the maximum amount of automatically created APPC devices allowed on a virtual controller. Specify the name of a decimal variable with a total length of 5 digits, and no decimal positions.

Top

CL var for HPRPTHMR (40) (HPRPTHMR)

Specifies the name of the CL variable that receives the current settings for the maximum amount of time in minutes for the HPR path switch timers. This field requires a 40 character variable, each 10 characters represents one of the four timer values in the order of network, high, medium and low priority.

Top

CL var for ALWADDCLU (10) (ALWADDCLU)

Specifies the name of the CL variable that receives the value that specifies whether this system will allow another system to add it as a node in a cluster. The character variable must have a minimum length of 10 characters.

The values that can be returned in the variable for allow add to cluster are:

***NONE**

No other system can add this system as a node in a cluster.

***ANY** Any other system can add this system as a node in a cluster.

***RQSAUT**

Any other system can add this system as a node in a cluster only after the cluster add request has been authenticated.

Top

CL var for MDMCNTRYID (2) (MDMCNTRYID)

Specifies the name of the CL variable that receives the network attribute for the country or region identifier associated with a modem. The character variable must have a minimum length of 2 characters.

MDMCNTRYID defines the country-specific or region-specific default characteristics for modems which are internal to system I/O adapters. This value must be configured correctly to insure proper operation and, in some countries or regions, meet legal requirements. The adapter will fail the vary on of the line if modem country or region ID is not set.

Top

Examples

Example 1: Retrieving Current System Name

```
DCL VAR(&SNAME) TYPE(*CHAR) LEN(8)
RTVNETA SYSNAME(&SNAME)
```

This command retrieves the current system name.

Example 2: Retrieving Virtual APPN Support, APPC Device Limits, and HPR Path Switch Timers

```
DCL  VAR(&ALWVRTAPPN) TYPE(*CHAR) LEN(10)
DCL  VAR(&VRTAUTODEV) TYPE(*DEC) LEN(5 0)
DCL  VAR(&HPRPTHMR) TYPE(*CHAR) LEN(40)
RTVNETA ALWVRTAPPN(&ALWVRTAPPN) VRTAUTODEV(&VRTAUTODEV)
        HPRPTHMR(&HPRPTHMR)
```

This command retrieves the current network attribute settings for allow virtual APPN support, automatically created APPC devices on a virtual controller, and the HPR path switch timers.

Top

Error messages

*ESCAPE Messages

CPF1844

Cannot access network attribute &1.

Top

Retrieve Object Description (RTVOBJD)

Where allowed to run: Compiled CL program or interpreted REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

Parameters
 Examples
 Error messages

The Retrieve Object Description (RTVOBJD) command retrieves the description of a specific object to a CL program or REXX procedure.

For parameters that are returned into CL variables by this command, the parameter descriptive title/prompt text lists the minimum length for the CL variable. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

1. You must have execute (*EXECUTE) authority for the library.
2. You must have some authority other than *EXCLUDE authority for the object. If the object is a file, you must have object operational (*OBJOPR) authority for the object.
3. You must have either all object (*ALLOBJ) or audit (*AUDIT) special authority to retrieve a value other than *NOTAVL for the OBJAUD parameter.

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, *AUTL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DOC, *DTAARA, *DTADCT, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FLR, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWSD, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *TIMZON, *USRIDX, *USRPRF, *USRQ, *USRSPC, *VLDL, *WSCST	Required, Positional 2
ASPDEV	ASP device	Single values: * <u> </u> , *ALLAVL, *CURASPGRP, *SYSBAS Other values: <i>Element list</i>	Optional
	Element 1: Device	<i>Name</i>	
	Element 2: Search type	* <u>ASP</u> , *ASPGRP	
RTNLIB	CL var for RTNLIB (10)	<i>Character value</i>	Optional
OBJATR	CL var for OBJATR (10)	<i>Character value</i>	Optional
USRDFNATR	CL var for USRDFNATR (10)	<i>Character value</i>	Optional

Keyword	Description	Choices	Notes
TEXT	CL var for TEXT (50)	Character value	Optional
OWNER	CL var for OWNER (10)	Character value	Optional
PGP	CL var for PGP (10)	Character value	Optional
ASP	CL var for ASP (2 0)	Decimal number	Optional
LIBASP	CL var for LIBASP (5 0)	Decimal number	Optional
OBJASPDEV	CL var for OBJASPDEV (10)	Character value	Optional
OBJASPGRP	CL var for OBJASPGRP (10)	Character value	Optional
LIBASPDEV	CL var for LIBASPDEV (10)	Character value	Optional
LIBASPGRP	CL var for LIBASPGRP (10)	Character value	Optional
OVFASP	CL var for OVFASP (1)	Character value	Optional
CRTDATE	CL var for CRTDATE (13)	Character value	Optional
CHGDATE	CL var for CHGDATE (13)	Character value	Optional
SAVDATE	CL var for SAVDATE (13)	Character value	Optional
SAVACTDATE	CL var for SAVACTDATE (13)	Character value	Optional
RSTDATE	CL var for RSTDATE (13)	Character value	Optional
CRTUSER	CL var for CRTUSER (10)	Character value	Optional
CRTSYSTEM	CL var for CRTSYSTEM (8)	Character value	Optional
OBJDMN	CL var for OBJDMN (2)	Character value	Optional
USEUPD	CL var for USEUPD (1)	Character value	Optional
USEDATE	CL var for USEDATE (7)	Character value	Optional
USECOUNT	CL var for USECOUNT (5 0)	Decimal number	Optional
RESETDATE	CL var for RESETDATE (7)	Character value	Optional
STG	CL var for STG (10)	Character value	Optional
CPR	CL var for CPR (1)	Character value	Optional
SIZE	CL var for SIZE (15 0)	Decimal number	Optional
SPCSIZE	CL var for SPCSIZE (15 0)	Decimal number	Optional
SPCALIGN	CL var for SPCALIGN (1)	Character value	Optional
SAVSIZE	CL var for SAVSIZE (15 0)	Decimal number	Optional
SAVCMD	CL var for SAVCMD (10)	Character value	Optional
SAVSEQNBR	CL var for SAVSEQNBR (4 0)	Decimal number	Optional
SAVLRGSEQ	CL var for SAVLRGSEQ (10 0)	Decimal number	Optional
SAVVOL	CL var for SAVVOL (71)	Character value	Optional
SAVDEV	CL var for SAVDEV (10)	Character value	Optional
SAVF	CL var for SAVF (10)	Character value	Optional
SAVFLIB	CL var for SAVFLIB (10)	Character value	Optional
SAVLABEL	CL var for SAVLABEL (17)	Character value	Optional
SRCF	CL var for SRCF (10)	Character value	Optional
SRCFLIB	CL var for SRCFLIB (10)	Character value	Optional
SRCMBR	CL var for SRCMBR (10)	Character value	Optional
SRCDATE	CL var for SRCDATE (13)	Character value	Optional
SYSLVL	CL var for SYSLVL (9)	Character value	Optional
COMPILER	CL var for COMPILER (16)	Character value	Optional
OBJLVL	CL var for OBJLVL (8)	Character value	Optional
ALWAPICHG	CL var for ALWAPICHG (1)	Character value	Optional

Keyword	Description	Choices	Notes
APICHG	CL var for APICHG (1)	Character value	Optional
USRCHG	CL var for USRCHG (1)	Character value	Optional
LICPGM	CL var for LICPGM (16)	Character value	Optional
PTF	CL var for PTF (10)	Character value	Optional
APAR	CL var for APAR (10)	Character value	Optional
OBJAUD	CL var for OBJAUD (10)	Character value	Optional
OBJSIG	CL var for OBJSIG (1)	Character value	Optional
SYSSIG	CL var for SYSSIG (1)	Character value	Optional
MLTSIG	CL var for MLTSIG (1)	Character value	Optional
JRNSTS	CL var for JRNSTS (1)	Character value	Optional
JRN	CL var for JRN (10)	Character value	Optional
JRNLIB	CL var for JRNLIB (10)	Character value	Optional
JRNIMG	CL var for JRNIMG (1)	Character value	Optional
JRNOMTE	CL var for JRNOMTE (1)	Character value	Optional
JRNSTRDATE	CL var for JRNSTRDATE (13)	Character value	Optional
STRJRNRCV	CL var for STRJRNRCV (10)	Character value	Optional
JRNRCVLIB	CL var for JRNRCVLIB (10)	Character value	Optional
RCVLIBASP	CL var for RCVLIBASP (10)	Character value	Optional
RCVLIBGRP	CL var for RCVLIBGRP (10)	Character value	Optional

Top

Object (OBJ)

Specifies the object for which you want to retrieve information.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object for which the description is to be retrieved.

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 type of the object for which you want to retrieve the information.

This is a required parameter.

object-type

Specify the type of the object for which information is to be retrieved.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device name where storage is allocated for the library containing the object. If the library resides in an ASP that is not part of the thread's library name space, this parameter must be specified to ensure the correct library is searched. If this parameter is used when the library qualifier specified for the **Object (OBJ)** parameter is *CURLIB or *LIBL, ASPDEV(*) is the only valid value. This parameter can be specified as a single value or a list of one or two elements.

Single values

*
_ 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 (ASP 2-32), and, if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.

*ALLAVL

All available ASPs will be searched. This includes the system ASP (ASP 1), all defined basic user ASPs (ASP 2-32), and all available primary and secondary ASPs (ASP 33-255 with a status of 'Available').

*CURASGRP

If the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group will be searched to find the library. The system ASP (ASP 1) and defined basic user ASPs (ASP 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 (ASP 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.

Element 1: Device

name

Specify the name of the primary or secondary ASP device to be searched. 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 (ASP 2-32) will not be searched.

Note: To specify a specific auxiliary storage pool (ASP) device name when the search type specified for element 2 is *ASP, you must have execute (*EXECUTE) authority for the specific ASP device.

To specify a specific auxiliary storage pool (ASP) device name when the search type specified for element 2 is *ASPGRP, you must have execute (*EXECUTE) authority for each ASP device in the ASP group.

Element 2: Search type

Specifies whether the single ASP device or the entire ASP group named in element 1 is to be searched.

***ASP** Only the single auxiliary storage pool (ASP) device named in element 1 is to be searched.

***ASPGRP**

The entire group of the primary auxiliary storage pool (ASP) device named in element 1 is to be searched.

Top

CL var for RTNLIB (10) (RTNLIB)

Specifies a 10-character variable used to return the name of the library that contains the object. If *LIBL or *CURLIB is specified for the library (qualifier 2) of the **Object (OBJ)** parameter, the value returned is the name of the library where the object was found. If a library name is specified, that name is returned by this parameter.

Top

CL var for OBJATR (10) (OBJATR)

Specifies a 10-character variable used to return an extended attribute of the object such as a program or file type. For example, the variable may be returned with PROD or CLP. No * will precede the value.

Top

CL var for USRDFNATR (10) (USRDFNATR)

Specifies 10-character variable used to return the user-defined attribute of the object. Blanks are returned if the retrieved object does not have a user-defined attribute.

Top

CL var for TEXT (50) (TEXT)

Specifies a 50-character CL variable used to return the text description of the object.

Top

CL var for OWNER (10) (OWNER)

Specifies a 10-character variable used to return the name of the owner of the object.

Top

CL var for PGP (10) (PGP)

Specifies a 10-character variable used to return the name of the user who is the primary group for the object. If there is no primary group for the object, this field contains a value of *NONE.

Top

CL var for ASP (2 0) (ASP)

Specifies a decimal (2 0) variable used to return the number of the auxiliary storage pool (ASP) number for the object. This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. The OBJASPDEV parameter should be used to return the ASP device name when the ASP number is larger than 32. The following values can be returned:

- 1 The object is in the system auxiliary storage pool.
- 2-32 The object is in a basic user auxiliary storage pool.
- 1 The object is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the OBJASPDEV parameter.

Top

CL var for LIBASP (5 0) (LIBASP)

Specifies a decimal (5 0) variable used to return the number of the auxiliary storage pool (ASP) where storage is allocated for the library that contains the object. This variable will contain an ASP number up to 32. If an ASP number is larger than 32, -1 is returned in this variable. The LIBASPDEV parameter should be used to return the ASP device name when the ASP number is larger than 32. The following values can be returned:

- 1 The library is in the system auxiliary storage pool.
- 2-32 The library is in a basic user auxiliary storage pool.
- 1 The library is in a primary or secondary auxiliary storage pool with an ASP number larger than 32. The ASP device name for the primary or secondary ASP can be retrieved with the LIBASPDEV parameter.

Top

CL var for OBJASPDEV (10) (OBJASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the object. The following special values can be returned:

- *N The name of the ASP device cannot be determined.
- *SYSBAS
The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

Top

CL var for OBJASPGRP (10) (OBJASPGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the object. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the OBJASPDEV parameter. The following special values can be returned:

- *N The name of the primary ASP device cannot be determined.
- *SYSBAS
The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for LIBASPDEV (10) (LIBASPDEV)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library that contains the object. The following special values can be returned:

*N The name of the ASP device cannot be determined.

*SYSBAS

The library is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for LIBASPGRP (10) (LIBASPGRP)

Specifies a 10-character variable used to return the name of the primary auxiliary storage pool (ASP) group where storage is allocated for the library that contains the object. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the LIBASPDEV parameter. The following special values can be returned:

*N The name of the primary ASP device cannot be determined.

*SYSBAS

The object is in the system ASP (ASP 1) or in a basic user ASP (ASPs 2-32).

CL var for OVFASP (1) (OVFASP)

Specifies a 1-character variable used to return the Object Overflowed auxiliary storage pool (ASP) flag.

'1' The object overflowed the ASP in which it resides.

'0' The object does not overflow the ASP. It is not possible for an object residing in the system ASP (ASP 1) or in a primary or secondary ASP (ASPs 33-255) to overflow its ASP, therefore, a value of '0' is always returned for objects in the system ASP (ASP 1) or in a primary or secondary ASP (ASPs 33-255).

CL var for CRTDATE (13) (CRTDATE)

Specifies a 13-character variable used to return the date and time the object was created. A value is returned in the form CYYMMDDHHMMSS where C = century; '0' indicates years 19xx and '1' indicates years 20xx; YY = year, MM = month, DD = day, HH = hour, MM = minutes and SS = seconds.

CL var for CHGDATE (13) (CHGDATE)

Specifies a 13-character variable used to return the date and time the object was last changed. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been changed.

CL var for SAVDATE (13) (SAVDATE)

Specifies a 13-character variable used to return the date and time the object was last saved. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been saved.

Top

CL var for SAVACTDATE (13) (SAVACTDATE)

Specifies a 13-character variable used to return the date and time the object was last saved during a save operation with the SAVACT parameter specified as something other than *NO. This date/time indicates when the object itself was saved; the SAVDAT parameter indicates when the save operation was started. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been saved or if SAVACT(*NO) was specified for the last save operation for the object.

Top

CL var for RSTDATE (13) (RSTDATE)

Specifies a 13-character variable used to return the date and time the object was last restored. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the object has not been restored.

Top

CL var for CRTUSER (10) (CRTUSER)

Specifies a 10-character variable used to return the name of the user that created the object.

Top

CL var for CRTSYSTEM (8) (CRTSYSTEM)

Specifies an 8-character variable used to return the name of the system on which the object was created.

Top

CL var for OBJDMN (2) (OBJDMN)

Specifies a 2-character variable used to return the object domain value for the object. The following values can be returned:

- *U The object is a user domain object.
- *S The object is a system domain object.

Top

CL var for USEUPD (1) (USEUPD)

Specifies a 1-character variable used to return whether the object usage information is updated for this object type. The following values can be returned:

- Y Object usage information is updated for this object type.
- N Object usage information is not updated for this object type. If 'N' is returned, the last used date for the object is blank and the number of days the object has been used is zero (0).

Top

CL var for USEDATE (7) (USEDATE)

Specifies a 7-character variable used to return the date the object was last used. The date is returned in the form CYYMMDD or is returned blank if the object does not have a last used date.

Top

CL var for USECOUNT (5 0) (USECOUNT)

Specifies a decimal (5 0) variable used to return the number of days the object has been used. If the object does not have a last used date, zero (0) is returned.

Top

CL var for RESETDATE (7) (RESETDATE)

Specifies a 7-character variable used to return the date the days used count was last reset to zero (0). The date is returned in the form CYYMMDD or is returned blank if the days used count has not been reset.

Top

CL var for STG (10) (STG)

Specifies a 10-character variable used to return the storage status of the object data. The following values can be returned:

*FREE The object data has been freed and the object is suspended.

*KEEP

The object data has not been freed and the object has not been suspended.

Top

CL var for CPR (1) (CPR)

Specifies a 1-character variable used to return the compression status of the object. The following values can be returned:

- Y The object is compressed.
- X The object is ineligible for compression.
- N The object is permanently decompressed.
- T The object is temporarily decompressed.

F The object is eligible for compression but is saved with storage freed.

Top

CL var for SIZE (15 0) (SIZE)

Specifies a decimal (15 0) variable used to return the size of the object in bytes. This value includes the value returned by the SPCSIZE parameter.

Top

CL var for SPCSIZE (15 0) (SPCSIZE)

Specifies a decimal (15 0) variable used to return the size of the primary associated space of object in bytes. If the object has no associated space, zero (0) is returned.

Top

CL var for SPCALIGN (1) (SPCALIGN)

Specifies a 1-character variable used to return whether the space associated with the object has been optimally aligned. Optimum alignment may allow for better performance of applications that use the object. The following values can be returned:

- '0' The space associated with the object has not been optimally aligned.
- '1' The space associated with the object has been optimally aligned.
- '2' There is not a space associated with the object.

Top

CL var for SAVSIZE (15 0) (SAVSIZE)

Specifies a decimal (15 0) variable used to return the size of the object in bytes at the time of the last save operation. If the object has not been saved, zero (0) is returned.

Top

CL var for SAVCMD (10) (SAVCMD)

Specifies a 10-character variable used to return the command used to save the object. If the object has not been saved, the variable is returned blank.

Top

CL var for SAVSEQNBR (4 0) (SAVSEQNBR)

Specifies a decimal (4 0) variable used to return the tape sequence number assigned when the object was saved on tape. This variable will contain a sequence number up to 9999. If the object has not been saved or was not saved to tape, zero (0) is returned. If a sequence number is actually greater than 9999, -5 is returned in this variable. The SAVLRGSEQ parameter should be used to return a sequence number which can be larger than 9999.

CL var for SAVLRGSEQ (10 0) (SAVLRGSEQ)

Specifies a decimal (10 0) variable used to return the tape sequence number (similar to the SAVSEQNBR parameter). This variable can contain a larger tape sequence number than the SAVSEQNBR parameter.

Top

CL var for SAVVOL (71) (SAVVOL)

Specifies a 71-character variable used to return the tape, diskette, or optical volumes used for saving the object. The variable returns a maximum of 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. Each volume ID entry is separated by a single character. If the object was saved in parallel format, the separator character contains a '2' before the first volume in the second media file, a '3' before the third media file, and so on, up to a '0' before the tenth media file. Otherwise, the separator characters are blank. If more than 10 volumes are used and the object was saved in serial format, '1' is returned in the 71st character of the variable. If the object was saved in parallel format, a '2' is returned in the 71st character of the variable. Otherwise, the 71st character is blank. If the object was last saved to a save file or was never saved, the variable is returned blank.

Top

CL var for SAVDEV (10) (SAVDEV)

Specifies a 10-character variable used to return the type of the device to which the object was last saved. The variable is returned with one of the following values, dependent on the device used for the last save operation:

- *SAVF for a save file
- *DKT for a diskette
- *TAP for a tape
- *OPT for an optical volume
- The variable is returned blank if the object was not saved.

Top

CL var for SAVF (10) (SAVF)

Specifies a 10-character variable used to return the name of the save file if the object was saved to a save file. If the object was not saved to a save file, the variable is returned blank.

Top

CL var for SAVFLIB (10) (SAVFLIB)

Specifies a 10-character variable used to return the name of the library that contains the save file if the object was saved to a save file. If the object is not saved to a save file, the variable is returned blank.

Top

CL var for SAVLABEL (17) (SAVLABEL)

Specifies a 17-character variable used to return the file label used when the object was saved. If the object is not saved to tape, to diskette, or to an optical volume, the variable is returned blank. The value returned corresponds to the value specified for the LABEL parameter on the command used to save the object.

[Top](#)

CL var for SRCF (10) (SRCF)

Specifies a 10-character variable used to return the name of the source file that was used to create the object. If no source file was used to create the object, the variable is returned blank. For ILE *PGM and *SRVPGM objects, the source file and member are blank. The source file information is stored with the *MODULE object.

[Top](#)

CL var for SRCFLIB (10) (SRCFLIB)

Specifies a 10-character variable used to return the name of the library that contains the source file that was used to create the object. If no source file is used to create the object, the variable is returned blank.

[Top](#)

CL var for SRCMBR (10) (SRCMBR)

Specifies a 10-character variable used to return the name of the member in the source file (SRCF parameter). If no source file is used to create the object, the variable is returned blank.

[Top](#)

CL var for SRCDATE (13) (SRCDATE)

Specifies a 13-character variable used to return the date and time the member in the source file was last updated. The variable is returned in the same format as the CRTDATE parameter or is returned blank if the member is not updated.

[Top](#)

CL var for SYSLVL (9) (SYSLVL)

Specifies a 9-character variable used to return the level of the operating system when the object was created. The variable is returned with a 3-character version level starting in character position 1, a 3-character release level starting in character position 4, and a 3-character modification level starting in character position 7. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'.

[Top](#)

CL var for COMPILER (16) (COMPILER)

Specifies a 16-character variable used to return the licensed program identifier, version level, release level, and modification level of the compiler. The variable is returned with the 7-character program identifier starting in character position 1, the 3-character version level in character position 8, the 3-character release level in character position 11, and the 3-character modification level in character position 14. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. If no compiler was used, the variable is returned blank.

Top

CL var for OBJLVL (8) (OBJLVL)

Specifies a 8-character variable used to return the object control level for the created object.

Top

CL var for ALWAPICHG (1) (ALWAPICHG)

Specifies a 1-character variable used to return the Allow Change by Program flag. The following values can be returned:

- '1' The object can be changed with the Change Object Description (QLICOBJD) API.
- '0' The object cannot be changed with the API.

Top

CL var for APICHG (1) (APICHG)

Specifies a 1-character variable used to return the Changed by Program flag. The following values can be returned:

- '1' The object has been modified with the Change Object Description (QLICOBJD) API.
- '0' The object has not been changed by the API.

Top

CL var for USRCHG (1) (USRCHG)

Specifies a 1-character variable used to return whether the object has been modified by the user. The following values can be returned:

- '1' The object has been modified by the user.
- '0' The object has not been modified by the user.

Top

CL var for LICPGM (16) (LICPGM)

Specifies a 16-character variable used to return the name, version level, release level, and modification level of the licensed program if the retrieved object is part of a licensed program. The variable is returned with the 7-character name starting in character position 1, the 3-character version level in character position 8, the 3-character release level in character position 11, and the 3-character modification level in character position 14. The first character of the version level is always the letter 'V'; the first character of the release level is always the letter 'R'; the first character of the modification level is always the letter 'M'. If the retrieved object is not part of a licensed program, the variable is returned blank.

Top

CL var for PTF (10) (PTF)

Specifies a 10-character variable used to return the Program Temporary Fix number that resulted in the creation of the retrieved object. For user-created objects, the variable is returned blank.

Top

CL var for APAR (10) (APAR)

Specifies a 10-character variable used to return the Authorized Program Analysis Report identification (APAR ID). The variable is returned with the APAR ID that caused this object to be patched. If the object has not been changed as a result of an APAR, the variable is returned blank.

This field is not updated when IBM-supplied Program Temporary Fixes are applied. The field is changed in the following situations:

- The default for a command is changed with the Change Command Default (CHGCMDDFT) CL command. The field is set to CHGDFT.
- The Change Object Description (QLICOBJD) API can change this field to any value.

Top

CL var for OBJAUD (10) (OBJAUD)

Specifies a 10-character variable used to return the auditing value of the object. A value of *NOTAVL will be returned if you do not have either all object (*ALLOBJ) or audit (*AUDIT) special authority. The values that can be returned include *NONE, *USRPRF, *CHANGE, *ALL, and *NOTAVL. See the **Object auditing value (OBJAUD)** parameter on the Change Object Audit (CHGOBJAUD) command for more information.

Top

CL var for OBJSIG (1) (OBJSIG)

Specifies a 1-character variable used to return whether the object has a digital signature. The following values can be returned:

- '1' The object has a digital signature.
- '0' The object does not have a digital signature.

Top

CL var for SYSSIG (1) (SYSSIG)

Specifies a 1-character variable used to return whether the object is signed by a source that is trusted by the system. The following values can be returned:

- '1' The object is signed by a source that is trusted by the system. If the object has multiple signatures, at least one of the signatures came from a source that is trusted by the system.
- '0' None of the object signatures came from a source that is trusted by the system.

Top

CL var for MLTSIG (1) (MLTSIG)

Specifies a 1-character variable used to return whether the object has more than one digital signature. The following values can be returned:

- '1' The object has more than one digital signature. If the SYSSIG parameter has a value of '1', at least one of the signatures is from a source trusted by the system.
- '0' The object has only one digital signature, or does not have a digital signature. Refer to the OBJSIG parameter to determine whether the object has a digital signature.

Top

CL var for JRNSTS (1) (JRNSTS)

Specifies a 1-character variable used to return the current journaling status of the object. The following values can be returned:

- '1' The object is currently being journaled.
- '0' The object is currently not being journaled.

Note: Other journal fields may contain data even though the object is not currently being journaled.

Top

CL var for JRN (10) (JRN)

Specifies a 10-character variable used to return the name of the current or last journal. If the object has never been journaled, the variable is returned blank.

Top

CL var for JRNLIB (10) (JRNLIB)

Specifies a 10-character variable used to return the name of the library that contains the journal. If the object has never been journaled, the variable is returned blank.

Top

CL var for JRNIMG (1) (JRNIMG)

Specifies a 1-character a variable used to return the journal image information. The following values can be returned dependent upon what images are generated for changes to the object:

- '0' Only *after* images are written to the journal for changes to the object.
- '1' Both *before* and *after* images are written to the journal for changes to the object.

If the object has never been journaled, the variable is returned blank.

Top

CL var for JRNOMTE (1) (JRNOMTE)

Specifies 1-character variable used to return information regarding journal entries to be omitted. The following values can be returned for the journal entries to be omitted:

- '1' *open* and *close* operations on the specified objects do not generate *open* and *close* journal entries.
- '0' No journal entries are omitted.

If the object has never been journaled, the variable is returned blank.

Top

CL var for JRNSTRDATE (13) (JRNSTRDATE)

Specifies a 13-character variable used to return the date and time journaling was last started. The value is returned in the same format as the CRTDATE parameter or is returned blank if the object has never been journaled.

Top

CL var for STRJRNRCV (10) (STRJRNRCV)

Specifies a 10-character variable used to return the name of the oldest journal receiver needed to successfully use the Apply Journaled Changes (APYJRNCHG) or Remove Journaled Changes (RMVJRNCHG) command. This field will be blank if the object has never been journaled.

For a file object, the journal receiver will contain the entry representing the start-of-the-save operation. However, if there are members within the file that contain partial transactions, then those members may require an earlier journal receiver. Use the Display File Description (DSPFD) command to determine the partial transaction state of the members of the file.

Top

CL var for JRNRCVLIB (10) (JRNRCVLIB)

Specifies a 10-character variable used to return the name of the library that contains the starting journal receiver. This field will be blank if the object has never been journaled.

Top

CL var for RCVLIBASP (10) (RCVLIBASP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) device where storage is allocated for the library that contains the starting journal receiver. This field will be blank if the object has never been journaled. The following special values can be returned:

*N The name of the ASP device cannot be determined.

***SYSBAS**

The library is in the system ASP (ASP 1) or in a basic user ASP (ASP's 2-32).

Top

CL var for RCVLIBGRP (10) (RCVLIBGRP)

Specifies a 10-character variable used to return the name of the auxiliary storage pool (ASP) group where storage is allocated for the library that contains the starting journal receiver. The ASP group name is the name of the primary ASP within the ASP group. The value returned may be the same as the value returned for the RCVLIBASP parameter. This field will be blank if the object has never been journaled. The following special values can be returned:

*N The name of the ASP device cannot be determined.

***SYSBAS**

The library is in the system ASP (ASP 1) or in a basic user ASP (ASP's 2-32).

Top

Examples

Jane Brown enters the following command to create a library:

```
CRTLIB LIB(PGMLIB) TYPE(*PROD)
      TEXT('Library for test programs')
```

Later, Jane, or anyone else with the proper authority, can retrieve the attributes of the library in a CL program as shown below:

```
DCL &CRTDATE *CHAR 13
DCL &OWN *CHAR 10
DCL &RTNLIB *CHAR 10
DCL &OBJASPDEV *CHAR 10
:
RTV OBJD OBJ(*LIBL/PGMLIB) OBJTYPE(*LIB) TEXT(&TEXT) +
      CRTDATE(&CRTDATE) OWNER(&OWN) +
      RTNLIB(&RTNLIB) OBJASPDEV(&OBJASPDEV)
```

The values returned in the variables of the CL program are shown below:

```
&TEXT = Library for test programs
&CRTDATE = 0900211130000
&OWN = JBROWN
&RTNLIB = QSYS
&OBJASPDEV = *SYSBAS
```

The value returned in the variable &CRTDATE indicates that PGMLIB was created on the eleventh day of February, 1990, at 1300 hours. The value returned in the variable &OWN indicates that the library was created by user profile JBROWN. The value returned in the variable &RTNLIB indicates that PGMLIB is in library QSYS. The value returned in the variable &OBJASPDEV indicates that storage for PGMLIB is allocated from *SYSBAS which includes the system auxiliary storage pool (ASP 1) and any defined basic user ASPs (ASP's 2-32).

Error messages

*ESCAPE Messages

CPFB8ED

Device description &1 not correct for operation.

CPF2115

Object &1 in &2 type *&3 damaged.

CPF2150

Object information function failed.

CPF2151

Operation failed for &2 in &1 type *&3.

CPF2173

Value for ASPDEV not valid with special value for library.

CPF218C

&1 not a primary or secondary ASP.

CPF218D

&1 not a primary ASP when *ASPGRP specified.

CPF2451

Message queue &1 is allocated to another job.

CPF3202

File &1 in library &2 in use.

CPF3203

Cannot allocate object for file &1 in &2.

CPF36F7

Message queue QSYSOPR is allocated to another job.

CPF980B

Object &1 in library &2 not available.

CPF9801

Object &2 in library &3 not found.

CPF9802

Not authorized to object &2 in &3.

CPF9803

Cannot allocate object &2 in library &3.

CPF9807

One or more libraries in library list deleted.

CPF9808

Cannot allocate one or more libraries on library list.

CPF9810

Library &1 not found.

CPF9811

Program &1 in library &2 not found.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9821

Not authorized to program &1 in library &2.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9830

Cannot assign library &1.

CPF9831

Cannot assign device &1.

CPF9833

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

Top

Retrieve PDG Profile (RTVPDGPRF)

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

Parameters
Examples
Error messages

The Retrieve Print Descriptor Group Profile (RTVPDGPRF) command is used in a CL program or REXX procedure to retrieve one or more of the print descriptor group profile values associated with a user profile. The values are returned in the specified CL variables for the desired user.

Restrictions:

1. The program must have *OBJOPR authority to the command.
2. The program must have *READ authority to the user's profile.

Top

Parameters

Keyword	Description	Choices	Notes
USER	User profile	Name, *CURRENT	Required, Positional 1
RTNUSER	CL var for RTNUSER	Character value	Optional
PDG	CL var for PDG	Character value	Optional
PDGLIB	CL var for PDGLIB	Character value	Optional
PRTD	CL var for PRTD	Character value	Optional

Top

User profile (USER)

Specifies the user profile name to be checked for its print descriptor group (PDG) profile.

The possible values are:

*CURRENT

Checks the user profile of the user of the current job.

user-name

Specify the user profile to check.

Top

CL var for RTNUSER (RTNUSER)

In control language (CL) programs, specifies the name of the 10-character variable used to get the name of the user profile for which information is requested.

Top

CL var for PDG (PDG)

In control language (CL) programs, specifies the name of the 10-character variable used to get the PDG profile of the user for which information is requested.

Top

CL var for PDGLIB (PDGLIB)

In control language (CL) programs, specifies the name of the 10-character variable used to get the library of the PDG profile of the user for which information is requested.

Top

CL var for PRTD (PRTD)

In control language (CL) programs, specifies the name of the 256-character variable used to get the print descriptor name from the PDG profile of the user for which information is requested.

Top

Examples

Assume a user with *OBJMGT authority entered the following command:

```
CHGUSRPRF  USER(JWONG)  PDG(*LIBL/TAXFORMS)  PRTD(FORM_C1)
```

Also assume the program with *OBJMGT authority contains the following commands and declarations:

```
DCL  VAR(&USER)      TYPE(*CHAR)  LEN(10)
DCL  VAR(&GROUP)     TYPE(*CHAR)  LEN(10)
DCL  VAR(&LIBRARY)   TYPE(*CHAR)  LEN(10)
DCL  VAR(&DESCRIPT)  TYPE(*CHAR)  LEN(256)
RTVPDGPRF  USER(JWONG)  RTNUSER(&USER)  PDG(&GROUP)  PDGLIB(&LIBRARY)
           PRTD(&DESCRIPT)
```

When the above program is called, the following values are returned:

```
&USER      'JWONG      '
&GROUP     'TAXFORMS  '
&LIBRARY   'TAXLIB    '
&DESCRIPT  'FORM_C1 ... '
```

Note: The value returned in variable &DESCRIPT is FORM_C1 followed by 249 blanks.

Top

Error messages

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2217

Not authorized to user profile &1.

CPF2247

Internal security object not available. Reason code &1.

Top

Retrieve Power Schedule Entry (RTVPWRSCDE)

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

Parameters
Examples
Error messages

The Retrieve Power On/Off Schedule Entry (RTVPWRSCDE) command retrieves a power on/off schedule entry value for use in a CL or REXX program. The value is returned (copied) to the specified CL variable in the program.

Restrictions:

1. To use this command, you must have use (*USE) authority to the Display Power On/Off Schedule (DSPPWRSCD) command.

Top

Parameters

Keyword	Description	Choices	Notes
DAY	Day	Date, * TODAY , *SUN, *MON, *TUE, *WED, *THU, *FRI, *SAT	Optional, Positional 1
PWRONTIME	CL var for PWRONTIME (6)	Character value	Optional
PWROFFTIME	CL var for PWROFFTIME (6)	Character value	Optional
DAYDESC	CL var for DAYDESC (38)	Character value	Optional
MSGITV	CL var for MSGITV (2 0)	Decimal number	Optional

Top

Day (DAY)

Specifies the day for which you are retrieving a power on/off schedule entry.

*TODAY

The current date's schedule entry is retrieved.

*SUN The schedule entry for Sunday is retrieved.

*MON

The schedule entry for Monday is retrieved.

*TUE The schedule entry for Tuesday is retrieved.

*WED The schedule entry for Wednesday is retrieved.

*THU The schedule entry for Thursday is retrieved.

*FRI The schedule entry for Friday is retrieved.

*SAT The schedule entry for Saturday is retrieved.

date Specify the date for which a schedule entry is to be retrieved. The date must be specified in the same format as specified by your job attributes.

Top

CL var for PWRONTIME (6) (PWRONTIME)

Specifies the name of the CL variable that receives the power on time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format **hhmmss**, where **hh** = **hours**, **mm** = **minutes**, and **ss** = **seconds**, is returned.

Top

CL var for PWROFFTIME (6) (PWROFFTIME)

Specifies the name of the CL variable that receives the power off time. The variable named has a minimum length of 6 characters. The special value *NONE or the time in the format **hhmmss**, where **hh** = **hours**, **mm** = **minutes**, and **ss** = **seconds**, is returned.

Top

CL var for DAYDESC (38) (DAYDESC)

Specifies the name of the CL variable that receives the day description value. The day description is an explanation of the power on/off schedule for that date. The variable named has a minimum length of 38 characters.

Top

CL var for MSGITV (2 0) (MSGITV)

Specifies the name of the CL variable that receives the message interval value. The message interval is the number of minutes before the scheduled power off that a message is sent to all work stations warning users of the power off. The variable named has a minimum length of 2 characters.

Top

Examples

Example 1: Retrieving Today's Schedule Entry

```
DCL VAR(&ONTIME) TYPE(*CHAR) LEN(6)
DCL VAR(&OFFTIME) TYPE(*CHAR) LEN(6)
RTVPWRSCDE DAY(*TODAY) PWRONTIME(&ONTIME) +
            PWROFFTIME(&OFFTIME)
```

This command retrieves the power on and off times for today.

Example 2: Retrieving Tuesday's Schedule Entry

```
DCL VAR(&ONTIME) TYPE(*CHAR) LEN(6)
DCL VAR(&OFFTIME) TYPE(*CHAR) LEN(6)
RTVPWRSCDE DAY(*TUE) PWRONTIME(&ONTIME) PWROFFTIME(&OFFTIME)
```

This command retrieves the power on and power off times for Tuesday's.

Error messages

*ESCAPE Messages

CPF1E2B

Power scheduler and cleanup options not found.

CPF1E23

Power schedule or cleanup options in use by another user.

CPF1E99

Unexpected error occurred.

CPF2105

Object &1 in &2 type *&3 not found.

CPF9808

Cannot allocate one or more libraries on library list.

Retrieve Query Management Form (RTVQMFORM)

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

Parameters
Examples
Error messages

The Retrieve Query Management Form (RTVQMFORM) command allows you to retrieve encoded form source records from a query management form (QMFORM) object. The source records are placed into a source file member that can be edited.

Form source can also be retrieved from a query definition (QRYDFN) object when the specified QMFORM does not exist.

Top

Parameters

Keyword	Description	Choices	Notes
QMFORM	Query management report form	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Query management report form	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *QMFORM</i>	Optional, Positional 3
ALWQRYDFN	Allow information from QRYDFN	<i>*NO, *YES, *ONLY</i>	Optional

Top

Query management report form (QMFORM)

Specifies the query management form object whose source is being retrieved.

This is a required parameter.

Qualifier 1: Query management report form

name Specify the name of the form being retrieved.

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 form. 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.

Source file (SRCFILE)

Specifies the previously-created source physical file into which the encoded form source records are to be written.

This is a required parameter.

Qualifier 1: Source file

name Specify the name of the source file that is to receive the form source.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source physical file member into which the encoded form source records are to be written. If a source file member name is not specified, then the form name specified on the QMFORM parameter is used.

If the member existed before running this command, it is cleared before any source statements are written into it. If the member does not exist, it is created.

***QMFORM**

The member name is the same as the form name specified on the QMFORM parameter.

name Specify the name of the member to receive the form source.

Top

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether form information is taken from a QRYDFN object when a query management form (QMFORM) object cannot be found using the specified object name. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

***NO** Information is not taken from a QRYDFN object.

***YES** Information is taken from a QRYDFN object when the specified QMQRY object is not found.

***ONLY**

Information is taken only from a QRYDFN. Query management objects are ignored.

Top

Examples

Example 1: Retrieving Encoded Form Source

```
RTVQMFORM QMFORM(RPTLIB/SALFORM) SRCFILE(FORMS)
          SRCMBR(EMPFORM)
```

This command retrieves the encoded form source from the form named SALFORM located in the RPTLIB library. The encoded form source records that are retrieved are placed into the newly created or cleared member EMPFORM in the first file named FORMS in the user's library list.

Example 2: Retrieving Source From Either the QMFORM or the QRYDFN

```
RTVQMFORM QMFORM(RPTLIB/SALFORM) SRCFILE(FORMS)
          SRCMBR(EMPFORM) ALWQRYDFN(*YES)
```

This command retrieves the encoded form source from the query management form (QMFORM) named SALFORM located in the RPTLIB library. If there is no QMFORM object named SALFORM in the RPTLIB library, then the form source is retrieved from the query definition (QRYDFN) named SALFORM in the RPTLIB library. The encoded form source records that are retrieved are placed into the first file named FORMS in the user's library list.

[Top](#)

Error messages

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2705

Source file &1 in &2 not available.

[Top](#)

Retrieve Query Mgmt Query (RTVQMQR)

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

Parameters
Examples
Error messages

The Retrieve Query Management Query (RTVQMQR) command allows you to retrieve Structured Query Language (SQL) source from a query management query (QMQR) object. The source records are placed into an editable source file member.

You can also retrieve query source from a query definition (QRYDFN) object when the specified QMQR does not exist.

Top

Parameters

Keyword	Description	Choices	Notes
QMQR	Query management query	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Query management query	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *QMQR</i>	Optional, Positional 3
ALWQRYDFN	Allow information from QRYDFN	<i>*NO, *YES, *ONLY</i>	Optional

Top

Query management query (QMQR)

Specifies the query management query whose source is to be retrieved.

This is a required parameter.

Qualifier 1: Query management query

name Specify the name of the query whose source is to be retrieved.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Source file (SRCFILE)

Specifies the previously-created source physical file into which the query source records are to be written.

This is a required parameter.

Qualifier 1: Source file

name Specify the name of the source file to receive the query source being retrieved.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member into which the query source records are being written. If the member existed before running this command, it is cleared before any source records are written to it. If the member does not exist, it is created.

***QMQRY**

The member name is the same as the query name specified on the QMQRY parameter.

name Specify the name of the file member to receive the query source.

Top

Allow information from QRYDFN (ALWQRYDFN)

Specifies whether query information is taken from a query definition (QRYDFN) object when a query management (QMORY) object cannot be found using the specified object name. Any information that has to be derived in this way is discarded when the command has completed processing. No query management object is created.

***NO** Information is not taken from a QRYDFN object.

***YES** Information is taken from a QRYDFN object when the specified QMORY object is not found.

***ONLY**

Information is taken only from a QRYDFN object. Query management objects are ignored.

Top

Examples

Example 1: Retrieving SQL Source

```
RTVQMORY QMORY(RPTLIB/SALQRY) SRCFILE(QRYS) SRCMBR(EMPQRY)
```

This command retrieves the source from the query named SALQRY located in the RPTLIB library. The source records that are retrieved are placed into the newly created or cleared member EMPQRY in the first file named QRYS in the user's library.

Example 2: Retrieving Source From Either the QMORY or the QRYDFN

```
RTVQMORY QMORY(RPTLIB/SALQRY) SRCFILE(QRYS) SRCMBR(EMPQRY)  
ALWQRYDFN(*YES)
```

This command retrieves the source from the query management query (QMORY) named SALQRY in the RPTLIB library. If there is no QMORY object named SALQRY in the RPTLIB library, then the query source is retrieved from the query definition (QRYDFN) named SALQRY in the RPTLIB library. The source records are placed into the newly created or cleared member EMPQRY in the first file named QRYS in the user's library list.

[Top](#)

Error messages

*ESCAPE Messages

QWM2701

&1 command failed.

QWM2703

&1 command ended.

QWM2705

Source file &1 in &2 not available.

[Top](#)

Retrieve S/36 Environment Attr (RTVS36A)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)
- Interactive REXX procedure (*IREXX)

Threadsafe: No

Parameters
Examples
Error messages

The Retrieve System/36 Attributes (RTVS36A) command allows the user to retrieve specific attribute information about the System/36 environment configuration and provide that information to a specified variable of a CL program or REXX procedure. Information about one or more attributes can be retrieved.

More information about the System/36 attributes that can be retrieved is in the help information for the Change System/36 Attributes (CHGS36A) command.

Top

Parameters

Keyword	Description	Choices	Notes
ENV	Environment name	<i>Name, #LIBRARY</i>	Optional, Key
SLIB	CL var for SLIB (8)	<i>Character value</i>	Optional
FLIB	CL var for FLIB (10)	<i>Character value</i>	Optional
LIBL	CL var for LIBL (4)	<i>Character value</i>	Optional
DATDIFF	CL var for DATDIFF (4)	<i>Character value</i>	Optional
S36ESHARE	CL var for S36ESHARE (4)	<i>Character value</i>	Optional
RCDBLK	CL var for RCDBLK (4)	<i>Character value</i>	Optional
CACHEDLTF	CL var for CACHEDLTF (4)	<i>Character value</i>	Optional
LPPAGE	CL var for LPPAGE (3)	<i>Character value</i>	Optional
FORMTYPE	CL var for FORMTYPE (4)	<i>Character value</i>	Optional
DFTMSGACN	CL var for DFTMSGACN (9)	<i>Character value</i>	Optional
HALTOPT	CL var for HALTOPT (4)	<i>Character value</i>	Optional
EVKJOBINIT	CL var for EVKJOBINIT (6)	<i>Character value</i>	Optional
EVKJOBPOL	CL var for EVKJOBPOL (8)	<i>Character value</i>	Optional
EVKJOBPTY	CL var for EVKJOBPTY (10)	<i>Character value</i>	Optional
SRCRCLEN	CL var for SRCRCLEN (3)	<i>Character value</i>	Optional
CHGACT	CL var for CHGACT (4)	<i>Character value</i>	Optional
ADDS36ONLY	CL var for ADDS36ONLY (4)	<i>Character value</i>	Optional
ICFSUBST	CL var for ICFSUBST (4)	<i>Character value</i>	Optional
MRTUSRPRF	CL var for MRTUSRPRF (8)	<i>Character value</i>	Optional
MRTAUT	CL var for MRTAUT (8)	<i>Character value</i>	Optional
MRTDLY	CL var for MRTDLY (5)	<i>Character value</i>	Optional
MRTJOBINIT	CL var for MRTJOBINIT (6)	<i>Character value</i>	Optional
MRTJOBPOL	CL var for MRTJOBPOL (8)	<i>Character value</i>	Optional
MRTJOBPTY	CL var for MRTJOBPTY (10)	<i>Character value</i>	Optional

Environment name (ENV)

Specifies the name of the System/36 environment from which you are retrieving attributes. The value is #LIBRARY and cannot be changed.

This is a required parameter.

Top

CL var for SLIB (8) (SLIB)

Specifies the name of an 8-character variable to receive the name of the default session library for jobs running in the System/36 environment.

Top

CL var for FLIB (10) (FLIB)

Specifies the name of a 10-character variable to receive the name of the default files library for jobs running in the System/36 environment.

Top

CL var for LIBL (4) (LIBL)

Specifies the name of a 4-character variable to receive information on whether the library list is used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

Top

CL var for DATDIFF (4) (DATDIFF)

Specifies the name of a 4-character variable to receive information on whether files with the same name but different dates can be used for jobs running in the System/36 environment. A value of *YES or *NO is returned in the variable.

Top

CL var for S36ESHARE (4) (S36ESHARE)

Specifies the name of a 4-character variable to receive information on whether programs share an open data path (ODP) to database files opened in the System/36 environment. A value of *YES or *NO is returned in the variable.

Top

CL var for RCDBLK (4) (RCDBLK)

Specifies the name of a 4-character variable to receive information on whether record blocking is used for sequential database files sharing an open data path in the System/36 environment. A value of *YES or *NO is returned in the variable.

Top

CL var for CACHEDLTF (4) (CACHEDLTF)

Specifies the name of a 4-character variable to receive information on whether deleted files are stored in a cache in the System/36 environment. A value of *YES or *NO is returned in the variable.

Top

CL var for LPPAGE (3) (LPPAGE)

Specifies the name of a 3-character variable to receive the number of lines printed on a page for jobs running in the System/36 environment. A value ranging from 1 through 112 is returned in the variable.

Top

CL var for FORMTYPE (4) (FORMTYPE)

Specifies the name of a 4-character variable to receive the form type of the printer form used when printing a job in the System/36 environment. A value of *STD or a user-defined form type is returned in the variable.

Top

CL var for DFTMSGACN (9) (DFTMSGACN)

Specifies the name of a 9-character variable to receive the default message action used by the System/36 environment when an error occurs during the running of a CL command within a System/36 environment procedure. A value of *CONTINUE, *HALT, IGNORE, or *CANCEL is returned in the variable.

Top

CL var for HALTOPT (4) (HALTOPT)

Specifies the name of a 4-character variable to receive the options list for continuation after an error occurs in the System/36 environment and *HALT is specified for the default message action.

Top

CL var for EVKJOBINIT (6) (EVKJOBINIT)

Specifies the name of a 6-character variable to receive the value for the method used to start EVOKE jobs or job steps in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

Top

CL var for EVKJOBPOL (8) (EVKJOBPOL)

Specifies the name of an 8-character variable to receive the value for the storage pool used for jobs started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

Top

CL var for EVKJOBPTY (10) (EVKJOBPTY)

Specifies the name of a 10-character variable to receive the value for the priority at which a job is started when it is started with the *IMMED option in the System/36 environment. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

Top

CL var for SRCRCLEN (3) (SRCRCLEN)

Specifies the name of a 3-character variable to receive the record length in bytes for System/36 source files QS36PRC and QS36SRC. A value ranging from 52 through 132 is returned in the variable.

Top

CL var for CHGACT (4) (CHGACT)

Specifies the name of a 4-character CL variable to receive the value indicating whether the configuration object can be updated using the Change System/36 (CHGS36) command while others are signed on to the System/36 environment. A value of *NO or *YES is returned in the variable.

Top

CL var for ADDS36ONLY (4) (ADDS36ONLY)

Specifies the name of a 4-character CL variable to receive the value indicating whether workstation devices can be added to the System/36 environment configuration when the device signs on to the System/36 environment. A value of *NO or *YES is returned in the variable.

Top

CL var for ICFSUBST (4) (ICFSUBST)

Specifies the name of a 4-character CL variable to receive the value indicating whether to scan intersystem communications function (ICF) start requests for substitution expressions. A value of *NO or *YES is returned in the variable.

Top

CL var for MRTUSRPRF (8) (MRTUSRPRF)

Specifies the name of an 8-character variable to receive the user profile under which the Multiple Requester Terminal (MRT) program is running. A value of *OWNER or *FRSTUSR is returned in the variable.

CL var for MRTAUT (8) (MRTAUT)

Specifies the name of an 8-character variable to receive the user authority to files used by the MRT program. A value of *ALLUSR or *FRSTUSR is returned in the variable.

Top

CL var for MRTDLY (5) (MRTDLY)

Specifies the name of a 5-character variable to receive the time in seconds that the system delays before ending the MRT program. A value ranging from 0 through 32767 is returned in the variable.

Top

CL var for MRTJOBINIT (6) (MRTJOBINIT)

Specifies the name of a 6-character variable to receive the value for the method used to start an MRT job in the System/36 environment. A value of *IMMED or *JOBQ is returned in the variable.

Top

CL var for MRTJOBPOL (8) (MRTJOBPOL)

Specifies the name of an 8-character variable to receive the value for the storage pool to be used for an MRT job started with the *IMMED option in the System/36 environment. A value of *BASE or *CURRENT is returned in the variable.

Top

CL var for MRTJOBPTY (10) (MRTJOBPTY)

Specifies the name of a 10-character variable to receive the the value for the priority to start an MRT job started with the *IMMED option. A value ranging from 1 through 99 or the value *SUBMITTER is returned in the variable.

Top

Examples

```
RTVS36A RCBBLK(&RBLOCK) HALT(&OPTION) MRTUSRPRF(&USERID)
```

This command retrieves the shared file record blocking value, the halt options list, and the user profile under which the MRT is running. The file record blocking value is copied into the CL variable &RBLOCK, which must be 4 characters in length. The halt options list is copied into the CL variable &OPTION, which must be at least 4 characters in length. The user profile under which the MRT is running is copied into the CL variable &USERID, which must be 8 characters in length.

Top

Error messages

None

[Top](#)

Retrieve Service Agent (RTVSRVAGT)

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

Parameters
Examples
Error messages

The Retrieve Service Agent (RTVSRVAGT) command allows a user to retrieve information about Service Agent.

Up to three valid days of the week for which Service Agent automatic processing can occur may be retrieved and placed in specified CL variable(s). The days of the week are valid for the system or logical partition on which the command is used.

Top

Parameters

Keyword	Description	Choices	Notes
DAY1	CL variable for day 1	<i>Character value</i>	Optional
DAY2	CL variable for day 2	<i>Character value</i>	Optional
DAY3	CL variable for day 3	<i>Character value</i>	Optional

Top

CL variable for day 1 (DAY1)

Specifies the name of the CL variable that receives the first day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

Top

CL variable for day 2 (DAY2)

Specifies the name of the CL variable that receives the second day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

Top

CL variable for day 3 (DAY3)

Specifies the name of the CL variable that receives the third day of the week for which Service Agent automatic processing can occur. The variable must be a character variable with a minimum length of 4 characters.

The returned day of the week is one of the following special values: *MON, *TUE, *WED, *THU, *FRI, *SAT, or *SUN.

Top

Examples

```
DCL &FIRSTDAY TYPE(*CHAR) LEN(4)
DCL &SECONDDAY TYPE(*CHAR) LEN(4)
DCL &THIRDDAY TYPE(*CHAR) LEN(4)
:
RTVSRVAGT DAY1(&FIRSTDAY) DAY2(*SECONDDAY) DAY3(&THIRDDAY)
```

This command retrieves the three valid days of the Service Agent automatic functions (Auto PTF and Auto test).

Top

Error messages

*ESCAPE Messages

CPF9899

Error occurred during processing of command.

Top

Retrieve Stop Word List Source (RTVSWLSRC)

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

Parameters
Examples
Error messages

The Retrieve Stop Word List Source (RTVSWLSRC) command is used to retrieve the words from an IBM-supplied or user-created stop word list into a source file.

Top

Parameters

Keyword	Description	Choices	Notes
LANGID	Language ID	<i>Character value</i>	Required, Positional 1
SRCFILE	Source file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *LANGID</i>	Optional
TYPE	Stop word list type	<i>*IBM, *USER</i>	Optional

Top

Language ID (LANGID)

Specifies the language identifier (ID) for the stop word list.

This is a required parameter.

Top

Source file (SRCFILE)

Specifies the qualified name of the source file used to receive the stop word list words. The contents of the source file are replaced.

source-file-name

Specify the name of the source file.

The possible library values are:

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

***CURLIB**

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

library-name

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

This is a required parameter.

Source member (SRCMBR)

Specifies the name of the source file member used to receive the stop word list words. The member is in the source file specified on the SRCFILE parameter.

*LANGID

The language ID is used as the source file member name.

source-file-member-name

Specify the name of the member in the source file used to receive the stop word list.

Stop word list type (TYPE)

Specifies the type of stop word list being retrieved

*IBM The stop word list is IBM-supplied.

*USER

The stop word list is user-created.

Examples

```
RTVSWLSRC  LANGID(ENG)  SRCFILE(MYLIB/MYFILE)
```

This command retrieves the stop word list into source file MYFILE in library MYLIB that has the language ID ENG.

Error messages

*ESCAPE Messages

CPF8723

Record length too small for database source file.

CPF8725

&1 type stop word list not supported for language.

CPF9899

Error occurred during processing of command.

Retrieve System Information (RTVSYSDNF)

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

Parameters
Examples
Error messages

The Retrieve System Information (RTVSYSDNF) command gathers key system information from your system and places it in the library specified for the Library (LIB) parameter.

Restrictions: To use this command, you must have *SAVSYS authority.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	<i>Name</i>	Required, Positional 1

Top

Library (LIB)

Specifies the library in which objects containing system information are to be stored. There is no default value, and the library must exist on the system.

library

Specify the library in which objects containing system information are to be stored. The library must exist on the system.

Top

Examples

None

Top

Error messages

*ESCAPE Messages

CPF2110

Library &1 not found.

CPF222E

&1 special authority is required.

CPFA95F

Errors occurred while retrieving system information.

Top

Retrieve System Value (RTVSYVAL)

Where allowed to run: Compiled CL program or interpreted
REXX (*BPGM *IPGM *BREXX *IREXX)
Threadsafe: Yes

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

The Retrieve System Value (RTVSYVAL) command is used in a CL program to retrieve the value from the specified system value so that it can be used in the program. The value is returned (copied) to the specified CL variable in the program.

Restrictions:

1. This command is valid only in CL programs or REXX procedures.
2. The attributes of the system value and the receiving CL variable must be compatible.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
SYSVAL	System value	QABNORMSW, QACGLVL, QACTJOB, QADLACTJ, QADLSPLA, QADLTOTJ, QALWJOBITP, QALWOBJRST, QALWUSRDMN, QASTLVL, QATNPGM, QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2, QAUTOCFG, QAUTORMT, QAUTOSPRPT, QAUTOVRT, QBASACTLVL, QBASPOOL, QBOOKPATH, QCCSID, QCENTURY, QCFGMSGQ, QCHRID, QCHRIDCTL, QCMNARB, QCMNRCYLMT, QCNTYID, QCONSOLE, QCRTAUT, QCRTOBJAUD, QCTLSBSD, QCURSYM, QDATE, QDATETIME, QDATFMT, QDATSEP, QDAY, QDAYOFWEEK, QDBFSTCCOL, QDBRCVYWT, QDECFMT, QDEVNAMING, QDEVRCYACN, QDSCJOBIV, QDSPSGNINE, QDYNPTYADJ, QDYNPTYSCD, QENDJOBMT, QFRCCVNRST, QHOUR, QHSTLOGSIZ, QIGC, QIGCCDEFNT, QIGCFNTSIZ, QINACTMSGQ, QINACTITV, QIPLDATTIM, QIPLSTS, QIPLTYPE, QJOBMSGQFL, QJOBMSGQMX, QJOBMSGQSZ, QJOBMSGQTL, QJOBSPLA, QKBDUF, QKBDTYPE, QLANGID, QLEAPADJ, QLIBLCKLVL, QLMTDEVSSN, QLMTSECOFR, QLOCALE, QLOGOUTPUT, QMAXACTLVL, QMAXJOB, QMAXSGNACN, QMAXSIGN, QMAXSPLF, QMCHPOOL, QMINUTE, QMLTHDACN, QMODEL, QMONTH, QPASTHRSVR, QPFRADJ, QPRBFTR, QPRBHLDTIV, QPRCFEAT, QPRCMLTTSK, QPRTDEV, QPRTKEYFMT, QPRTTXT, QPWDCHGBLK, QPWDEXPITV, QPWDEXPWRN, QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDLVL, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIE, QPWDRQDDGT, QPWDRQDDIF, QPWDRULES, QPWDVLDPGM, QPWRDWNLMT, QPWRSTIPL, QQRYDEGREE, QQRYTIMLMT, QRCLSPLSTG, QRETSVRSEC, QRMTIPL, QRMTSIGN, QRMTSRVATR, QSAVACCPH, QSCANFS, QSCANFCTL, QSCPFCNS, QSECOND, QSECURITY, QSETJOBATR, QSFWERRLOG, QSHRMEMCTL, QSPCENV, QSPLFACN, QSRLNBR, QSRTSEQ, QSRVDMP, QSSLSL, QSSLSLCTL, QSSLPCL, QSTGLOWACN, QSTGLOWLMT, QSTRPRTWTR, QSTRUPPGM, QSTMSG, QSVRAUTITV, QSYSLIBL, QTHDRSCADJ, QTHDRSCAFN, QTIMADJ, QTIME, QTIMSEP, QTIMZON, QTOTJOB, QTSEPOOL, QUPSDLYTIM, QUPSMMSGQ, QUSEADPAUT, QUSRLIBL, QUTCOFFSET, QVFOBJRST, QYEAR	Required, Positional 1
RTNVAR	CL variable for returned value	<i>Not restricted</i>	Required, Positional 2

Top

System value (SYSVAL)

Specifies the name of the system value whose value is retrieved and returned for use in the program. The names and descriptions of the system values that can be specified are in the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

The system values are:

QABNORMSW

Previous end of system indicator. This value cannot be changed.

- '0' means previous end was normal.
- '1' means previous end was abnormal.

QACGLVL

Accounting level. Changes made to this system value take effect for jobs started after the change is made.

- *NONE - No accounting information is written to a journal.
- *JOB - Job resource use is written to a journal.
- *PRINT - Spooled and printer file resource use is written to a journal.

QACTJOB

Initial number of active jobs for which storage is allocated. Changes made to this system value take effect at the next IPL.

QADLACTJ

Additional number of active jobs for which storage is allocated. Changes made to this system value take effect immediately.

QADLSPLA

Additional storage for extending spooling control block (bytes). The operating system no longer uses this system value. Changes made to this system value have no effect.

QADLTOTJ

Additional total number of jobs for which storage is allocated. Changes made to this system value take effect immediately.

QALWJOBITP

Allow jobs to be interrupted. This system value specifies how the system responds to user initiated requests to interrupt a job to run a user-defined exit program in that job. The Call Job Interrupt Program (QWCJBITP) API in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> contains information on using job interrupt exit programs. The Change Job Interrupt Status (QWCCJITP) API in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> contains information on retrieving and changing the interrupt status of a job. The interrupt status of an active job can be changed at any time but will only take effect when the value of QALWJOBITP allows jobs to be interrupted. Changes made to this system value take effect immediately. The shipped value is 0.

- 0 means the system will not allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be uninterruptible. All active jobs are uninterruptible regardless of what the job interrupt status is set to.
- 1 means the system will allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be uninterruptible.
- 2 means the system will allow jobs to be interrupted to run user-defined exit programs. All new jobs becoming active will default to be interruptible.

QALWBJRST

Allow object to be restored. This system value determines whether objects with security-sensitive attributes are restored. See Restore options for additional information.

QALWUSRDMN

Allow user domain objects in libraries or directories. This system value specifies which libraries on the system can contain the user domain user objects *USRSPC (user space), *USRIDX (user index), and *USRQ (user queue). Changes made to this system value take effect immediately.

QASTLVL

Assistance level. Indicates the Operational Assistant level of system displays for user profiles where ASTLVL(*SYSVAL) is specified. Changes made to this system value take effect immediately.

- *BASIC - The Operational Assistant user interface is used.
- *INTERMED - The system interface is used.
- *ADVANCED - The expert system interface is used.

If a command does not have an *ADVANCED level interface, *INTERMED is used.

QATNPGM

Attention program. If *ASSIST is specified for this system value, the Operational Assistant main menu is called when the user presses the Attention (Attn) key. This value can be changed to the name of a program, which will be called when the user presses the Attn key in a job where ATNPGM(*SYSVAL) is specified in the user profile. Changes made to this system value take effect immediately.

QAUDCTL

Audit control. This system value contains the on and off switches for object and user action auditing. This system value activates auditing on the system that is selected by the Change Object Auditing (CHGOBJAUD) and Change User Auditing (CHGUSRAUD) commands and the QAUDLVL and QAUDLVL2 system values. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *NONE - No security auditing is done on the system. This is the shipped value.
- *AUDLVL - The actions specified in the QAUDLVL and QAUDLVL2 system values will be logged to the security journal. Also actions specified by a user profile's action auditing values will be audited. A user profile's action auditing values are set through the AUDLVL parameter on the CHGUSRAUD command.
- *OBJAUD - Actions against objects that have an object audit value other than *NONE will be audited. An object's audit value is set through the Change Auditing Value (CHGAUD) command or the CHGOBJAUD command.
- *NOQTEMP - No auditing of most objects in QTEMP is done. You must specify *NOQTEMP with either *OBJAUD or *AUDLVL. You can not specify *NOQTEMP by itself.

QAUDENDACN

Audit journal error action. This system value specifies the action to be taken by the system if errors occur when an audit journal entry is being sent by the operating system to the security audit journal. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *NOTIFY - Notification of failure is sent to the QSYSOPR and QSYSMSG message queues, and then the action that caused the audit attempt continues.
- *PWRDWN SYS - The Power Down System (PWRDWN SYS) command is issued. The system will then be brought up in a restricted state on the following IPL, and then only a user with audit (*AUDIT) and all object (*ALLOBJ) special authority can sign on the system.

QAUDFRCLVL

Force audit journal. This system value specifies the number of audit journal entries that can be written to the security auditing journal before the journal entry data is forced to auxiliary storage.

- 1 through 100.
- *SYS - The system determines when the journal entries are to be written to auxiliary storage based on internal system processing. *SYS cannot be returned in a decimal variable, so the command returns 0 when the value *SYS is specified.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL). *NOTAVL cannot be returned in a decimal variable, so the command returns -1 in place of *NOTAVL.

Changes made to this system value take effect immediately.

QAUDLVL

Security auditing level. Controls the level of action auditing on the system. Changes made to this system value take effect immediately for all jobs running on the system.

- *NONE - No security action auditing will occur on the system. This is the shipped value.
- *AUDLVL2 - Both QAUDLVL and QAUDLVL2 system values will be used to determine the security actions to be audited.

Note:

- If you wish to use the QAUDLVL2 system value exclusively, set the QAUDLVL system value to *AUDLVL2 and add your auditing values to the QAUDLVL2 system value.
- If you wish to use both system values you can set your values in the QAUDLVL system value along with the *AUDLVL2 value, then add any additional values to the QAUDLVL2 system value.
- *ATNEVT - Attention events are audited. Attention events are conditions that require further evaluation to determine the condition's security significance. For example, intrusion monitor events need to be examined to determine whether the condition is an intrusion or a false positive.
- *AUTFAIL - Authorization failures are audited.
- *CREATE - All object creations are audited. Objects created into library QTEMP are not audited.
- *DELETE - All deletions of external objects on the system are audited. Objects deleted from library QTEMP are not audited.
- *JOBBAS - Base actions that affect a job are audited.
- *JOBCHGUSR - Actions that change a thread's active user profile or its group profiles are audited.
- *JOBDTA - Actions that affect a job are audited.

Note: *JOBDTA is composed of two values to allow you to better customize your auditing. If you specify both of the values, you will get the same auditing as if you specified *JOBDTA. The following values make up *JOBDTA.

- *JOBBAS
- *JOBCHGUSR
- *NETBAS - Network base functions are audited.
- *NETCLU - Cluster and cluster resource group operations are audited.
- *NETCMN - Networking and communications functions are audited.

Note: *NETCMN is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *NETCMN. The following values make up *NETCMN.

- *NETBAS
- *NETCLU
- *NETFAIL
- *NETSCK
- *NETFAIL - Network failures are audited.
- *NETSCK - Socket tasks are audited.
- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).

- *OBJMGT - Generic object tasks are audited.
- *OFCSRV - OfficeVision tasks are audited.
- *OPTICAL - All optical functions are audited.
- *PGMADP - Adopting authority from a program owner is audited.
- *PGMFAIL - Program failures are audited.
- *PRTDTA - Printing functions are audited.
- *SAVRST - Save and restore information is audited.
- *SECCFG - Security configuration is audited.
- *SECDIRSRV- Changes or updates when doing directory service functions are audited.
- *SECIPC - Changes to interprocess communications are audited.
- *SECNAS - Network authentication service actions are audited.
- *SECRUN - Security run time functions are audited.
- *SECSCCKD - Socket descriptors are audited.
- *SECURITY - All security-related functions are audited.

Note: *SECURITY is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *SECURITY. The following values make up *SECURITY.

- *SECCFG
- *SECDIRSRV
- *SECIPC
- *SECNAS
- *SECRUN
- *SECSCCKD
- *SECVFY
- *SECVLDL
- *SECVFY - Use of verification functions are audited.
- *SECVLDL - Changes to validation list objects are audited.
- *SERVICE - For a list of all the service commands and API calls that are audited, see the System i Security Reference, SC41-5302 publication
- *SPLFDTA - Spooled file functions are audited.
- *SYSMGT - System management tasks are audited.

QAUDLVL2

Security auditing level extension. This system value is required when more than sixteen auditing values are needed. Specifying *AUDLVL2 as one of the values in the QAUDLVL system value will cause the system to also look for auditing values in the QAUDLVL2 system value. Changes made to this system value take effect immediately for all jobs running on the system.

- *NONE - No auditing values are contained in this system value. This is the shipped value.
- *ATNEVT - Attention events are audited. Attention events are conditions that require further evaluation to determine the condition's security significance. For example, intrusion monitor events need to be examined to determine whether the condition is an intrusion or a false positive.
- *AUTFAIL - Authorization failures are audited.
- *CREATE - All object creations are audited. Objects created into library QTEMP are not audited.
- *DELETE - All deletions of external objects on the system are audited. Objects deleted from library QTEMP are not audited.
- *JOBBAS - Base actions that affect a job are audited.

- *JOBCHGUSR- Actions that change a thread's active user profile or its group profiles are audited.
- *JOBDDTA - Actions that affect a job are audited.

Note: *JOBDDTA is composed of two values to allow you to better customize your auditing. If you specify both of the values, you will get the same auditing as if you specified *JOBDDTA. The following values make up *JOBDDTA.

 - *JOBDBAS
 - *JOBCHGUSR
- *NETBAS - Network base functions are audited.
- *NETCLU - Cluster and cluster resource group operations are audited.
- *NETCMN - Networking and communications functions are audited.

Note: *NETCMN is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *NETCMN. The following values make up *NETCMN.

 - *NETBAS
 - *NETCLU
 - *NETFAIL
 - *NETSCK
- *NETFAIL - Network failures are audited.
- *NETSCK - Socket tasks are audited.
- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *OBJMGT - Generic object tasks are audited.
- *OFCSRV - OfficeVision tasks are audited.
- *OPTICAL - All optical functions are audited.
- *PGMADP - Adopting authority from a program owner is audited.
- *PGMFAIL - Program failures are audited.
- *PRTDDTA - Printing functions are audited.
- *SAVRST - Save and restore information is audited.
- *SECCFG - Security configuration is audited.
- *SECDIRSRV- Changes or updates when doing directory service functions are audited.
- *SECIPC - Changes to interprocess communications are audited.
- *SECNAS - Network authentication service actions are audited.
- *SECRUN - Security run time functions are audited.
- *SECSCKD - Socket descriptors are audited.
- *SECURITY - All security-related functions are audited.

Note: *SECURITY is composed of several values to allow you to better customize your auditing. If you specify all of the values, you will get the same auditing as if you specified *SECURITY. The following values make up *SECURITY.

 - *SECCFG
 - *SECDIRSRV
 - *SECIPC
 - *SECNAS
 - *SECRUN
 - *SECSCKD
 - *SECVFY

- *SECVLDL
- *SECVFY - Use of verification functions are audited.
- *SECVLDL - Changes to validation list objects are audited.
- *SERVICE - For a list of all the service commands and API calls that are audited, see the System i Security Reference, SC41-5302 publication
- *SPLFDTA - Spooled file functions are audited.
- *SYSMGT - System management tasks are audited.

QAUTOCFG

Automatic device configuration indicator. Changes made to this system value take effect immediately.

- 0 means auto-configuration is off.
- 1 means auto-configuration is on.

QAUTOSRPT

Automatic system disabled reporting. The operating system no longer uses this system value. Changes made to this system value have no effect.

QAUTORMT

Automatic configuration for remote controllers. The QAUTORMT system value controls the automatic configuration of remote controllers.

- 0 means auto-configuration is off.
- 1 means auto-configuration is on.

QAUTOVRT

Automatic virtual device configuration indicator. The user must have *ALLOBJ authority to change this system value. Changes made to this system value take effect immediately. See Autoconfigure virtual devices for additional information.

QBASACTLVL

Activity level of base storage pool. Changes made to this system value take effect immediately.

QBASPOOL

Minimum size of base storage pool (in Kilobytes). Changes made to this system value take effect immediately.

QBOOKPATH

Book and bookshelf search path. The operating system no longer uses this system value. Changes made to this system value have no effect.

QCCSID

Coded character set identifier. Changes made to this system value take effect for jobs started after the change is made.

QCENTURY

Century value for the system date.

- 0 indicated years 19XX.
- 1 indicates years 20XX.

QCFGMSGQ

Configuration message queue used to specify the message queue to receive communication messages. Both an object name and library name can be specified. A change to this system value takes effect when a line, controller, or device description that supports the MSGQ parameter is varied on.

QCHRID

Default graphic character set and code page used for displaying or printing data. Changes made to this system value take effect for display files, display device descriptions, and printer files that are created, changed, or overridden after the change.

QCHRIDCTL

Character identifier control for the job. This attribute controls the type of CCSID conversion that occurs for display files, printer files, and panel groups. The *CHRIDCTL special value must be specified for the CHRID parameter on the create, change, or override commands for display files, printer files, and panel groups before this attribute is used.

- 0 means the *DEVDS special value is used.
- 1 means the *JOBCCSID special value is used.

QCMNARB

Communication arbiters. The number of communication arbiter jobs that are available to process work for controllers and devices. A change to this value takes effect on the next IPL. The shipped value is *CALC.

- *CALC: The operating system calculates the number of communication arbiter jobs.
- 0 - 99: Specifies the number of communication arbiter jobs that are available to process work for controllers and devices.

Note: If this system value is set to zero (0), the work in these jobs is done in QSYSARB and QCLUS system jobs as opposed to the communication arbiters.

QCMNRCYLMT

Provides recovery limits for system communications recovery. Specifies the number of recovery attempts to make and when an inquiry message is sent to the device message queue or to the system operator when the specified number of recovery attempts have been reached. Changes made to this system value do not affect a currently varied on device, but is in effect when a device is varied on after the change.

QCNTYID

Default country or region identifier. Changes to this system value take effect for jobs started after the change is made.

QCONSOLE

System console. This value is not changeable.

QCRTAUT

Public authority for created objects. You must have *ALLOBJ and *SECADM special authorities to change this system value. Changes made to this system value take effect immediately.

- *CHANGE means the user can change the object and perform basic functions on the object. Change authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. Change authority provides object operational authority and all data authority.
- *ALL means the user can control the object's existence, specify the security for the object, change the object, change the owner for the object, and perform basic functions on the object. All authority allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management rights. If the object is an authorization list, the user cannot add, change, or remove users, or transfer ownership of the authorization list.
- *USE means the user can perform basic operations on the object, such as run a program or read a file. The user is prevented from changing the object. Use authority provides object operational authority and read authority.
- *EXCLUDE authority prevents the user from accessing the object.

QCRTOBJAUD

Create object auditing. This system value specifies the default object auditing value for an object created into a library or directory. The object auditing value determines whether an audit journal entry is sent to the system auditing journal when an object is used or changed. Changes made to this system value take effect immediately.

- *NOTAVL - The user performing the command is not allowed to display the current auditing value. You cannot change the system value to not available (*NOTAVL).
- *NONE - No auditing entries are sent for the object.
- *USRPRF - Auditing entries are sent if the user is currently being audited.
- *CHANGE - Auditing entries are sent if the object is changed.
- *ALL - Auditing entries are sent if the object is used or changed.

QCTLSBSD

Controlling subsystem description name. Both an object name and library name can be specified. Changes made to this system value take effect at the next IPL.

QCURSYM

Currency symbol. Changes made to this system value take effect immediately.

QDATE

System date. Changes made to this system value take effect immediately.

QDATETIME

System date and time. This is the date and time for the local system time as a single value. Retrieving or changing this value is similar to retrieving or changing QDATE and QTIME in a single operation. The format of the field is YYYYMMDDHHNNSSXXXXXX where YYYY is the year, MM is the month, DD is the day, HH is the hours, NN is the minutes, SS is the seconds, and XXXXXX is the microseconds. Changes made to this system value take effect immediately.

QDATFMT

Date format. Changes made to this system value take effect for jobs started after the change is made.

QDATSEP

Date separator. Changes made to this system value take effect for jobs started after the change is made.

QDAY Day of the month (day of the year if the system date format is Julian). Changes made to this system value take effect immediately.

QDAYOFWEEK

The day of the week.

- *SUN - Sunday
- *MON - Monday
- *TUE - Tuesday
- *WED - Wednesday
- *THU - Thursday
- *FRI - Friday
- *SAT - Saturday

QDBFSTCCOL

Database file statistics collection. Specifies the type of statistics collection requests that are allowed to be processed in the background by system job, QDBFSTCCOL. Changes made to this system value take effect immediately.

- *ALL means all user requested database file statistics collection requests and statistics collections automatically requested by the database manager are allowed to be processed by the database statistics system job.
- *SYSTEM means only automatically requested database statistics collection requests by the database manager are allowed to be processed by the database statistics system job.
- *USER means only user requested database file statistics collection requests are allowed to be processed by the database statistics system job.

- *NONE means no database file statistics collection requests are allowed to be processed by the database statistics system job.

QDBRCVYWT

Database recovery wait indicator. Changes to this system value take effect at the next IPL in unattended mode.

- 0 means do not wait.
- 1 means wait.

QDECFMT

Decimal format. Changes made to this system value take effect immediately.

QDEVNAMING

Indicates the device naming convention. Changes made to this system value take effect the next time a device is automatically configured. Existing configured device names are not changed.

- *NORMAL means follow System i standards.
- *S36 means follow S/36 standards.
- *DEVADR means device names are derived from resource names.

QDEVRCYACN

Specifies the action taken when an I/O error occurs for the job's requesting program device. Changes made to this system value take effect for jobs started after the change is made.

- *DSCMSG disconnects the job. On reconnection, an error message will be sent to the user's application program.
- *DSCENDRQS disconnects the job. On reconnection, a cancel request function should be performed to return control of the job back to the last request level.
- *ENDJOB ends the job. A job log will be produced for the job. A message will be sent to the job log and to the QHST log indicating that the job was ended because of device error.
- *ENDJOBNOLOG ends the job. A job log will not be produced for the job. A message will be sent to the QHST log indicating that the job was ended because of device error.
- *MSG signals the I/O error message to the application program. The application program performs error recovery itself.

QDSCJOBTV

Time interval that a job can be disconnected before it is ended. Changes made to this system value take effect immediately. An interactive job can be disconnected with the Disconnect Job (DSCJOB) command when it has been inactive for an interval of time (the system values QINACTIV and QINACTMSGQ), or when an Input/Output error occurs at the interactive job's work station (the system value QDEVRCYACN).

- 5-1440 is the time out interval in minutes.
- *NONE means no time out interval.

QDSPSGNINF

Controls the display of sign-on information. Changes made to this system value take effect immediately.

- 0 means the sign-on information is not displayed.
- 1 means the sign-on information is displayed.

QDYNPTYADJ

Dynamic priority adjustment. The QDYNPTYADJ system value controls whether the priority of interactive jobs is dynamically adjusted to maintain high performance of batch job processing. This adjustment capability is only effective on systems that are rated for both interactive and non-interactive throughput and have Dynamic Priority Scheduling enabled. A change to this value takes effect at the next IPL.

- 0 means the dynamic priority adjustment support is turned off.

- 1 means the dynamic priority adjustment support is turned on.

QDYNPTYSCD

Dynamic priority scheduler. The QDYNPTYSCD system value controls the dynamic priority scheduler algorithm. The value allows the use of dynamic priority scheduling.

- 0 means the dynamic priority scheduler is off.
- 1 means the dynamic priority scheduler is on.

QENDJOBLMT

Maximum time (in seconds) for application clean up during immediate ending of a job.

When a job being ended has a signal handling procedure for the asynchronous signal SIGTERM, the SIGTERM signal is generated for that job. When the signal handling procedure for the SIGTERM signal is given control, the procedure can take the appropriate actions to avoid undesirable results such as application data that has been partially updated. If the SIGTERM signal handler has not completed in the specified time, the system ends the job.

When the job is ended in a controlled manner, the maximum time for the SIGTERM signal handler is specified on the command. When the job is ended in an immediate manner, the maximum time for the SIGTERM signal handler is specified by this system value. This time limit is used when ending one job, when ending all the jobs in a subsystem, or when ending all jobs in all subsystems. After two minutes, the system operator can use the End Job (ENDJOB) command with OPTION(*IMMED) to override the QENDJOBLMT value and end individual jobs immediately.

A change to this value takes effect immediately. Jobs that are already ending are not affected.

QFRCCVNRST

Force conversion on restore. This system value allows you to specify whether or not to convert programs, service programs, SQL packages, and module objects during the restore. It can also prevent some objects from being restored. The default value on the restore commands use the value of this system value. Changes to this system value will take effect immediately.

- 0** Do not convert anything. Do not prevent anything from being restored.
- 1** Objects with validation errors will be converted.
- 2** Objects requiring conversion to be used on the current version of the operating system or on the current machine will be converted. Objects with validation errors will also be converted.
- 3** Objects suspected of having been tampered with, objects containing validation errors, and objects requiring conversion to be used by the current version of the operating system or on the current machine will be converted.
- 4** Objects that contain sufficient creation data to be converted and do not have valid digital signatures will be converted. An object that does not contain sufficient creation data will be restored without conversion. NOTE: Objects (signed and unsigned) that have validation errors, are suspected of having been tampered with, or require conversion to be used by the current version of the operating system, but cannot be converted will not be restored.
- 5** Objects that contain sufficient creation data will be converted. An object that does not contain sufficient creation data will be restored. NOTE: Objects that have validation errors, are suspected of having been tampered with, or require conversion to be used on the current version of the operating system, but cannot be converted will not be restored.
- 6** All objects that do not have a valid digital signature will be converted. NOTE: An object with a valid digital signature that also has a validation error, is suspected of having been tampered with, or requires conversion to be used on the current version of the operating system, but cannot be converted will not be restored.

7 Every object will be converted.

When an object is converted, its digital signature is discarded. The state of the converted object is set to user state. After conversion, objects will have a good validation value and are not suspected of having been tampered with.

QHOURL

Hour of the day. Changes made to this system value take effect immediately.

QHSTLOGSIZ

Maximum number of records for each version of the history log. Valid values range from 1 to 10,000,000 or the special value *DAILY which means that a new version of the history log is created each time the date in the history log messages changes, or when the current log version reaches the maximum size of 10,000,000 records. *DAILY cannot be returned in a decimal variable, so the Retrieve System Value (RTVSYVAL) command returns a value of -1 when the system value is set to *DAILY. Specifying a value of -1 on the Change System Value (CHGSYSVAL) command has the same effect as specifying *DAILY. Changes made to this system value take effect when the next version of the history log is created.

QIGC Indicates whether the double-byte character set (DBCS) version of the system is installed. This value cannot be changed.

- 0 means the DBCS version is not installed.
- 1 means the DBCS version is installed.

QIGCCDEFNT

Double byte character set (DBCS) coded font name. Used when transforming an SNA character string (SCS) into an Advanced Function Printing data stream (AFPDS) and when creating an AFPDS spooled file with shift in/shift out (SI/SO) characters in the data. Changes made to this system value take effect immediately.

QIGCFNTSIZ

Double byte coded font point size. Used along with the system value, QIGCCDEFNT, double byte coded font. They will be used when transforming SNA character string (SCS) into an Advanced Function Printing Data Stream (AFPDS) and when creating an AFPDS spooled file with shift in/ shift out (SI/SO) characters present in the data.

- *NONE means that no point size is identified to the system. The point size is selected by the system based on the type of printer used.
- 000.1 - 999.9 means the point size for the double byte coded font.

QINACTIV

Inactive interactive job time out interval in minutes. When the time interval is changed to a value other than *NONE a new inactivity interval is established and the analysis of job inactivity is started again. The system value QINACTMSGQ determines the action the system takes. For information on enforcement for target pass-through and TELNET sessions, see the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>

Local jobs that are currently signed on to a remote system are excluded. For example, a work station is directly attached to system A, and system A has QINACTIV set on. If Display Station Pass-through or TELNET is used to sign on to system B, this work station is not affected by the QINACTIV value set on system A. Changes made to this system value take effect immediately.

- *NONE means that the system does not check for inactivity.
- 5 - 300 means the number of minutes a job can be inactive before action is taken.

QINACTMSGQ

The qualified name of a message queue to which job inactive messages will be sent if QINACTMSGQ is not *NONE. The message queue must exist before the system value can be

changed to a message queue name. Both an object name and library name can be specified. Changes made to this system value take effect immediately.

- *ENDJOB means that interactive jobs, secondary jobs, and group jobs will be ended.
- *DSCJOB means that interactive jobs, secondary jobs, and group jobs will be disconnected.
- Message queue name is the name of a message queue that receives a message when a job has been inactive.

QIPLDATTIM

Date and time for automatic IPL. This system value can be set independently in each partition. If the primary partition is powered down at the time an automatic IPL should occur in a secondary partition, the IPL will not occur. When the primary partition does IPL, the secondary partition will be IPLed if its IPL date and time is past due. The secondary partition will not IPL if it was configured with an IPL action of hold. Changes made to this system value take effect immediately.

QIPLDATTIM is a single system value with two parts:

- Date: The date an IPL automatically occurs on the system. The date is specified in QDATFMT format with no date separators.
- Time: The time an IPL automatically occurs on the system. The time is specified with no time separators.

*NONE, which indicates that no timed automatic IPL is scheduled, can be specified instead of a specific date and time.

The following example shows how to change the IPL date and time to September 10, 1993 (QDATFMT is MDY) at 9:00 a.m.

```
CHGSYSVAL SYSVAL(QIPLDATTIM) VALUE('091093 090000')
```

QIPLSTS

Initial program load (IPL) status indicator.

- 0 means operator panel IPL.
- 1 means auto-IPL after power restored.
- 2 means restart IPL.
- 3 means time of day IPL.
- 4 means remote IPL.

QIPLTYPE

Indicates the type of IPL to perform. Changes made to this system value take effect at the next manual IPL.

- 0 means unattended IPL.
- 1 means attended IPL with dedicated service tools.
- 2 means attended IPL with console in debug mode.

Note: You should only use this for problem analysis because it prevents other devices on the work station controller from being used.

QJOBMSGQFL

Job message queue full action. This system value specifies how to handle the job message queue when it is considered full. Changes made to this system value take effect for jobs started after the change is made.

- *NOWRAP - The job message queue is not wrapped.
- *WRAP - The job message queue is wrapped.
- *PRTWRAP - The job message queue is wrapped and the messages that are being overlaid are printed.

QJOBMSGQMX

Job message queue maximum size. This system value specifies how large (in megabytes) a message queue can be before it is considered full. Changes made to this system value take effect for jobs started after the change is made.

QJOBMSGQSZ

Initial size of job message queue in kilobytes (KB). The operating system no longer uses this system value. Changes made to this system value have no effect.

QJOBMSGQTL

Maximum size of job message queue (in KB). The operating system no longer uses this system value. Changes made to this system value have no effect.

QJOBSPLA

Initial size of spooling control block for a job (in bytes). Changes made to this system value take effect when a cold start is requested during the installation of the operating system licensed program.

QKBDDBUF

Keyboard buffer. Changes made to this system value take effect the next time someone logs on.

- *NO means turn off the type-ahead feature and the attention key buffering option.
- *TYPEAHEAD means turn on the type-ahead feature but turn off the attention key buffering option.
- *YES means turn on the type-ahead feature and the attention key buffering option.

QKBDTYPE

Keyboard language character set. Changes made to this system value take effect immediately.

QLANGID

Default language identifier. Changes to this system value take effect for jobs started after the change is made.

QLEAPADJ

Leap year adjustment. This system value is used to adjust the system calendar algorithm for the leap year in different calendar systems.

This system value is determined by the year offset that is associated with the time zone description specified in the system value QTIMZON. A change to a different time zone description for QTIMZON may result in a different associated adjustment.

QLEAPADJ cannot be changed to a value that is different than its current value. If an attempt is made to do so, the diagnostic message CPD168B will be issued. The value of QLEAPADJ is managed by the system.

QLIBLCKLVL

Library locking level. Specifies whether libraries in a job's library search list are locked by that job. A change to this system value takes effect for all jobs that become active after the change.

- 0 means the libraries in a user job's library search list are not locked.
- 1 means the libraries in a user job's library search list are locked by that job.

QLMTDEVSSN

Limits concurrent device sessions. Changes made to this system value take effect immediately.

- 0 means users are not limited to a specific number of device sessions.
- 1-9 indicates maximum number of concurrent device sessions.

QLMTSECOFR

Limit security officer device access. Changes made to this system value take effect immediately.

- 0 means users with *ALLOBJ or *SERVICE special authority can sign on any work station.
- 1 means users with *ALLOBJ or *SERVICE special authority must have explicit authority to a work station.

QLOCALE

Locale path name. This system value is used to set the locale for the system. The locale path name must be a path name that specifies a locale. A locale is made up of the language, territory, and code set combination used to identify a set of language conventions. The maximum path length allowed for the locale path name on the Change System Value (CHGSYSVAL) command is 1,024 bytes.

A change to this system value takes effect immediately. The shipped value may be different for different countries.

- *NONE means there is no locale path name for the QLOCALE system value.
- *C means the C locale is to be used.
- *POSIX means the POSIX locale is to be used.

QLOGOUTPUT

Job log output. This system value specifies how the job log will be produced when a job completes. This does not affect job logs produced when the message queue is full and the job message queue full action specifies *PRTWRAP. Messages in the job message queue are written to a spooled file, from which the job log can be printed, unless the Control Job Log Output (QMHCTLJL) API was used in the job to specify that the messages in the job log are to be written to a database file.

Changes made to this system value take effect immediately for jobs entering the system after the change is made.

- *JOBEND means the job log will be produced by the job itself. If the job cannot produce its own job log, the job log will be produced by a job log server.
- *JOBLOGSVR means the job log will be produced by a job log server.
- *PND means the job log will not be produced. The job log remains pending until removed.

QMAXACTLVL

Maximum activity level of the system. Changes made to this system value take effect immediately.

QMAXJOB

Maximum number of jobs that are allowed on the system. Changes made to this system value take effect immediately.

QMAXSGNACN

The system's response when the limit imposed by QMAXSIGN system value is reached. Changes made to this system value take effect the next time someone attempts to sign on the system.

- 1 means the device will be disabled.
- 2 means the user profile will be disabled.
- 3 means the device and the user profile will be disabled.

QMAXSIGN

Maximum number of not valid sign-on attempts allowed. Changes made to this system value take effect the next time someone attempts to sign on the system.

QMAXSPLF

Maximum number of spooled files that can be created per job. Changes made to this system value take effect immediately. Spooled files will not be deleted when this value is changed to a lower number. See the Printer Device Programming book for information on how this system value affects spooling for a job.

QMCHPOOL

Machine storage pool size (in KB). Changes made to this system value take effect immediately.

Note: Changes to the size of a pool may require pages to be written to auxiliary storage. The time required for the system to complete a large change may be greater than your default wait time. If this occurs, message CPF1001 (Wait time expired for system response.) is issued, even though the change completes.

QMINUTE

Minute of the hour. Changes made to this system value take effect immediately.

QMLTTHDACN

Multithreaded job action. This value controls the action to be taken when a function that may not be threadsafe is called in a multithreaded job. Changes made to this system value take effect immediately. The shipped value is 2.

- 1 means perform the function that is not threadsafe without sending a message.
- 2 means perform the function that is not threadsafe and send an informational message.
- 3 means do not perform the function that is not threadsafe.

QMODEL

System model number. The number or letters used to identify the model of the system. You cannot change QMODEL, but the 4-character value can be displayed or retrieved in user-written programs. The system model number system value is the same in each partition on a system.

QMONTH

Month of the year (not used for Julian dates). Changes made to this system value take effect immediately.

QPASTHRSVR

Pass-through servers. The number of target display station pass-through server jobs that are available to process display station pass-through, IBM System i Access for Windows workstation function (WSF), and other 5250 emulation programs on programmable workstations. Changes made to this system value take effect immediately. The shipped value is *CALC.

QPFRAJ

Initial program load (IPL) performance adjustment and dynamic performance tuning. Dynamic performance tuning automatically changes storage pool sizes and activity levels for shared storage pools. Private storage pools are not changed. Changes made to this system value take effect immediately.

- 0 means no performance adjustment. Dynamic performance tuning is not started.
- 1 means performance adjustment at IPL. Dynamic performance tuning is not started.
- 2 means performance adjustment at IPL. Dynamic performance tuning is started. If QPFRAJ is changed from 2 to 0 or 1, dynamic performance tuning is stopped.
- 3 means dynamic performance tuning is started. If QPFRAJ is changed from 3 to 0 or 1, dynamic performance tuning is stopped.

If you create journal QPFRAJ in library QSYS, the dynamic tuning program keeps a record of the changes made to storage pool sizes, activity levels, and the performance level of the system when the changes were made (faulting rates per pool, pool sizes, and activity levels).

QPRBFTR

Problem filter name. Specifies the name of the filter object used by the service activity manager when processing problems. Changes to this system value take effect immediately.

QPRBHLDTV

Problem log entry hold interval. Changes made to this system value take effect immediately.

QPRCFEAT

Processor feature. This is the processor feature code level of the system. You cannot change QPRCFEAT, but the 4-character value can be displayed or retrieved in user-written programs. The processor feature system value is the same in each partition on a system.

QPRCMLTTSK

Processor multitasking. If the hardware on your system supports processor multitasking, this system value allows you to set the multitasking capability to be on, off, or System-controlled. Changes to this system value can affect the performance of your system.

An IPL may be required for this system value to take effect. See the System Values topic in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for additional information.

- 0 means that processor multitasking is turned off.
- 1 means that processor multitasking is turned on.
- 2 means that processor multitasking is under system control.

On some partitioned systems, this system value can only be changed from the primary partition. For more information on partitions, see the Logical Partitions topic in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

QPRTDEV

Default printer device description. Changes made to this system value take effect for jobs started after the change is made.

QPRTKEYFMT

Print key format. Changes made to this system value take effect for jobs started after the change is made.

- *PRTHDR means that header information is printed when the print key is pressed.
- *PRTBDR means that border information is printed when the print key is pressed.
- *PRTALL means that border information and header information are printed when the print key is pressed.
- *NONE means that border information and header information are not printed when the print key is pressed.

QPRTTXT

Up to 30 characters of text that can be printed at the bottom of listings and separator pages. Changes made to this system value take effect for jobs started after the change is made.

QPWDCHGBLK

Specifies the time period during which a password is blocked from being changed following the prior successful password change operation. This system value does not restrict password changes made by the Change User Profile (CHGUSRPRF) command. Changes made to this system value take effect immediately.

- *NONE means there is no restriction on how frequently a user can change a password.
- 1-99 indicates the number of hours a user must wait after the prior successful password change operation before they can change the password again.

QPWDEXPITV

The number of days for which a password is valid. Changes made to this system value take effect immediately.

- *NOMAX means a password can be used an unlimited number of days.
- 1-366 means the number of days before the password ends.

QPWDEXPWRN

Controls the number of days prior to a password expiring to begin displaying password expiration warning messages on the Sign-on Information display.

- 1-99 indicates the number of days prior to the password expiring to begin displaying the password expiration warning message.

QPWDLMTAJC

Limits the use of adjacent numbers in a password. Changes made to this system value take effect the next time a password is changed.

- 0 means adjacent numbers are allowed.
- 1 means adjacent numbers are not allowed.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDLMTCHR

Limits the use of certain characters in a password. Changes made to this system value take effect the next time a password is changed.

- *NONE means there are no restricted characters.
- restricted-characters means up to 10 restricted characters enclosed in apostrophes can be specified. Valid characters are: A-Z, 0-9, and special characters #, \$, @, or underscore (_).

Note: This system value is ignored if the system is operating at QPWDLVL 2 or 3.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDLMTREP

Limits the use of repeating characters in a password. Changes made to this system value take effect the next time a password is changed.

- 0 means characters can be used more than once.
- 1 means characters cannot be used more than once.
- 2 means characters cannot be used consecutively.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDLVL

Specifies the password level.

Changing this system value requires careful consideration. If your system connects to other systems in a network then all systems must be able to run with the password rules that will be in effect.

See the System i Security Reference, SC41-5302 publication for additional considerations prior to changing this system value.

Changes to this system value will take effect on the next IPL.

- 0 means passwords from 1-10 characters are allowed.
- 1 means passwords from 1-10 characters are allowed. i5/OS NetServer passwords for Windows 95/98/ME clients will be removed from the system making the product unavailable for use.
- 2 means passwords from 1-128 characters are allowed. Passwords can consist of any character and will be case sensitive.
- 3 means passwords from 1-128 characters are allowed. Passwords can consist of any character and will be case sensitive. i5/OS NetServer passwords for Windows 95/98/ME clients will be removed from the system making the product unavailable for use.

QPWDMAXLEN

The maximum number of characters in a password. Changes made to this system value take effect the next time a password is changed.

- 1-128 means a value from 1 to 128 can be specified as the maximum number of characters in a password.

If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDMINLEN

The minimum number of characters in a password. Changes made to this system value take effect the next time a password is changed.

- 1-128 means a value from 1 to 128 can be specified as the minimum number of characters in a password.

If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDDIF

Controls the position of characters in a new password. Changes made to this system value take effect the next time a password is changed.

- 0 means the same characters can be used in a position corresponding to the same position in the previous password.
- 1 means the same character cannot be used in a position corresponding to the same position in the previous password.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDRQDDGT

Require number in a new password. Changes made to this system value take effect the next time a password is changed.

- 0 means numbers are not required.
- 1 means one or more numbers are required.

Note: If the QPWDRULES system value specifies any value other than *PWDSYSVAL, this system value will be ignored when new passwords are checked to see if they are formed correctly. In addition, this system value cannot be changed if QPWDRULES has a value other than *PWDSYSVAL.

QPWDRQDDIF

Controls whether the password must be different than the previous passwords. Changes made to this system value take effect the next time a password is changed.

- 0 means a password can be the same as one previously used.
- 1 means a password must be different than the previous 32 passwords.
- 2 means a password must be different than the previous 24 passwords.
- 3 means a password must be different than the previous 18 passwords.

- 4 means a password must be different than the previous 12 passwords.
- 5 means a password must be different than the previous 10 passwords.
- 6 means a password must be different than the previous 8 passwords.
- 7 means a password must be different than the previous 6 passwords.
- 8 means a password must be different than the previous 4 passwords.

QPWDRULES

Specifies the rules used to check whether a password is formed correctly. Changes made to this system value take effect the next time a password is changed.

- *PWDSYSVAL - This system value is ignored and the other password system values are used to check whether a password is formed correctly. Specifically, the QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, and QPWDRQDDGT system values will be used instead of QPWDRULES.

Note: If any value other than *PWDSYSVAL is specified for QPWDRULES, the QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, and QPWDRQDDGT system values are ignored when a new password is checked to see if it is formed correctly.

- *CHRLMTAJC - The password may not contain 2 or more occurrences of the same character that are positioned adjacent (consecutive) to each other. This value cannot be specified if the *CHRLMTREP value is also specified.
- *CHRLMTREP - The password may not contain 2 or more occurrences of the same character. This value cannot be specified if the *CHRLMTAJC value is also specified.
- *DGTLMTAJC - The password may not contain 2 or more adjacent (consecutive) digit characters.
- *DGTLMTFST - The first character of the password may not be a digit character. This value cannot be specified if *LTRLMTFST and *SPCCHRLMTFST values are also specified.
- *DGTLMTLST - The last character of the password may not be a digit character. This value cannot be specified if *LTRLMTLST and *SPCCHRLMTLST values are also specified.
- *DGTMAXn - Where n is a number from 0 to 9. Specifies the maximum number of digit characters that may occur in the password.

Only one *DGTMAXn value can be specified. If a *DGTMINn value is also specified, the n value specified for *DGTMAXn must be greater than or equal to the n value specified for *DGTMINn.

- *DGTMINn - Where n is a number from 0 to 9. Specifies the minimum number of digit characters that must occur in the password.

Only one *DGTMINn value can be specified. If a *DGTMAXn value is also specified, the n value specified for *DGTMAXn must be greater than or equal to the n value specified for *DGTMINn.

- *LMTSAMPOS - The same character cannot be used in a position corresponding to the same position in the previous password.
- *LMTPRFNAME - The uppercase password value may not contain the complete user profile name in consecutive positions.
- *LTRLMTAJC - The password may not contain 2 or more adjacent (consecutive) letter characters.
- *LTRLMTFST - The first character of the password may not be a letter character. This value cannot be specified if *DGTLMTFST and *SPCCHRLMTFST values are also specified. If the system is operating with a QPWDLVL of 0 or 1, *LTRLMTFST and *SPCCHRLMTFST cannot both be specified.
- *LTRLMTLST - The last character of the password may not be a letter character. This value cannot be specified if *DGTLMTLST and *SPCCHRLMTLST values are also specified.

- *LTRMAXn - Where n is a number from 0 to 9. Specifies the maximum number of letter characters that may occur in the password.
Only one *LTRMAXn value can be specified. If a *LTRMINn value is also specified, the n value specified for *LTRMAXn must be greater than or equal to the n value specified for *LTRMINn.
- *LTRMINn - Where n is a number from 0 to 9. Specifies the minimum number of letter characters that must occur in the password.
Only one *LTRMINn value can be specified. If a *LTRMAXn value is also specified, the n value specified for *LTRMAXn must be greater than or equal to the n value specified for *LTRMINn.
- *MAXLENnnn - Where nnn is a number from 1 to 128 (without leading zeroes). The maximum number of characters in a password.
If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.
The nnn value specified must be large enough to accommodate all *MIXCASEn, *DGTMAXn, *LTRMAXn, *SPCCHRMAXn, first and last character restrictions, and non-adjacent character requirements.
If *MINLENnnn is also specified, the nnn value specified for *MAXLENnnn must be greater than or equal to the nnn value specified for *MINLENnnn.
If no *MAXLENnnn value is specified, a value of *MAXLEN10 is assumed if the system is operating with a QPWDLVL value of 0 or 1 or a value of *MAXLEN128 is assumed if the system is operating with a QPWDLVL value of 2 or 3.
- *MINLENnnn - Where nnn is a number from 1 to 128 (without leading zeroes). The minimum number of characters in a password.
If the system is operating at QPWDLVL 0 or 1, the valid range is 1-10. If the system is operating at QPWDLVL 2 or 3, the valid range is 1-128.
If *MAXLENnnn is also specified, the nnn value specified for *MAXLENnnn must be greater than or equal to the nnn value specified for *MINLENnnn.
If no *MINLENnnn value is specified, a value of *MINLEN1 is assumed.
- *MIXCASEn - Where n is a number from 0 to 9. The password must contain at least n uppercase and n lowercase letters. This value is rejected if the system is operating with a QPWDLVL value of 0 or 1 because passwords are required to be uppercase.
Only one *MIXCASEn value can be specified.
If a *LTRMAXn value is specified, the n value specified for *LTRMAXn must be greater than or equal to two times the n value specified for *MIXCASEn.
- *REQANY3 - The password must contain characters from at least three of the following four types of characters.
 - Uppercase letters
 - Lowercase letters
 - Digits
 - Special characters
 When the system is operating with a QPWDLVL of 0 or 1, *REQANY3 has the same effect as if *DGTMIN1, *LTRMIN1, and *SPCCHRMIN1 were all specified.
- *SPCCHRLMTAJC - The password may not contain 2 or more adjacent (consecutive) special characters.
- *SPCCHRLMTFST - The first character of the password may not be a special character. This value cannot be specified if *DGTLMTFST and *LTRLMTFST values are also specified. If the system is operating with a QPWDLVL value of 0 or 1, *LTRLMTFST and *SPCCHRLMTFST cannot both be specified.
- *SPCCHRLMTLST - The last character of the password may not be a special character. This value cannot be specified if *DGTLMTLST and *LTRLMTLST values are also specified.

- *SPCCHRMAXn - Where n is a number from 0 to 9. Specifies the maximum number of special characters that may occur in the password.

Only one *SPCCHRMAXn value can be specified. If a *SPCCHRMINn value is also specified, the n value specified for *SPCCHRMAXn must be greater than or equal to the n value specified for *SPCCHRMINn.

- *SPCCHRMINn - Where n is a number from 0 to 9. Specifies the minimum number of special characters that must occur in the password.

Only one *SPCCHRMINn value can be specified. If a *SPCCHRMAXn value is also specified, the n value specified for *SPCCHRMAXn must be greater than or equal to the n value specified for *SPCCHRMINn.

QPWDVLDPGM

Password validation program provides the ability for a user-written program to do additional validation on passwords. Changes made to this system value take effect the next time a password is changed. See Password validation program for additional information.

QPWRDWNLMT

Maximum amount of time (in seconds) allowed for PWRDWN SYS *IMMED. This is the time used to wait for power down to complete normally after either of the following happens:

- A Power Down System (PWRDWN SYS) command with *IMMED specified for the **How to end** (OPTION) parameter is entered.
- A PWRDWN SYS command with *CNTRLD specified for the **How to end** (OPTION) parameter is entered and the time specified for the **Controlled end delay time** (DELAY) parameter has ended.

Changes to this value take effect when a PWRDWN SYS command is entered.

QPWRRSTIPL

Automatic initial program load (IPL) after power restored allowed. Changes made to this system value take effect the next time there is a power failure.

- 0 means no auto-IPL after power restored.
- 1 means auto-IPL after power restored.

On partitioned AS/400 7xx and iSeries 8xx servers, this system value can only be changed from the primary partition. Whether or not a secondary partition is IPLed at the same time as the primary partition depends on the secondary partition's configuration value for IPL action.

On partitioned eServer i5 servers and System i, this system value must be changed from the service processor's Advanced System Management (ASM) interface.

For more information on partitions, see the Logical Partitions topic in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

QQRVDEGREE

Query parallel processing degree. The value specifies the parallel processing degree available to users of the system.

- *NONE means no parallel processing is allowed for database query processing or database file keyed access path builds or rebuilds.
- *IO means any number of tasks can be used when the database query optimizer chooses to use I/O parallel processing for queries. SMP parallel processing is not allowed, including when building or rebuilding database file keyed access paths.
- *OPTIMIZE means the query optimizer can choose to use any number of tasks for either I/O or SMP parallel processing to process the query or database file keyed access path build or rebuild. Use of parallel processing and the number of tasks used is determined with respect to the number of processors available in the pool in which the job is run, and whether the expected elapsed time for the query or database file keyed access path build or rebuild, is limited by CPU processing or I/O resources.

- *MAX means the query optimizer can choose to use either I/O or SMP parallel processing to process the query. The choices made by the query optimizer will be similar to those made for the value *OPTIMIZE except the optimizer will assume that all active memory in the pool can be used to process the query or database file keyed access path build or rebuild.

QQRYTIMLMT

Query processing time limit.

- *NOMAX means the maximum query interval is used.
- 0-2147352578 means the number of seconds allowed for query processing.

QRCLSPLSTG

Automatic deletion of empty spooled members is allowed based on the member retention interval. Changes made to this system value take effect immediately.

- *NONE means no retention interval.

Note: Using this value can have adverse effects on system performance. More information is in the Files and file systems category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

- *NOMAX means all empty members are kept.
- 1-366 means the number of days that empty spooled members are kept for new spooled file use.

QRETSVRSEC

Retain server security data indicator. This value determines whether the security data needed by a server to authenticate a user on a target system through client/server interfaces can be retained on this system.

- 0 means that the server security data is not retained.
- 1 means that the server security data is retained.

QRMTSRVATR

Remote service attribute. The QRMTSRVATR system value controls the remote service problem analysis ability. The value allows the system to be analyzed remotely.

- 0 means the remote service attribute is off.
- 1 means the remote service attribute is on.

QRMTIPL

Remote power on and IPL indicator. Changes made to this system value take effect immediately.

- 0 means remote power on and IPL are not allowed.
- 1 means remote power on and IPL are allowed.

Note: Any telephone call will cause the system to IPL.

On partitioned AS/400 7xx and iSeries 8xx servers, this system value can only be changed from the primary partition. Whether or not a secondary partition is IPLed at the same time as the primary partition depends on the secondary partition's configuration value for IPL action.

On partitioned eServer i5 servers and System i, this system value must be changed from the service processor's Advanced System Management (ASM) interface.

For more information on partitions, see the Logical Partitions topic in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

QRMTSIGN

Remote sign-on control. Changes made to this system value take effect immediately.

- *FRCSIGNON means normal sign-on required.
- *SAMEPRF means when the source and target user profile are the same, the sign-on can be bypassed for remote sign-on attempts.
- *REJECT means no remote sign-on is allowed.

- *VERIFY means after verifying that the user has access to the system, the system allows the user to bypass the sign-on.
- program means you can specify a program to decide which remote sessions will be allowed and which user profiles can be automatically signed-on from which locations.

QSAVACPTH

Save access paths. Changes made to this system value take effect at the start of the next save operation.

- 0 means do not save logical file access paths that are dependent on the physical files that are being saved.
- 1 means save logical file access paths that are dependent on the physical files that are being saved.

QSCANFS

Scan file systems. This system value specifies the integrated file systems in which objects will be scanned when exit programs are registered with any of the integrated file system scan-related exit points. Changes made to this system value take effect immediately. See Scan file systems for additional information.

QSCANFSCTL

Scan file systems control. This system value controls the integrated file system scanning on the system when exit programs are registered with any of the integrated file system scan-related exit points. These controls apply to integrated file system objects in the file systems covered by the QSCANFS(Scan file systems) system value. Changes made to this system value take effect immediately. See Scan file systems control for additional information.

QSCPFCONS

IPL action with console problem. Changes to this system value take effect before the next IPL.

- 0 means end system.
- 1 means continue the unattended IPL.

QSECOND

Second of the minute. Changes made to this system value take effect immediately.

QSECURITY

System security level. Changes made to this system value take effect at the next IPL.

- 20 means the system requires a password to sign-on.
- 30 means password security at sign-on and object security at each access. You must have authority to access all system resources.
- 40 means password security at sign-on and object security at each access. Programs that try to access objects through interfaces that are not supported will fail.
- 50 means the system requires a password to sign on and users must have authority to access objects and system resources. The security and integrity of the QTEMP library and user domain objects are enforced. Programs that try to access objects through interfaces that are not supported or that try to pass unsupported parameter values to supported interfaces will fail.

QSFWERRLOG

Software error log. Indicates whether system-detected software problems are entered in the error log. Changes made to this system value take effect immediately.

- *LOG means that when a software error is detected by the system, the error is evaluated to determine if it should be logged unconditionally, or if the decision to log the error should be deferred to the policy based Service Monitor.

If the error is to be logged unconditionally, a PARable message is sent to QSYSOPR and an entry is created in the problem log. If the reporting component provides error data, a spooled file is created to contain the data. The spooled file name is stored in the error log and problem log entries.

If the error is to be conditionally logged, the decision to log the error will be made by the policy based Service Monitor. If the decision is to log the problem, an entry is created in the problem log. The problem data will be stored in a problem data library and the problem record entry will be updated with the name of the library.

- *NOLOG means no logging will occur if a software error is detected.

QSHRMEMCTL

Shared memory control. Specifies whether or not users can use shared memory, or use mapped memory that has write capability. Changes made to this system value take effect immediately.

- 0 means that users cannot use shared memory, or use mapped memory that has write capability.
- 1 means that users can use shared memory or mapped memory that has write capability.

QSPCENV

Special environment. The system environment used as the default for all users. Changes made to this system value take effect the next time a user signs on to the system.

- *NONE means no special environment is entered when you sign on.
- *S36 means the System/36 environment is entered when you sign on.

QSPLFACN

Spooled file action. Specifies whether spooled files are kept with a job or detached from the job. Keeping spooled files with jobs allows job commands such as the Work with Submitted Jobs (WRKSBMJOB) command to work with the spooled files even after the job has ended. Detaching spooled files from jobs reduces the use of system resources by allowing job structures to be recycled when the job ends. A change to this system value takes effect for all jobs that become active after the change. The shipped value is *KEEP.

- *KEEP means that when the job ends, as long as at least one spooled file for the job exists in the system auxiliary storage pool (ASP number 1) or in a basic user ASP (ASP numbers 2-32), the spooled files are kept with the job and the status of the job is updated to indicate that the job has completed. If all remaining spooled files for the job are in independent ASPs (ASP numbers 33-255), the spooled files will be detached from the job and the job will be removed from the system.
- *DETACH means the spooled files are detached from the job when the job ends.

QSRLNBR

System serial number. This value cannot be changed. If is retrieved from the data fields by the system when installing the operating system licensed program. You can display QSRLNBR, or you can retrieve this value in user-written programs. The system serial number is the same in each partition on a system.

QSRTSEQ

Sort sequence. This system value specifies the default sort sequence algorithm to be used by the system. Changes made to this system value take effect for jobs started after the change is made.

QSRVDMP

Service dumps. Indicates whether service dumps for escape messages that are not monitored are created. Changes made to this system value take effect immediately.

- *DMPUSRJOB means that service dumps are created only for user jobs, not system jobs.
- *DMPSYSJOB means that service dumps are created only for system jobs, not user jobs. System jobs include the operating system, subsystem monitors, LU service process, spooled readers and writers, and the SCPF job.
- *DMPALLJOB means that service dumps are created for all jobs.
- *NONE means no service dumps are created.

QSSLCSL

Secure Sockets Layer (SSL) cipher specification list. This system value specifies the list of cipher suites that are supported by System SSL. The values are read-only unless the QSSLCSLCTL (SSL cipher control) system value is set to *USRDFN.

For details on System SSL and SSL ciphers, see the SSL section of the Security Reference information in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

A change to this system value takes effect immediately for all subsequent System SSL sessions. The shipped value is *RSA_AES_128_CBC_SHA, *RSA_RC4_128_SHA, *RSA_RC4_128_MD5, *RSA_AES_256_CBC_SHA, *RSA_3DES_EDE_CBC_SHA, *RSA_DES_CBC_SHA, *RSA_EXPORT_RC4_40_MD5, *RSA_EXPORT_RC2_CBC_40_MD5, *RSA_NULL_SHA, and *RSA_NULL_MD5.

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

A cipher cannot be added to QSSLCSL if the required SSL protocol value for the cipher suite is not set for the QSSLPCL (SSL protocol list) system value.

- *RSA_AES_128_CBC_SHA - Use the RSA encoding algorithms for the Advanced Encryption Standard (AES) cipher with cipher block changing (CBC) and 128 bit keys. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_RC4_128_SHA - Use the RSA encoding algorithms for the Rivest Cipher 4 (RC4) cipher and 128 bit keys. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_RC4_128_MD5 - Use the RSA encoding algorithms for the Rivest Cipher 4 (RC4) cipher and 128 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).
- *RSA_AES_256_CBC_SHA - Use the RSA encoding algorithms for the Advanced Encryption Standard (AES) cipher with cipher block changing (CBC) and 256 bit keys. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_3DES_EDE_CBC_SHA - Use the RSA encoding algorithms for the Triple Data Encryption Standard (3DES) cipher with the encrypt/decrypt/encrypt (EDE) and cipher block changing (CBC) modes and 168 bit keys. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_DES_CBC_SHA - Use the RSA encoding algorithms for the Data Encryption Standard (DES) cipher with the cipher block changing (CBC) mode and 56 bit keys. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_EXPORT_RC2_CBC_40_MD5 - Use the RSA encoding algorithms for the Rivest Cipher 2 (RC2) cipher with the cipher block changing (CBC) mode and 40 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).
- *RSA_EXPORT_RC4_40_MD5 - Use the RSA encoding algorithms for the Rivest Cipher 4 (RC4) cipher and 40 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).
- *RSA_NULL_SHA - Use the RSA encoding algorithms but do not use any cipher. Use Secure Hash Algorithm (SHA) for generating message authentication codes (MAC).
- *RSA_NULL_MD5 - Use the RSA encoding algorithms but do not use any cipher. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).
- *RSA_RC2_CBC_128_MD5 - Use the RSA encoding algorithms for the Rivest Cipher 2 (RC2) cipher with the cipher block changing (CBC) mode and 128 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).
- *RSA_3DES_EDE_CBC_MD5 - Use the RSA encoding algorithms for the Triple Data Encryption Standard (3DES) cipher with the encrypt/decrypt/encrypt (EDE) and cipher block changing (CBC) modes and 168 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).

- *RSA_DES_CBC_MD5 - Use the RSA encoding algorithms for the Data Encryption Standard (DES) cipher with the cipher block changing (CBC) mode and 56 bit keys. Use message digest algorithm 5 (MD5) for generating message authentication codes (MAC).

Note: System SSL uses the sequence of the values in QSSLCSL to order the System SSL default cipher specification list. The default cipher specification list entries are system defined and can change on release boundaries. A default cipher removed from QSSLCSL results in the cipher's removal from the default list. The default cipher is added back to the default cipher specification list when it is added back into QSSLCSL. It is not possible to add other ciphers to the default list beyond the system defined set for the release.

QSSLCSLCTL

Secure Sockets Layer (SSL) cipher control. This system value specifies whether or not the QSSLCSL (SSL cipher specification list) system value is controlled by the system or by the user.

For details on System SSL and SSL ciphers, see the SSL section of the Security Reference information in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

A change to this system value takes effect immediately. The shipped value is *OPSYS.

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

- *OPSYS - The QSSLCSL (SSL cipher specification list) system value is read only. The values contained in the QSSLCSL (SSL cipher specification list) system value are automatically modified to contain the list of supported cipher suites as determined by the operating system release.

Note: *OPSYS allows the values to be automatically updated with newer and stronger ciphers when installing to a future release that has new cipher suite capabilities.

- *USRDFN - The QSSLCSL (SSL cipher specification list) system value is modifiable.

Note: Additional cipher suite capabilities will not be added automatically when moving to a future release. You will have to determine what if any new cipher suites are available and add them to the QSSLCSL (SSL cipher specification list) system value manually.

QSSLPCL

Secure Sockets Layer (SSL) protocols. This system value specifies the SSL protocol versions supported by System SSL.

For details on System SSL and SSL Protocols, see the SSL section of the Security Reference information in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

A change to this system value takes effect immediately for all subsequent System SSL sessions. The shipped value is *OPSYS.

Note: You must have *IOSYSCFG, *ALLOBJ, and *SECADM special authorities to change this system value.

- *OPSYS - The SSL protocols supported are determined by the system. The protocols can be different with each operating system release. See the SSL section of the Security Reference information in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for the values supported for your release.
- *TLSV1 - Transport Layer Security version 1.0 will be supported. This value cannot be specified if the *OPSYS value is also specified.
- *SSLV3 - Secure Sockets Layer version 3.0 will be supported. This value cannot be specified if the *OPSYS value is also specified.
- *SSLV2 - Secure Sockets Layer version 2.0 will be supported. This value cannot be specified if the *OPSYS value is also specified.

QSTGLOWACN

Auxiliary storage lower limit action. Specifies the action to take when the available storage in the system ASP goes below the auxiliary storage lower limit. A change to this system value takes effect immediately. The shipped value is *MSG.

- *MSG: Send message CPI099C to QSYSMSG and QSYSOPR message queue. This message is also sent for the other actions.
- *CRITMSG: Send critical message CPI099B to the user specified in the service attribute to receive critical messages.
- *REGFAC: Submit a job to call exit programs registered for the QIBM_QWC_QSTGLOWACN exit point.
- *ENDSYS: End the system to the restricted state.
- *PWRDWN SYS: Power down the system immediately and restart it.

QSTGLOWLMT

Auxiliary storage lower limit. Specifies the percent of available storage remaining in the system ASP when the auxiliary storage lower limit action is taken. A change to this system value takes effect immediately. The shipped value is 5.0.

- Lower limit: Percentage of available storage remaining in the system ASP when the action specified in QSTGLOWACN is taken. The percent of storage currently used in the system ASP can be viewed with the Work with System Status (WRKSYSSTS) command.

QSTRPRTWTR

Start print writers at initial program load (IPL). This system value is set by the system at the time of IPL or is set by the user on the IPL Options display. This system value cannot be changed using the Change System Value (CHGSYSVAL) command.

- 0 means print writers were not started.
- 1 means print writers were started.

QSTRUPPGM

Start-up program name from autostart job in the controlling subsystem. Both an object name and library name can be specified. Changes made to this system value take effect at the next IPL.

QSTSMMSG

Indicates whether status messages are shown. Changes made to this system value take effect the next time a user signs on to the system.

- *NORMAL means status messages will be shown.
- *NONE means status messages will not be shown.

QSVRAUTITV

Server authentication interval. The operating system no longer uses this system value. Changes made to this system value have no effect.

QSYSLIBL

System part of the library list. Changes made to this system value take effect for jobs started after the change is made.

QTHDRSCADJ

Thread resources adjustment. This system value specifies whether or not the system should dynamically make adjustments to the affinity or preference of threads currently running in the system to certain processors and memory. If some resources are being utilized more than others, the system may reassign some of the threads running on the more heavily utilized resources to have affinity to the less utilized resources. Changes made to this system value take effect immediately. The shipped value is '1.'

- '0' means no automatic adjustment of threads is made by the system. Threads will continue to have affinity to the resources which they are currently assigned to until they end or until the system value is changed.

- '1' means the system dynamically makes adjustments of threads' affinity to the system's resources. It does not change the grouping or level of affinity in the threads.

QTHDRSCAFN

Thread resources affinity. The affinity or preference of threads to certain processors and memory. Changes made to this system value take effect immediately for threads in jobs that are started after the change, but has no effect on threads currently running.

- *NOGROUP - Secondary threads will not necessarily have affinity to the same group of processors and memory as their initiating thread.
- *GROUP - Secondary threads will have affinity to the same group of processors and memory as their initiating thread.

The thread resources affinity level can be set to the following values:

- *NORMAL - A thread will use any processor or memory if the resources it has affinity to are not readily available.
- *HIGH - A thread will only use the resources it has affinity to, and will wait until they become available if necessary.

QTIMADJ

Time adjustment. This system value can be used to identify software that adjusts the system clock to keep it synchronized with an external time source. This value should be maintained by time adjustment software and is intended as an aid to prevent having multiple time adjustment applications conflict with each other. There are no checks performed by the system to verify this value or that software is or is not performing time adjustments. IBM time adjustment offerings will use identifiers that start with QIBM such as 'QIBM_OS400_SNTP'. Other software suppliers should follow a similar naming convention of company name and product name.

Time adjustment software should check QTIMADJ prior to starting. If QTIMADJ has an identifier for other time adjustment software, then the software being started should notify the user of this potential conflict and confirm that this time adjustment software should be started. When QTIMADJ is *NONE the software should update QTIMADJ to identify that it is now responsible for adjusting the system clock. Time adjustment software should check QTIMADJ again prior to ending. QTIMADJ should be set to *NONE only if the current value identifies this time adjustment software that is ending. Changes made to this system value take effect immediately. The shipped value is *NONE.

- *NONE - Indicates that time adjustment software has not been identified.
- Identifier - Identify the software that will be used to adjust the system clock.

QTIME

Time of day. Changes made to this system value take effect immediately.

QTIMSEP

Time separator. Changes made to this system value take effect for jobs started after the change is made.

This value affects jobs for which *SYSVAL is specified as the time separator. When specifying time on commands, users must use the time separator specified for their job or no time separator. If a time separator different from the job's time separator is used to specify time on a command, the command will fail.

QTIMZON

Time zone. This specifies the name of the time zone description used to calculate local system time. A change to a different time zone description may result in a different offset that is associated with this new time zone description. A different offset would cause the local system time (system value QTIME) to change. In addition, the system value QUTCOFFSET will be changed to match this new offset. Changes made to this system value take effect immediately.

QTOTJOB

The total number of jobs for which storage must be allocated. Changes made to this system value take effect the next time the job tables are rebuilt during the IPL.

QTSEPOOL

Indicates whether interactive jobs should be moved to another main storage pool when they reach time slice end. Changes made to this system value take effect for jobs started after the change is made.

- *NONE means jobs are not moved when time slice end is reached.
- *BASE means jobs are moved when time slice end is reached.

QUPSDLYTIM

Uninterruptible power supply delay time. Changes made to this system value take effect the next time there is a power failure.

- *BASIC and *CALC cause the Licensed Internal Code (LIC) to assign specific values as the delay time.
- *NOMAX means the system will not start any action on its own.
- 0 means the system will power down automatically when system utility power fails.
- 1-99999 means specify the delay time in seconds before the system powers down.

On some partitioned systems, this system value can only be changed from the primary partition.

For more information on partitions, see the Logical Partitions topic in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

QUPSMMSGQ

Message queue for uninterruptible power supply messages. Changes made to this system value take effect the next time there is a power failure.

QUSEADPAUT

Defines which users can create, change and update programs and service programs with the (use adopted authority) USEADPAUT(*YES) attribute. When a program or service program has a use adopted authority attribute of *YES, the program/service program can use any adopted authority that is being passed to it from a program/ service program higher in the call stack.

This system value has no effect on the following:

- Existing programs/service programs created with the USEADPAUT(*YES) attribute. Users are responsible for deciding which existing programs/service programs should be changed to have USEADPAUT(*NO).
- Restoring a program/service program that uses adopted authority. These program/service programs can still be restored on your system.
- Duplicating a program/service program that uses adopted authority. The USEADPAUT attribute of the existing program/service program is copied to the new object.

The following values can be specified:

- *NONE means there is no restriction on who can create, change or update a program/service program to use adopted authority. Any user can create, change or update a program/service program to have the USEADPAUT(*YES) attribute.
- Name means you can specify the name of the authorization list which will control which users can set the USEADPAUT(*YES) attribute. The user needs *USE authority to the authorization list to be able to create, change or update programs/service programs with the USEADPAUT(*YES) attribute. Authority to the authorization list cannot come from adopted authority. That is, if you are running a program that adopts authority, the adopted authority is not used when checking authority to the authorization list.

QUSRLIBL

User part of the library list. Changes made to this system value take effect for jobs started after the change is made.

QUTCOFFSET

Indicates the number of hours (in 24-hour format) and minutes that the current system time is offset from the Coordinated Universal Time (UTC).

- +hhmm means that the current system time is hh hours and mm minutes ahead of UTC.
- -hhmm means that the current system time is hh hours and mm minutes behind UTC.

Note: This system value must be the same as the offset that is associated with the time zone description specified in the system value QTIMZON. A change to a different time zone description for QTIMZON may result in a different associated offset. The system value QUTCOFFSET will be changed as well to match this new offset. QUTCOFFSET cannot be changed to a value that is different than the offset currently associated with QTIMZON. If an attempt is made to do so, the diagnostic message CPD1687 will be issued.

QVFYOBJRST

Verify object on restore. This system value specifies the policy to be used for object signature verification during a restore operation. This value applies to objects of types: *CMD, *PGM, *SRVPGM, *SQLPKG and *MODULE. It also applies to *STMF objects which contain Java programs. This value also specifies the policy for PTFs applied to the system including Licensed Internal Code fixes. Changes made to this system value take effect immediately. See Verify object on restore for additional information.

QYEAR

Year. Changes made to this system value take effect immediately.

Top

CL variable for returned value (RTNVAR)

Specifies the name of the CL program variable that receives the value of the system value being returned. The type and length for the CL variable when it was declared must be compatible with that of the system value being received. The attributes of individual system values are described in the Work management topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

In general, the return variable type must match the system value type. For character system values that are 1 character long, the CL variable can be a character or logical variable. For character and logical system values, the length of the CL variable must equal the length of the system value. For decimal system values, the CL variable length must be greater than or equal to the length of the system value.

The following table lists the type and length of each system value.

SYSTEM VALUE	TYPE	LENGTH
QABNORMSW	Character	1
QACGLVL	Character	80
QACTJOB	Decimal	(5 0)
QADLACTJ	Decimal	(5 0)
QADLSPLA	Decimal	(5 0)
QADLTOTJ	Decimal	(5 0)
QALWJOBITP	Character	1
QALWOBJRST	Character	150
QALWUSRDMN	Character	500
QASTLVL	Character	10
QATNPGM	Character	20
QAUDCTL	Character	50
QAUDENDACN	Character	10
QAUDFRCLVL	Decimal	(5 0)
QAUDLVL	Character	160
QAUDLVL2	Character	990

QAUTOCFG	Character	1
QAUTORMT	Character	1
QAUTOSPRPT	Character	1
QAUTOVRT	Decimal	(5 0)
QBASACTLVL	Decimal	(5 0)
QBASPOOL	Decimal	(10 0)
QBOOKPATH	Character	(315)
QCCSID	Decimal	(5 0)
QCENTURY	Character	1
QCFGMSGQ	Character	20
QCHRID	Character	20
QCHRIDCTL	Character	10
QCMNARB	Character	10
QCMNRCYLMT	Character	20
QCNTYID	Character	2
QCONSOLE	Character	10
QCRTAUT	Character	10
QCRTOBJAUD	Character	10
QCTLSBSD	Character	20
QCURSYM	Character	1
QDATE	Character	5 (for Julian dates) or 6
QDATETIME	Character	20
QDATFMT	Character	3
QDATSEP	Character	1
QDAY	Character	2 or 3 (for Julian dates)
QDAYOFWEEK	Character	4
QDBFSTCCOL	Character	10
QDBRCVYWT	Character	1
QDECFMT	Character	1
QDEVNAMING	Character	10
QDEVRCYACN	Character	20
QDSCJOBITV	Character	10
QDSPSGNINF	Character	1
QDYNPTYADJ	Character	1
QDYNPTYSCD	Character	1
QENDJOBLMT	Decimal	(5 0)
QFRCCVNRST	Character	1
QHOUR	Character	2
QHSTLOGSIZ	Decimal	(8 0)
QIGC	Character	1
QIGCCDEFNT	Character	20
QIGCFNTSIZ	Decimal	(4 1)
QINACTIV	Character	10
QINACTMSGQ	Character	20
QIPLDATTIM	Character	20
QIPLSTS	Character	1
QIPLTYPE	Character	1

QJOBMSGQFL	Character	10
QJOBMSGQMX	Decimal	(5 0)
QJOBMSGQSZ	Decimal	(5 0)
QJOBMSGQTL	Decimal	(5 0)
QJOBSPLA	Decimal	(5 0)

QKBDBUF	Character	10
QKBDTYPE	Character	3

QLANGID	Character	3
QLEAPADJ	Decimal	(5 0)
QLIBLCKLVL	Character	1
QLMTDEVSSN	Character	1
QLMTSECOFR	Character	1
QLOCALE	Character	1024
QLOGOUTPUT	Character	10

QMAXACTLVL	Decimal	(5 0)
QMAXJOB	Decimal	(10 0)
QMAXSGNACN	Character	1
QMAXSIGN	Character	6
QMAXSPLF	Decimal	(10 0)
QMCHPOOL	Decimal	(10 0)
QMINUTE	Character	2
QMLTTHDACN	Character	1
QMODEL	Character	4
QMONTH	Character	2

QPASTHRSVR	Character	10
QPFRADJ	Character	1
QPRBFTR	Character	20
QPRBHLDTIV	Decimal	(5 0)
QPRCFEAT	Character	4
QPRCMLTTSK	Character	1
QPRTDEV	Character	10
QPRTKEYFMT	Character	10
QPRTTXT	Character	30
QPWDCHGBLK	Character	10
QPWDEXPITV	Character	6
QPWDEXPWRN	Decimal	(5 0)
QPWDLMTAJC	Character	1
QPWDLMTCHR	Character	10
QPWDLMTREP	Character	1
QPWDLVL	Decimal	(5 0)
QPWDMAXLEN	Decimal	(5 0)
QPWDMINLEN	Decimal	(5 0)
QPWDPOSDIF	Character	1
QPWDRQDDGT	Character	1
QPWDRQDDIF	Character	1
QPWDRULES	Character	750
QPWDVLDPGM	Character	20
QPWRDWNLMT	Decimal	(5 0)
QPWRRSTIPL	Character	1

QQRVDEGREE	Character	10
QQRVTIMLMT	Character	10

QRCLSPLSTG	Character	10
QRETSVRSEC	Character	1
QRMTIPL	Character	1
QRMTSRVATR	Character	1
QRMTSIGN	Character	20
QSAVACPTH	Character	1
QSCANFS	Character	200
QSCANFCTL	Character	200
QSCPFCONS	Character	1
QSECOND	Character	2
QSECURITY	Character	2
QSETJOBATR	Character	160
QSFWRRLLOG	Character	10
QSHRMEMCTL	Character	1
QSPCENV	Character	10
QSPLFACN	Character	10
QSRLNBR	Character	8
QSRSEQ	Character	20
QSRVDM	Character	10
QSSLCSL	Character	1280
QSSLCSLCTL	Character	10
QSSLPCL	Character	100
QSTGLOWACN	Character	10
QSTGLOWMT	Decimal	(7 4)
QSTRPRTWTR	Character	1
QSTRUPPGM	Character	20
QSTMSG	Character	10
QSVRAUTITV	Decimal	(6 0)
QSYSLIBL	Character	150
QTHDRSCADJ	Character	1
QTHDRSCAFN	Character	20
QTIMADJ	Character	30
QTIME	Character	6, 7, 8, or 9 (for seconds, tenths, hundredths, or thousandths of a second)
QTIMSEP	Character	1
QTIMZON	Character	10
QTOTJOB	Decimal	(5 0)
QTSEPOOL	Character	10
QUPSDLYTIM	Character	20
QUPSMGQ	Character	20
QUSEADPAUT	Character	10
QUSRLIBL	Character	250
QUTCFFSET	Character	5
QVFYOBJRST	Character	1
QYEAR	Character	2

This is a required parameter.

Examples

```
RTVSYSVAL  SYSVAL(QDATE)  RTNVAR(&DATE)
```

This command retrieves the date value from the system value QDATE and copies it into the CL variable &DATE. The CL variable must be declared as a 6-character variable to match the attributes of the QDATE system value.

Top

Error messages

*ESCAPE Messages

CPF1028

&1 not valid for parameter SYSVAL.

CPF1074

SYSVAL(QMONTH) not valid for Julian date format.

CPF1094

CL variable not same type as system value &1.

CPF1095

CL variable length not valid for system value &1.

CPF1842

Cannot access system value &1.

CPF268D

Unable to access system value &1.

Top

Retrieve Table Source (RTVTBLSRC)

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

Parameters
Examples
Error messages

The Retrieve Table Source (RTVTBLSRC) command is used to retrieve the source for a *CVT or *SRTSEQ table. These source statements are placed into a source file member, which can be used as input when creating a table with the CRTTBL command. Additionally, this command will convert the source of a *SRTSEQ table to the format used to create a *UCSSRTSEQ table.

Top

Parameters

Keyword	Description	Choices	Notes
TBL	Table	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Table	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCFILE	Source file	<i>Qualified object name</i>	Required, Positional 2
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name, *TBL</i>	Optional, Positional 3
CVTTOUCS	Convert to UCS-2	<i>*YES, *NO</i>	Optional
TEXT	Text 'description'	<i>Character value, *BLANK</i>	Optional

Top

Table (TBL)

Specifies the qualified name of the table whose source is being retrieved.

Top

Source file (SRCFILE)

Specifies the qualified name of the previously created database source file into which the table source is being written.

The name of the source file can be qualified by one of the following library values:

***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.

library-name

Specify the name of the library to be searched.

Source member (SRCMBR)

Specifies the name of the database source file member into which the table source is being written. If not specified, the table name is assumed. If the member existed before running the command, it is cleared before any source statements are written into it. If the member did not exist, it is created.

The possible values are:

***TBL** The name of the table is used as the member name.

source-member-name

Specify the name of the source file member to contain the retrieved table source.

Convert to UCS-2 (CVTTOUCS)

Specifies whether the source of a *SRTSEQ table should be converted when placed in the file from the format of a *SRTSEQ table to that of a *UCSSRTSEQ. You would want to do this if you want to create a *UCSSRTSEQ but want to base it on an existing *SRTSEQ table.

The possible values are:

***NO** The information is not converted when placed in the file.

***YES** The information is converted when placed in the file.

Text 'description' (TEXT)

Specifies the text that briefly describes the object.

***BLANK**

No text is specified.

'description'

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

Examples

```
RTVTBLSRC  TBL(QSYS/TABLE1)  SRCFILE(TBLSRC)  SRCMBR(TABLEOUT)
           CVTTOUCS(*NO)
```

This command retrieves the source statements from the table named TABLE1 in library QSYS. The retrieved source is placed into the file named TBLSRC and is named as member TABLEOUT and not converted to a *UCSSRTSEQ format.

Error messages

Unknown

[Top](#)

Retrieve User Profile (RTVUSRPRF)

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

Parameters
 Examples
 Error messages

The Retrieve User Profile (RTVUSRPRF) command is used in a control language (CL) program or a REXX procedure to get one or more of the values that are stored and associated with a user. The values are returned in the specified variables for the desired user.

The parameter prompt text lists the minimum length for the variables next to the appropriate parameters you want to retrieve. For character variables, a single number is shown. For decimal variables, two numbers are shown. The first number indicates the minimum variable length and the second number indicates the minimum number of decimal positions.

Restrictions:

- Read (*READ) authority is required for the user profile specified on the USRPRF parameter.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
USRPRF	User profile	Name, <u>*CURRENT</u>	Optional, Positional 1
RTNUSRPRF	CL var for RTNUSRPRF (10)	Character value	Optional
SPCAUT	CL var for SPCAUT (100)	Character value	Optional
MAXSTG	CL var for MAXSTG (11 0)	Decimal number	Optional
STGUSED	CL var for STGUSED (15 0)	Decimal number	Optional
PTYLMT	CL var for PTYLMT (1)	Character value	Optional
INLPGM	CL var for INLPGM (10)	Character value	Optional
INLPGMLIB	CL var for INLPGMLIB (10)	Character value	Optional
JOBBD	CL var for JOBBD (10)	Character value	Optional
JOBDLIB	CL var for JOBDLIB (10)	Character value	Optional
GRPPRF	CL var for GRPPRF (10)	Character value	Optional
OWNER	CL var for OWNER (10)	Character value	Optional
GRPAUT	CL var for GRPAUT (10)	Character value	Optional
ACGCDE	CL var for ACGCDE (15)	Character value	Optional
MSGQ	CL var for MSGQ (10)	Character value	Optional
MSGQLIB	CL var for MSGQLIB (10)	Character value	Optional
OUTQ	CL var for OUTQ (10)	Character value	Optional
OUTQLIB	CL var for OUTQLIB (10)	Character value	Optional
TEXT	CL var for TEXT (50)	Character value	Optional
PWDCHGDAT	CL var for PWDCHGDAT (6)	Character value	Optional
USRCLS	CL var for USRCLS (10)	Character value	Optional
ASTLVL	CL var for ASTLVL (10)	Character value	Optional
SPCENV	CL var for SPCENV (10)	Character value	Optional

Keyword	Description	Choices	Notes
CURLIB	CL var for CURLIB (10)	Character value	Optional
INLMNU	CL var for INLMNU (10)	Character value	Optional
INLMNULIB	CL var for INLMNULIB (10)	Character value	Optional
LMTCPB	CL var for LMTCPB (10)	Character value	Optional
DLVRY	CL var for DLVRY (10)	Character value	Optional
SEV	CL var for SEV (2 0)	Decimal number	Optional
PRTDEV	CL var for PRTDEV (10)	Character value	Optional
ATNPGM	CL var for ATNPGM (10)	Character value	Optional
ATNPGMLIB	CL var for ATNPGMLIB (10)	Character value	Optional
USROPT	CL var for USROPT (240)	Character value	Optional
DSPSGNINF	CL var for DSPSGNINF (7)	Character value	Optional
PWDEXPITV	CL var for PWDEXPITV (5 0)	Decimal number	Optional
PWDEXP	CL var for PWDEXP (4)	Character value	Optional
STATUS	CL var for STATUS (10)	Character value	Optional
PRVSIGN	CL var for PRVSIGN (13)	Character value	Optional
NOTVLDSIGN	CL var for NOTVLDSIGN (11 0)	Decimal number	Optional
LMTDEVSSN	CL var for LMTDEVSSN (7)	Character value	Optional
KBDBUF	CL var for KBDBUF (10)	Character value	Optional
LANGID	CL var for LANGID (10)	Character value	Optional
CNTRYID	CL var for CNTRYID (10)	Character value	Optional
CCSID	CL var for CCSID (5 0)	Decimal number	Optional
SRTSEQ	CL var for SRTSEQ (10)	Character value	Optional
SRTSEQLIB	CL var for SRTSEQLIB (10)	Character value	Optional
OBJAUD	CL var for OBJAUD (10)	Character value	Optional
AUDLVL	CL var for AUDLVL (640)	Character value	Optional
GRPAUTYP	CL var for GRPAUTYP (10)	Character value	Optional
SUPGRPPRF	CL var for SUPGRPPRF (150)	Character value	Optional
UID	CL var for UID (10 0)	Decimal number	Optional
GID	CL var for GID (10 0)	Decimal number	Optional
SETJOBATR	CL var for SETJOBATR (160)	Character value	Optional
CHRIDCTL	CL var for CHRIDCTL (10)	Character value	Optional
LCLPDMGT	CL var for LCLPDMGT (10)	Character value	Optional
PWDCHGBLK	CL var for PWDCHGBLK (10)	Character value	Optional

Top

User profile (USRPRF)

Specifies the user profile whose information you want to retrieve. If a variable is specified, it must be 10 characters in length and contain a user name or the value *CURRENT.

*CURRENT

The user profile that is currently running is used.

name Specify the name of the user profile whose information you want to retrieve.

Top

CL var for RTNUSRPRF (10) (RTNUSRPRF)

Specifies the name of a variable that is used to retrieve the name of the user profile. In CL programs, this should be a 10-character variable. If *CURRENT is specified for the **User profile (USRPRF)** parameter, the value returned is the currently running user profile name. If a name is specified, that name is returned for this parameter.

Top

CL var for SPCAUT (100) (SPCAUT)

Specifies the name of a variable that is used to retrieve the list of special authorities the user has. In CL programs, this should be a 100-character variable. The format returned is a list of up to 10 entries. Each entry is 10 characters in length and each entry contains a special authority. If there are fewer than 10 special authorities in the list, the remaining entries are padded on the right with blanks. If the user has no special authorities, the first entry contains the value of *NONE, followed by blanks.

Top

CL var for MAXSTG (11 0) (MAXSTG)

Specifies the name of a variable that is used to retrieve the maximum amount of auxiliary storage that can be assigned to store permanent objects owned by the specified user. In CL programs, this should be a decimal variable length of (11 0). The value returned is either an 11-digit value or a value of -1 if it is *NOMAX. The variable must be an 11-digit value with no decimal positions.

Top

CL var for STGUSED (15 0) (STGUSED)

Specifies the name of a variable that is used to get the amount of auxiliary storage that is currently being used to store permanent objects owned by the specified user profile. In CL programs, this should be a decimal variable length of (15 0). The value is returned in kilobytes (1 kilobyte (KB) equals 1024 bytes).

Top

CL var for PTYLMT (1) (PTYLMT)

Specifies the name of a variable used to retrieve the highest scheduling priority the user is allowed to have for each job submitted to the system. In CL programs, the variable has a length of 1 character. This value controls the job processing priority that any job running under this user can have. This means that values specified in the JOBPTY and OUTPTY parameters of any job command cannot exceed the PTYLMT value specified for the user under which the job is run. The scheduling priority can have a value ranging from 0 through 9, where 0 is the highest priority and 9 is the lowest priority.

Top

CL var for INLPGM (10) (INLPGM)

Specifies the name of a variable that is used to retrieve the name of the initial program that starts when the specified user signs on to the system. In CL programs, this should be a 10-character variable. If no initial program name is associated with the specified user, the value returned in the variable is *NONE.

Top

CL var for INLPGMLIB (10) (INLPGMLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the initial program associated with the specified user. In CL programs, this should be a 10-character variable. If there is no initial program associated with the specified user, blanks are returned in the variable.

Top

CL var for JOB (10) (JOB)

The name of a variable that is used to retrieve the name of the job description associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for JOBDLIB (10) (JOBDLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the job description associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for GRPPRF (10) (GRPPRF)

Specifies the name of a variable that is used to retrieve the name of the group profile. In CL programs, this should be a 10-character variable. If no group profile exists for the specified user profile, a value of *NONE is returned in the variable.

Top

CL var for OWNER (10) (OWNER)

Specifies the name of a variable that is used to retrieve the special value of *USRPRF or *GRPPRF. This parameter The owner of newly created objects. This is either the specified user or the user's group profile. In CL programs, this should be a 10-character variable. If no group profile exists for the specified user profile, the value returned in the variable is *USRPRF.

Top

CL var for GRPAUT (10) (GRPAUT)

Specifies the name of a variable used to retrieve the authority granted to the group profile for newly-created objects. The special value of *NONE, *CHANGE, *ALL, *USE, or *EXCLUDE is returned in the variable. If there is no group profile for the specified user, or if the group profile is the owner of the objects the specified user creates, the special value returned is *NONE. In CL programs, this should be a 10-character variable.

Top

CL var for ACGCDE (15) (ACGCDE)

The name of a variable that is used to retrieve the value of the accounting code assigned to the specified user. In CL programs, this should be a 15-character variable. If no accounting code exists for the user profile, blanks are returned.

Top

CL var for MSGQ (10) (MSGQ)

The name of a variable that is used to retrieve the name of the message queue associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for MSGQLIB (10) (MSGQLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the message queue associated with the specified user. In CL programs, this should be a 10-character variable.

Top

CL var for OUTQ (10) (OUTQ)

Specifies the name of a variable that is used to retrieve the name of the output queue associated with the specified user. In CL programs, this should be a 10-character variable. The special value *DEV or *WRKSTN is returned in the variable.

Top

CL var for OUTQLIB (10) (OUTQLIB)

Specifies the name of a variable that is used to retrieve the name of the library that contains the output queue associated with the specified user. In CL programs, this should be a 10-character variable. Blanks are returned if the current value for the **Output queue (OUTQ)** parameter is *DEV or *WRKSTN.

Top

CL var for TEXT (50) (TEXT)

Specifies the name of a variable that is used to retrieve the user-defined description for the specified user profile. In CL programs, this should be a 50-character variable. If there is no text associated with the user, blanks are returned in the CL variable.

CL var for PWDCHGDAT (6) (PWDCHGDAT)

Specifies the name of a variable that is used to retrieve the date when the password for the specified user was last changed. The date is returned in the form YYMMDD. In CL programs, this should be a 6-character variable. If the user does not have a date, blanks are returned.

Top

CL var for USRCLS (10) (USRCLS)

Specifies the name of a variable that is used to retrieve the user class for the specified user. A special value of *USER, *SYSOPR, *PGMR, *SECADM, or *SECOFR is returned in the variable. In CL programs, this should be a 10-character variable.

Top

CL var for ASTLVL (10) (ASTLVL)

Specifies the name of a variable that is used to retrieve the assistance level for the specified user. A special value of *SYSVAL, *BASIC, *INTERMED, or *ADVANCED is returned in the variable. In CL programs, this should be a 10-character variable.

Top

CL var for SPCENV (10) (SPCENV)

Specifies the name of a variable that is the starting environment for the specified user. A special value of *SYSVAL, *NONE, or *S36 is returned in the variable. In CL programs, this should be a 10-character variable.

Top

CL var for CURLIB (10) (CURLIB)

The name of the variable that is used to retrieve the name of the job's default library for the specified user. A value of *CRTDFT is returned in the variable if no current library exists for this user. In CL programs, this should be a 10-character variable.

Top

CL var for INLMNU (10) (INLMNU)

Specifies the name of a variable that is used to retrieve the name of the initial menu that is shown when the specified user signs on to the system. In CL programs, this should be a 10-character variable.

Top

CL var for INLMNULIB (10) (INLMNULIB)

The name of the variable that is used to retrieve the library name that contains the initial menu. In CL programs, this should be a 10-character variable.

Top

CL var for LMTCPB (10) (LMTCPB)

Specifies the name of a variable that is used to retrieve the values for the limits to which users can change their user profiles and run commands. In CL programs, this should be a 10-character variable. A special value of *NO, *YES, or *PARTIAL is returned in the variable.

Top

CL var for DLVRY (10) (DLVRY)

Specifies the name of a variable that is used to retrieve the message control delivery value for the specified user profile. In CL programs, this should be a 10-character variable. The special value of *NOTIFY, *BREAK, *HOLD, or *DFT is returned in the variable.

Top

CL var for SEV (2 0) (SEV)

Specifies the name of a variable that is used to retrieve the message control severity level for the specified user. In CL programs this should be a variable length of (2 0).

Top

CL var for PRTDEV (10) (PRTDEV)

Specifies the name of a variable that is used to retrieve the name of the printer device for the specified user. In CL programs, this should be a 10-character variable. A value of *SYSVAL is returned if the printer device name is from the system value QPRTDEV. A value of *WRKSTN is returned if the printer device name is from the printer device assigned to the user's work station.

Top

CL var for ATNPGM (10) (ATNPGM)

Specifies the name of a variable that is used to retrieve the name of the Attention key handling program for the specified user. In CL programs, the variable has a length of 10 characters. A value of *SYSVAL if the attention key handling program is from system value QATNPGM. A value of *NONE is returned if no Attention key handling program was specified for this user.

Top

CL var for ATNPGMLIB (10) (ATNPGMLIB)

Specifies the name of a variable that is used to retrieve the name of the Attention key handling program library for the specified user. In CL programs, this should be a 10-character variable. If *NONE is the current value for the **Attention program (ATNPGM)** parameter, blanks are returned in the variable.

CL var for USROPT (240) (USROPT)

Specifies the name of a variable that is used to retrieve the list of values for user options for the specified user. In CL programs, this should be a 240-character variable. The special value of *NONE or a list of values is returned in the variable.

Top

CL var for DSPSGNINF (7) (DSPSGNINF)

Specifies the name of a variable that is used to retrieve the sign-on information display indicator for the specified user. In CL programs, this should be a 7-character variable. The special value of *SYSVAL, *YES, or *NO is returned in the variable. If *SYSVAL is returned, the display sign-on information indicator is from the system value QDSPSGNINF.

Top

CL var for PWDEXPITV (5 0) (PWDEXPITV)

Specifies the name of a variable that is used to retrieve the password expiration interval for the specified user. In CL programs, the variable specified must be packed (5,0) in length. The value returned is either a number ranging from 1 through 366, 0 if it is *SYSVAL, or -1 if it is *NOMAX. If 0 is returned (*SYSVAL), then the password expiration interval is from the system value QPWDEXPITV.

Top

CL var for PWDEXP (4) (PWDEXP)

Specifies the name of a variable that is used to retrieve the password expired indicator for the specified user. In CL programs, this should be a 4-character variable. The special value of *YES or *NO is returned in the variable.

Top

CL var for STATUS (10) (STATUS)

The name of a variable that is used to retrieve the status of the specified user profile. In CL programs, this should be a 10-character variable. The special value of *ENABLED or *DISABLED is returned in the variable.

Top

CL var for PRVSIGN (13) (PRVSIGN)

Specifies the name of a variable that is used to retrieve the previous sign-on date and time for the specified user. The date and time are returned in the form CYYMMDDHHMMSS. In CL programs, this should be a 13-character variable. If the user has not signed on previously, blanks are returned.

Top

CL var for NOTVLDSIGN (11 0) (NOTVLDSIGN)

Specifies the name of a variable that is used to retrieve the number of password verification attempts that were not valid for the specified user. In CL programs, the variable specified must be packed (11 0) in length.

[Top](#)

CL var for LMTDEVSSN (7) (LMTDEVSSN)

Specifies the name of a variable that is used to retrieve the limit device sessions indicator for the specified user. A value of *SYSVAL, *YES, *NO, 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9 can be returned in the CL variable. If *SYSVAL is returned, the limit device sessions indicator is from the system value QLMTDEVSSN. In CL programs, this should be a 7-character variable.

[Top](#)

CL var for KBDBUF (10) (KBDBUF)

Specifies the name of a variable that is used to retrieve the keyboard buffering value for the specified user. The special value of *SYSVAL, *NO, *TYPEAHEAD, or *YES is returned in the CL variable. If *SYSVAL is returned, the keyboard buffering value is the same as the system value QKBDBUF. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for LANGID (10) (LANGID)

Specifies the name of a variable that is used to retrieve the language identifier for the specified user. The special value *SYSVAL or the language identifier is returned in the variable. If *SYSVAL is returned, the language identifier for the user is determined by the QLANGID system value. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for CNTRYID (10) (CNTRYID)

Specifies the name of a variable that is used to retrieve the country or region identifier for the specified user. The special value *SYSVAL or the country or region identifier is returned in the variable. If *SYSVAL is returned, the country or region identifier for the user is determined by the QCNTRYID system value. In CL programs, this should be a 10-character variable.

[Top](#)

CL var for CCSID (5 0) (CCSID)

Specifies the name of a variable that is used to retrieve the coded character set identifier (CCSID) for the specified user. A 5-digit value is returned or, if the system value QCCSID is used to determine the CCSID for the user, a value of -2 is returned. In CL programs, this should be a decimal variable length of (5 0).

[Top](#)

CL var for SRTSEQ (10) (SRTSEQ)

Specifies the name of a variable used to retrieve the sort sequence table for the specified user. The value returned is one of the following: *HEX, *LANGIDUNQ, *LANGIDSHR, *SYSVAL, or the 10-character table identifier. If *SYSVAL is returned, the table identifier for the user is determined by the QSRTSEQ system value.

Top

CL var for SRTSEQLIB (10) (SRTSEQLIB)

Specifies the name of a variable used to retrieve the sort sequence table library for the specified user. The value returned is the 10-character library identifier. The variable is set to blanks unless a sort sequence table name is specified.

Top

CL var for OBJAUD (10) (OBJAUD)

The name of a 10-character variable that is used to retrieve the object auditing value for the specified user. The special value of *NONE, *CHANGE, or *ALL, as specified on the Change User Audit (CHGUSRAUD) command, is returned in the variable. When the user of this command does not have either *ALLOBJ or *AUDIT special authority, *NOTAVL is returned. When less than 10 characters are returned, the variable is padded on the right with blanks.

Top

CL var for AUDLVL (640) (AUDLVL)

Specifies the name of a variable used to retrieve the action auditing values for the specified user. In CL programs, the variable has a length of 640 characters. The format returned is a list of a maximum of 64 action auditing values, with each value being 10 characters long. If there are fewer than 64 action auditing values to be returned, the remaining values are padded on the right with blanks.

If the user of this command does not have either *ALLOBJ or *AUDIT special authority, the first value returned will be *NOTAVL followed by blanks.

If the user has no action auditing values, the first value returned will be *NONE followed by blanks.

If the user has action auditing values, one or more of the following special values, as specified in the Change User Audit (CHGUSRAUD) command, is returned in the variable.

- *AUTFAIL
- *CMD
- *CREATE
- *DELETE
- *JOBBAS
- *JOBCHGUSR
- *JOBDTA
- *NETBAS
- *NETCLU
- *NETCMN
- *NETFAIL

- *NETSCK
- *OBJMGT
- *OFCSRV
- *OPTICAL
- *PGMADP
- *PGMFAIL
- *PRTDTA
- *SAVRST
- *SECCFG
- *SECDIRSRV
- *SECIPC
- *SECNAS
- *SECRUN
- *SECSCKD
- *SECURITY
- *SECVFY
- *SECVLDL
- *SERVICE
- *SPLFDTA
- *SYSMGT

Top

CL var for GRPAUTYP (10) (GRPAUTYP)

Specifies the name of a variable used to retrieve the type of authority to be granted to the group profile for newly-created objects. In CL programs, the variable has a length of 10 characters. The special value of *PRIVATE or *PGP is returned in the variable.

Top

CL var for SUPGRPPRF (150) (SUPGRPPRF)

Specifies the name of a variable used to retrieve the supplemental group profiles for the specified user. In CL programs, the variable has a length of 150 characters. The format returned is a list of up to 15 supplemental group profiles, with each entry 10 characters long. If there are fewer than 15 supplemental groups in the list, the remaining entries are padded on the right with blanks. If the user has no supplemental groups, the first entry contains the value of *NONE followed by blanks.

The supplemental group profiles are returned in the following format:

Entry-1

Supplemental group profile CHAR(10)

Entry-2

Supplemental group profile CHAR(10)

.

.

.

Entry-15

Supplemental group profile CHAR(10)

Top

CL var for UID (10 0) (UID)

Specifies the name of a variable used to retrieve the user ID number (uid) for the specified user. In CL programs, this should be a decimal variable length of (10 0).

Top

CL var for GID (10 0) (GID)

Specifies the name of a variable used to retrieve the group ID number (gid) for the specified user. In CL programs, this should be a decimal variable length of (10 0). The value returned is either a 10-digit value or a value of 0 if the gid is *NONE. The variable must be a 10-digit value with no decimal positions.

Top

CL var for SETJOBATR (160) (SETJOBATR)

Specifies the name of a variable used to retrieve the job attributes set from the LOCALE path name of the specified user. In CL programs, the variable has a length of 160 characters. The format returned is a list of up to 16 job attributes that are set, with each entry 10 characters long. If there are fewer than 16 attributes in the list, the remaining entries are padded on the right with blanks. If no attributes are set from the locale path name for the user, the first entry contains the value of *NONE followed by blanks.

Top

Character identifier control (CHRIDCTL)

Specifies the name of a variable used to retrieve the character identifier control for the specified user. In CL programs, this variable should have a length of 10 characters. The special value of *DEVVD, *JOBCCSID, or *SYSVAL is returned in the variable. If *SYSVAL is returned, the CHRID control for the user is determined by the QCHRIDCTL system value.

Top

CL var for LCLPWDMGT (10) (LCLPWDMGT)

Specifies the name of a variable used to retrieve local password management for the specified user. In CL programs, this variable should have a length of 10 characters. The special value of *YES or *NO is returned in the variable.

Top

CL var for PWDCHGBLK (10) (PWDCHGBLK)

Specifies the name of a variable used to retrieve the block password change value for the specified user. A value of *NONE, *SYSVAL, or 1-99 can be returned in the CL variable. If *SYSVAL is returned, the block password change value for the user is determined by the QPWDCHGBLK system value. In CL programs, this should be a 10-character variable.

Top

Examples

```
CRTUSRPRF  USRPRF(SMITH)  SPCAUT(*SAVSYS *SECADM)
           MAXSTG(*NOMAX) PTYLM(4)
           INLPGM(*NONE)  MSGQ(QGPL/SMITHMQ)
           OUTQ(QGPL/QSMITH) TEXT('John Smith User Profile')
```

If the SMITH user profile is created using the above command, when user SMITH calls a CL program containing the following:

```
DCL  &UNAME    *CHAR  10
DCL  &URIGHT   *CHAR 100
DCL  &IPGM     *CHAR  10
DCL  &IPGMLB   *CHAR  10
DCL  &UMSGQ    *CHAR  10
DCL  &UMSQLB   *CHAR  10
DCL  &USED     *CHAR  10
:
RTVUSRPRF  USRPRF(*CURRENT) STGUSED(&USED) +
           RTNUSRPRF(&UNAME) SPCAUT(&URIGHT) +
           INLPGM(&IPGM)  INLPGMLIB(&IPGMLB)
```

This command retrieves the user profile information for the job's current user profile and returns the information into the following CL program variables:

```
&UNAME      'SMITH      '
&URIGHT     '*SAVSYS  *SECADM  ...
           |_____100 characters_____|
&IPGM       '*NONE      '
&IPGMLB     '           '
           |_____100 characters_____|
```

Top

Error messages

*ESCAPE Messages

CPF2203

User profile &1 not correct.

CPF2204

User profile &1 not found.

CPF2213

Not able to allocate user profile &1.

CPF2217

Not authorized to user profile &1.

CPF2225

Not able to allocate internal system object.

CPF8134

User profile &4 damaged.

Top

Retrieve User Print Info (RTVUSRPTI)

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

Parameters
Examples
Error messages

The Retrieve User Print Information (RTVUSRPTI) command is used in a CL program to retrieve the user print information value associated with a user profile. The values are returned in the specified CL variables for the desired user.

[Top](#)

Parameters

Keyword	Description	Choices	Notes
USER	User	<i>Name</i> , *CURRENT	Required, Positional 1
RTNTEXT	CL var for RTNTEXT	<i>Character value</i>	Required, Positional 2

[Top](#)

User (USER)

Specifies the user whose user print information is being retrieved.

This is a required parameter.

*CURRENT

The user profile under which the current job is running is used.

name Specify the name of the user whose user print information is being retrieved.

[Top](#)

CL var for RTNTEXT (RTNTEXT)

Specifies a 10-character CL variable used to retrieve the user print information of the user profile for which information is requested.

This is a required parameter.

[Top](#)

Examples

```
RTVUSRPTI  USER(FEIST)  RTNTEXT(&TEXT)
```

This command retrieves user print information for user profile FEIST and stores it in CL program variable &TEXT.

Top

Error messages

*ESCAPE Messages

CPF2204

User profile &1 not found.

CPF2217

Not authorized to user profile &1.

CPF2247

Internal security object not available. Reason code &1.

CPF34D5

CCSID translation error.

Top

Retrieve WSCST source (RTVWSCST)

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

Parameters
Examples
Error messages

The Retrieve Work Station Customizing Object (RTVWSCST) command allows the user to retrieve a system-supplied set of table attributes for a given device type, keyboard language type, and keyboard type or a given manufacturer, type, and model of an ASCII printer into a source physical file member.

Top

Parameters

Keyword	Description	Choices	Notes
DEVTYPE	Device type	*TRANSFORM, *CTXFORM, 3101, 3151, 3161, 3162, 3163, 3164, 3179, 3180, 3196, 3197, 3476, 3477, 3486, 3487, 4019, 4201, 4202, 4207, 4208, 4216, 4224, 4234, 5204, 5251, 5291, 5292, 6252, D220, T910, T925, T955, V100, V220, W30, W50, W60	Required, Positional 1
KBDTYPE	Keyboard language type	AGB, AGE, AGI, ALI, BGB, BLI, BLM, BRB, BRE, CAB, CAI, CAM, CLB, CSB, CYB, DMB, DME, DMI, FAB, FAE, FAI, FNB, FNE, FNI, FQB, FQI, GKB, GNB, HNB, ICB, ICE, ICI, INB, INI, ITB, ITE, IRB, ITI, JEB, JEL, JPB, KAB, MKB, NCB, NEB, NEE, NEI, NWB, NWE, NWI, PLB, PRB, PRE, PRI, RMB, ROB, RUB, SFI, SFM, SGI, SGM, SKB, SPB, SPE, SQB, SPI, SSB, SSE, SSI, SWB, SWE, SWI, THB, TKB, TRB, UKB, UKE, UKI, USB, USE, USI, YGI	Optional, Positional 2
MFRTPMDL	Manufacturer type and model	<i>Character value</i>	Optional
SRCMBR	Source member	<i>Name</i>	Optional, Positional 3
KBD	Keyboard attached	*DATA5250, *TYPE5250, *DATA122, *TYPE122, *ENHANCED	Optional
SRCFILE	Source file	<i>Qualified object name</i>	Optional
	Qualifier 1: Source file	<i>Name</i> , <u>QTXTSRC</u>	
	Qualifier 2: Library	<i>Name</i> , *CURLIB, *LIBL	
TEXT	Text 'description'	<i>Character value</i> , *BLANK	Optional

Top

Device type (DEVTYPE)

Specifies the device type.

The possible values are:

*TRANSFORM

The SCS-to-ASCII host print transform function support is used by the ASCII printer.

*CTXFORM

IBM Transform Services configuration support.

device-type

Specify the device type to be used. See the Workstation Customization Programming book for a list of allowed device types.

Top

Keyboard language type (KBDTYPE)

Specifies the 3-character keyboard language identifier (used for EBCDIC and ASCII) for this display station.

The possible value is:

keyboard-language-type

Specify the 3-character country or region identifier (used for EBCDIC and ASCII) for this display station.

See the **Keyboard language type** prompt (KBDTYPE parameter) on the Change Device Description (Display) (CHGDEV DSP) or the Create Device Description (Display) (CRTDEV DSP) in the CL Reference for a list of the valid identifiers and the language the identifier represents. The ASCII device groups (if applicable) are also shown for each language.

Top

Manufacturer type and model (MFRTYPMDL)

Specifies the manufacturer, type, and model for an ASCII printer using host print transform function support or for Transform Services configuration support. See the Create Device Description (Printer) (CRTDEVPRT) command in the CL Reference for a list of the supported manufacturers, types, and models for ASCII printers using host print transform function support or Transform Services configuration support.

Top

Source member (SRCMBR)

Specifies the name of the source file member to receive the retrieved table attributes.

Top

Keyboard attached (KBD)

Specifies the keyboard type.

The possible values are:

***DATA5250**

A 5250 data entry keyboard is specified.

***TYPE5250**

A 5250 typewriter keyboard is specified.

***DATA122**

A 122 key data entry keyboard is specified.

***TYPE122**

A 122 key typewriter keyboard is specified.

***ENHANCED**

An enhanced keyboard is specified.

Top

Source file (SRCFILE)

Specifies the name of the source file in which a member is created to contain the retrieved table attributes. If the source file does not exist, it is created. The coded character set identifier for the source file is *HEX.

The possible library values are:

***LIBL** The library list is used to locate the source file.

***CURLIB**

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

library-name

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

The possible values are:

QXTSRC

The IBM-supplied source file QXTSRC is used.

source-file-name

Specify the name of the source file.

Top

Text 'description' (TEXT)

Specifies the description of the created source physical file member.

The possible values are:

***BLANK**

Text is not specified.

'description'

Specify a description for the source physical file member.

Top

Examples

```
RTVWSCST  DEVTYP(5251)  KBDTYPE(USB)  SRCMBR(MYSOURCE)
          KBD(*DATA5250)  SRCFILE(MYLIB/QXTSRC)
```

This command retrieves the system mapping tables for a 5251 twinaxial display with a 5250 data entry type keyboard attached using the U.S. basic language. The tables are stored in source member MYSOURCE in source file QXTSRC in library MYLIB.

Top

Error messages

*ESCAPE Messages

CPF5D33

Request for system table failed. Reason code &4.

[Top](#)

Run Backup (RUNBCKUP)

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

Parameters
Examples
Error messages

The Run Backup (RUNBCKUP) command allows the user to run a predefined backup of specified objects to tape. The backup may include libraries (all user libraries or those selected in the backup list), folders (all folders or root folders selected in the backup list), security data, configuration data, mail, and calendars.

Top

Parameters

Keyword	Description	Choices	Notes
BCKUPOPT	Backup options	*DAILY, *WEEKLY, *MONTHLY	Required, Key, Positional 1
DEV	Device	Single values: *BCKUPOPT Other values (up to 4 repetitions): <i>Name</i>	Optional, Positional 2

Top

Backup options (BCKUPOPT)

Specifies the backup options to use.

This is a required parameter.

*DAILY

The daily backup options are used.

*WEEKLY

The weekly backup options are used.

*MONTHLY

The monthly backup options are used.

Top

Device (DEV)

Specifies a list of tape devices to use for the backup.

Single values

*BCKUPOPT

The tape device names stored in the specified options are used for the backup.

Other values (up to 4 repetitions)

name Specify a list of tape devices used for the backup. If you are using a virtual tape device you can only specify one device name.

Examples

Example 1: Running a Daily Backup

```
RUNBCKUP  BCKUPOPT(*DAILY)
```

This command runs the daily backup using the devices specified in the options.

Example 2: Running a Monthly Backup

```
RUNBCKUP  BCKUPOPT(*MONTHLY)  DEV(TAP02)
```

This command runs the monthly backup using device TAP02 instead of those specified in the options.

Top

Error messages

*ESCAPE Messages

CPF1EE3

Not authorized to backup options.

CPF1EE4

Not authorized to run backup.

CPF1EE6

Devices specified cannot be used.

CPF1EE7

Unexpected error occurred during backup.

CPF1EE8

Unexpected error from exit program &2/&1.

CPF1E6C

Backup options in use.

CPF1E6E

Nothing selected for backup.

CPF1E63

Error occurred while checking backup authority.

CPF1E67

Backup options and library backup list damaged.

CPF1E68

Backup incomplete.

CPF1E99

Unexpected error occurred.

Top

Run DNS Update (RUNDNSUPD)

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

Parameters
Examples
Error messages

The Run DNS Update (RUNDNSUPD) command, or its alias NSUPDATE, is used to submit Dynamic Updates requests to a Domain Name System (DNS) server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via RUNDNSUPD or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with RUNDNSUPD have to be in the same zone. Requests are sent to the zones master server. This is identified by the MNAME field of the zones SOA record.

This utility will be run in interactive mode if *NONE is specified for the **Batch input file (BCHFILE)** parameter. To run this utility in non-interactive mode, specify a file name for the BCHFILE parameter.

Restrictions:

- You must have execute (*X) authority to the directories in the path of the batch input file.
- You must have read (*R) authority to the batch input file.
- You must have execute (*X) authority to the directories in the path of the key file.
- You must have read (*R) authority to the key file.
- You must have execute (*X) authority to the directories in the path of the output file.
- You must have write (*W) authority to the output file if it already exists.
- You must have read, write and execute (*RWX) authority to the output file's parent directory if the output file does not already exist.

Top

Parameters

Keyword	Description	Choices	Notes
BCHFILE	Batch input file	<i>Path name</i> , *NONE	Optional, Positional 1
DEBUG	Show debug information	*NO, *YES	Optional, Positional 2
KEYNAME	Key name	<i>Character value</i> , *NONE	Optional
KEYFILE	Key file	<i>Path name</i> , *NONE	Optional
TIMEOUT	Update timeout	1-300, <u>300</u> , *NOMAX	Optional
UDPRTYITV	UDP timeout	1-65535, <u>3</u> , *CALC	Optional
UDPNBRRTY	UDP retry	0-100, <u>3</u>	Optional
PROTOCOL	Network protocol	*UDP, *TCP	Optional
TOSTMF	Output file	<i>Path name</i> , *STDOUT	Optional

Batch input file (BCHFILE)

Specifies a stream file with all update commands to be used as batch input.

Each command in the file is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates will be rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line (or the send command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats are as follows. Lines in the file beginning with a semicolon are considered comments and are ignored:

```
server {servername} [port]
local {address} [port]
zone {zonename}
class {classname}
key {name} {secret}
prereq nxdomain {domain-name}
prereq yxdomain {domain-name}
prereq nxrrset {domain-name} [class] {type}
prereq yxrrset {domain-name} [class] {type}
prereq yxrrset {domain-name} [class] {type} {data...}
update delete {domain-name} [ttl] [class] [type [data...]]
update add {domain-name} [ttl] [class] {type} {data...}
show
send
answer
quit
```

See the examples for descriptions of these commands.

*NONE

Do not use a batch file. This will enter interactive mode. To return from interactive mode, you must enter the command 'quit'.

Note: This value is only valid if the command is run in an interactive job.

path-name

Specify the path for a stream file from which input is read. You must have read (*R) authority to this file.

Show debug information (DEBUG)

Specifies whether or not to turn debugging mode on. More information is displayed about the packet sent to the server and the resulting answer when debugging mode is on.

*NO Turn off debugging messages.

***YES** Turn on debugging messages.

Top

Key name (KEYNAME)

Specifies a Transaction Signature (TSIG) key to sign the DNS queries. The only message digest algorithm currently used for TSIG is HMAC-MD5, although others may be added later. The TSIG key is a base-64 encoded string, typically generated by the Generate DNS Key (GENDNSKEY) command. The DNS server that is queried needs to be configured with the TSIG key and algorithm that is being used or the transaction will fail. See RFC 2845 for TSIG.

*NONE

Do not specify a key name.

character-value

Specify the Transaction Signature key. The format is 'name:key' where name is the key name, and key is the actual key as a base-64 encoded string. Example:

```
my-tsig-key:JNvcpxysbJ2hsd0qQ5qrQ==
```

The key name in this case is 'my-tsig-key' and the base-64 encoded key is 'JNvcpxysbJ2hsd0qQ5qrQ=='.

The DNS server being queried needs to include this key and algorithm in its configuration in order to accept this TSIG key from clients.

Top

Key file (KEYFILE)

Specifies a file containing a SIG(0) key used to authenticate Dynamic DNS update requests. In this case, the key specified is not an HMAC-MD5 key. SIG(0) uses public key cryptography. To use a SIG(0) key, the public key must be stored in a KEY record in a zone served by the name server. See RFC 3535 and RFC 2931 for SIG(0).

Note: If a value other than *NONE is specified for this parameter, do not specify a value for the KEYNAME parameter.

*NONE

Do not specify a key file.

path-name

Specify the path for a stream file containing key information. For example, '/home/myprofile/my-key-file' could be specified.

Top

Update timeout (TIMEOUT)

Specifies the maximum time an update request can take before it is aborted.

300 The default timeout is 300 seconds.

***NOMAX**

The update request timeout is disabled.

1-300 Specify a valid timeout value in seconds.

Top

UDP timeout (UDPRTYITV)

Specifies the interval, in seconds, between UDP retries.

3 A retry interval of 3 seconds is used.

*CALC

The retry interval is calculated by the utility based on the values specified for the **Update timeout (TIMEOUT)** parameter and the **UDP retry (UDPNBRRTY)** parameter.

1-65535

Specify the number of seconds to use between UDP retries.

Top

UDP retry (UDPNBRRTY)

Specifies the number of UDP retries.

3 Three UDP retries will be used.

0-100 Specify the number of UDP retries. If zero is specified, only one update request will be made.

Top

Network protocol (PROTOCOL)

Specified whether to use TCP or UDP when sending requests to the server.

*UDP Use UDP to send the query. However, TCP will be automatically selected for queries that require it, such as zone transfer (AXFR) requests.

*TCP Use TCP to send the query.

Top

Output file (TOSTMF)

Specifies the name of a stream file where all command output is written.

*STDOUT

All command output goes to the standard output device (normally the display).

path-name

Specify the path for a stream file where output should be written.

Top

Examples

The command formats and their meaning are as follows:

server {servername} [port]

Sends all dynamic update requests to the DNS server `servername`. When no server statement is provided, RUNDNSUPD will send updates to the master server of the correct zone. The MNAME field of that zone's SOA record will identify the master server for that zone. `port` is the port number on `servername` where the dynamic update requests get sent. If no port number is specified, the default DNS port number of 53 is used.

local {address} [port]

Sends all dynamic update requests using the local address. When no local statement is provided, RUNDNSUPD will send updates using an address and port chosen by the system. `port` can additionally be used to make requests come from a specific port. If no port number is specified, the system will assign one.

zone {zonename}

Specifies that all updates are to be made to the zone `zonename`. If no zone statement is provided, RUNDNSUPD will attempt to determine the correct zone to update based on the rest of the input.

class {classname}

Specify the default class. If no class is specified the default class is IN.

key {name} {secret}

Specifies that all updates are to be TSIG signed using the keyname/keysecret pair. The `key` command overrides any key specified on the command line via `-y` or `-k`.

prereq nxdomain {domain-name}

Requires that no resource record of any type exists with name `domain-name`.

prereq yxdomain {domain-name}

Requires that `domain-name` exists (has at least one resource record, of any type).

prereq nxrrset {domain-name} [class] {type}

Requires that no resource record exists of the specified type, class and `domain-name`. If class is omitted, IN (internet) is assumed.

prereq yxrrset {domain-name} [class] {type}

This requires that a resource record of the specified type, class and `domain-name` must exist. If class is omitted, IN (internet) is assumed.

prereq yxrrset {domain-name} [class] {type} {data...}

The data from each set of prerequisites of this form sharing a common type, class, and `domain-name` are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone at the given type, class, and `domain-name`. The data are written in the standard text representation of the resource records RDATA.

update delete {domain-name} [ttl] [class] {type} [data...]

Deletes any resource records named `domain-name`. If type and data is provided, only matching resource records will be removed. The internet class is assumed if class is not supplied. The `ttl` is ignored, and is only allowed for compatibility.

update add {domain-name} [ttl] [class] {type} {data...}

Adds a new resource record with the specified `ttl`, class and data.

show Displays the current message, containing all of the prerequisites and updates specified since the last send.

send Sends the current message. This is equivalent to entering a blank line.

answer

Displays the answer.

quit Exit interactive mode.

Example 1: Update Example

```

RUNDNSUPD  BCHFILE(*NONE)
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
> quit

```

This command will start the interactive mode for the update utility. The lines that have the '>' prefix are interactive commands.

Insert and delete resource records from the 'example.com' zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master DNS server for example.com.

Any A records for oldhost.example.com are deleted and an A record for newhost.example.com with IP address 172.16.1.1 is added. The newly-added record has a 1 day TTL (86400 seconds)

Example 2: Pre-requisite Example

```

RUNDNSUPD
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
> quit

```

The prerequisite condition gets the DNS server to check that there are no resource records of any type for nickname.example.com. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in RFC1034 that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC2535 to allow CNAMEs to have RRSIG, DNSKEY and NSEC records.)

Example 3: Batch File Example

```

RUNDNSUPD  BCHFILE('/home/ibmuser/my-updates')

```

This command sends the updates in the file '/home/ibmuser/my-updates' to the server defined in the files **server** command. For example, the contents of the file might look like this:

```

server myserver.i5os.ibm.com 53
zone i5os.ibm.com
class in
prereq yxdomain box1.i5os.ibm.com.
update delete box1.i5os.ibm.com. A
update add box1.i5os.ibm.com. 3600 A 10.9.9.9
prereq yxdomain box2.i5os.ibm.com.
update delete box2.i5os.ibm.com. A
update add box2.i5os.ibm.com. 3600 A 10.9.9.10

```

And the results displayed on the screen might look like this:

```

; TSIG error with server: tsig indicates error
update failed: NOTAUTH(BADKEY)

```

Or the results displayed on the screen might look like this:

```
; TSIG error with server: tsig indicates error  
update failed: REFUSED
```

The first example represent the case when the key in the file is not valid. The second example represents the case when the server is not allowing dynamic updates from you. No output or blank output only means the update was successful.

[Top](#)

Error messages

*ESCAPE Messages

DNS0013

Error processing command parameters.

DNS0065

Option 33 of i5/OS is required, but is not installed.

TCP7124

Program &1 in library &2 type *PGM ended abnormally.

[Top](#)

Run LPDA-2 (RUNLPDA)

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

Parameters
Examples
Error messages

The Run LPDA-2 (RUNLPDA) command allows you to run a Link Problem Determination Aid-2 (LPDA-2) operational command on local or remote data circuit-terminating equipment (DCE). The RUNLPDA command can be used to:

- Establish or disconnect a switched telephone network connection.
- Open or close the relay contact in a coupler.
- Determine whether a relay contact is open or closed.
- Determine whether electric current is flowing through an internal sensor.
- Change the transmit speed of a DCE to full or backup.

The result of the RUNLPDA command is returned as a message.

Restrictions:

1. The RUNLPDA command is valid only for an analog LPDA-2 DCE attached to a nonswitched SDLC line.
2. This command is shipped with public *EXCLUDE authority and the QPGMR, QSYSOPR, QSRV, or QSRVBAS, user profiles have private authorities to use the command.

Top

Parameters

Keyword	Description	Choices	Notes
LINE	Line	<i>Name</i>	Required, Positional 1
OPTION	Option	*CALL, *DSC, *SETSPEED, *CONTSENSE, *CONTOPER	Required, Positional 2
LCLDCEADR	Local DCE address	01-FB, * <u>LCL</u>	Optional
RMTDCEADR	Remote DCE address	01-FB, * <u>NONE</u> , *ANY	Optional
TELNBR	Telephone number	Single values: * <u>STORED</u> Other values: <i>Element list</i>	Optional
	Element 1: Number 1	<i>Character value</i>	
	Element 2: Number 2	<i>Character value</i>	
CONTACT	Relay contact	*OPEN, *CLOSE	Optional
SPEED	Transmit speed	*FULL, *BACKUP	Optional
DTEPORT	DTE port	* <u>ALL</u> , A, B, C, D	Optional
DCERTY	DCE retry	* <u>NO</u> , *YES	Optional

Top

Line (LINE)

Specifies the name of the nonswitched SDLC line that is attached to the analog DCE on which the LPDA-2 operational command is to be run. If *CALL is specified on the **Option** prompt (OPTION parameter), the line must be varied on but not active. If *CONTOPER, *CONTSENSE, *DSC, or *SETSPEED is specified on the OPTION parameter, the line must be either varied on or active.

This is a required parameter.

Top

Option (OPTION)

Specifies which LPDA-2 operational command is run.

This is a required parameter.

The contact operate, contact sense, and set transmit speed commands can be run on a local DCE or a remote DCE.

To run one of these commands on a local DCE:

- On the **Local DCE address** prompt (LCLDCEADR parameter), specify the address of the local DCE.
- On the **Remote DCE address** prompt (RMTDCEADR parameter), specify *NONE.

To run one of these commands on a remote DCE:

- On the LCLDCEADR parameter, specify the address of the local DCE to which the remote DCE is connected.
- On the RMTDCEADR parameter, specify the address of the remote DCE.

Note: If the local DCE is configured as point-to-point secondary or multipoint tributary, LPDA-2 commands are not sent to the remote DCE.

The possible values are:

*CALL

The call out command is run. This command establishes a connection between a local and a remote DCE over a switched telephone network. This value is valid only if:

- Both the local DCE and the remote DCE have two-wire couplers installed or both the local DCE and the remote DCE have four-wire couplers installed.
- The line specified on the **Line** prompt (LINE parameter) is varied on, but not active.

*CONTOPER

The contact operate command is run. This command opens or closes the relay contact in the coupler, depending on the value specified on the **Relay contact** prompt (CONTACT parameter). This value is valid only if a two-wire coupler is installed in the DCE on which this LPDA-2 command is run.

*CONTSENSE

The contact sense command is run. This command reports whether the relay contact in the coupler is open or closed and whether electric current is flowing through the internal sensor. This option is valid only if a two-wire coupler is installed in the DCE on which this command is run.

***DSC** The disconnect command is run. This command disconnects the switched telephone network connection between the local DCE and the remote DCE. If you specify this value, you cannot specify *NONE on the **Remote DCE address** prompt (RMTDCEADR parameter).

***SETSPEED**

The set transmit speed command is run. This command changes the transmit speed of the DCE to full or backup, depending on the value specified on the **Transmit speed** prompt (SPEED parameter). For multiple port DCE configurations in which the data terminal equipment (DTE) ports can be set to different speeds, use the **DTE port** prompt (DTEPORT parameter) to specify the port.

Note: The set transmit speed command may not change the transmit speed, depending on the configuration options selected for the DCE. Refer to the DCE documentation for more information.

Top

Local DCE address (LCLDCEADR)

Specifies the hexadecimal address of the local DCE. Refer to the DCE documentation for more information on addressing.

The possible values are:

***LCL** X'01' is used for the address.

local-DCE-address

Specify the address of the local DCE. Valid values range from X'01' through X'FB'.

Top

Remote DCE address (RMTDCEADR)

Specifies the hexadecimal address of the remote DCE on which the LPDA-2 operational command is to be run.

The possible values are:

***NONE**

X'00' is used as the address, which indicates that the LPDA-2 command is to be run on the local DCE. You cannot specify *NONE on this parameter if you specify *DSC on the **Option** prompt (OPTION parameter).

***ANY** X'FD' is used for the address. Specify this value in the following situations:

- The LPDA-2 command is to be run on any remote DCE connected to the local DCE.
- You do not know the remote DCE address on a point-to-point line.
- To run the LPDA-2 operational command on all tributary DCEs on a multipoint line. In this case, no detailed response is received.

remote-DCE-address

Specify the address of the remote DCE. Valid values range from X'01' through X'FB'.

Top

Telephone number (TELNBR)

Specifies the telephone number or telephone numbers that the local DCE dials to establish a connection to the remote DCE. This parameter is valid only if *CALL is specified on the **Option** prompt (OPTION parameter).

The possible values are:

***STORED**

The telephone number or numbers stored in the local DCE when the DCE was configured are used.

telephone-number

Specify one telephone number if a two-wire coupler is installed in the local DCE. Specify two telephone numbers if a four-wire coupler is installed in the local DCE. Only numeric characters are processed by the LPDA-2 command, but you can also enter alphabetic characters or any other non-DBCS characters to improve readability. A comma (,) can be used to instruct the DCE to pause during dialing.

Top

Relay contact (CONTACT)

Specifies whether to open or close the relay contact in a two-wire coupler. This parameter is required if *CONTOPER is specified on the **Option** prompt (OPTION parameter).

The possible values are:

***OPEN**

The relay contact is opened.

***CLOSE**

The relay contact is closed.

Top

Transmit speed (SPEED)

Specifies desired transmit speed of the DCE. This parameter is required if *SETSPEED is specified on the **Option** prompt (OPTION parameter).

The possible values are:

***FULL** The transmit speed is set to full.

***BACKUP**

The transmit speed is set to backup.

Top

DTE port (DTEPORT)

Specifies the DTE port on the local or remote DCE for which the transmit speed is changed. This parameter is valid only if *SETSPEED is specified on the **Option** prompt (OPTION parameter) and is applicable only to multiple port DCEs that do not use the multiple address configuration option.

The possible values are:

***ALL** The aggregate speed of the DCE is changed. Refer to the DCE documentation for information on the effect of changing the aggregate speed on the transmit speed of individual ports.

A The transmit speed of the A-port is changed.

B The transmit speed of the B-port is changed.

C The transmit speed of the C-port is changed.

D The transmit speed of the D-port is changed.

Top

DCE retry (DCERTY)

Specifies whether the local DCE resends the LPDA-2 command to the remote DCE if no response is received from the remote DCE. No retry can be attempted if *CALL or *DSC is specified on the **Option** prompt (OPTION parameter).

The possible values are:

***NO** No retry is attempted.

***YES** One retry is attempted.

Top

Examples

Example 1: Establishing a Switched Telephone Network Connection

```
RUNLPDA  LINE(SDLCLINE) OPTION(*CALL) LCLDCEADR(*LCL)
          RMTDCEADR(*NONE) TELNBR(*STORED)
```

This command runs the call out command. The local DCE with address X'01' (*LCL) on line SDLCLINE dials the telephone numbers that are stored in the local DCE.

Example 2: Establishing a Switched Telephone Network Connection

```
RUNLPDA  LINE(SDLCLINE) OPTION(*CALL)
          LCLDCEADR(*LCL) RMTDCEADR(*NONE)
          TELNBR('9, 1-507-555-1212' '9, 1 (507) 555-1313')
```

This command runs the call out command. The local DCE dials the two numbers specified on the TELNBR parameter. The comma (,) indicates a pause during dialing. Other non-numeric characters are ignored, but are allowed for easier reading.

Example 3: Disconnecting a Switched Telephone Network Connection

```
RUNLPDA  LINE(SDLCLINE) OPTION(*DSC)
          LCLDCEADR(10) RMTDCEADR(*ANY)
```

This command runs the disconnect command. The local DCE with address X'10' disconnects from the switched telephone network.

Example 4: Closing the Relay Contact in the Local DCE

```
RUNLPDA  LINE(SDLCLINE) OPTION(*CONTOPER) LCLDCEADR(02)
          RMTDCEADR(*NONE) CONTACT(*CLOSE)
```

This command runs the contact operate command. The local DCE with address X'02' closes the relay contact in its two-wire coupler.

Example 5: Reporting the Status of the Relay Contact

```
RUNLPDA  LINE(SDLCLINE) OPTION(*CONSENSE)  
         LCLDCEADR(01) RMTDCEADR(04)
```

This command runs the contact sense command. A message reports the status of the relay contact in the remote DCE with address X'04'. (The correct local DCE address must be specified on the LCLDCEADR parameter.)

Example 6: Changing the Transmit Speed

```
RUNLPDA  LINE(SDLCLINE) OPTION(*SETSPEED) LCLDCEADR(05)  
         RMTDCEADR(*NONE) SPEED(*BACKUP) DTEPORT(B)
```

This command runs the set transmit speed command. The transmit speed for Port B of the local DCE with address X'05' is changed to backup speed.

Top

Error messages

*ESCAPE Messages

CPF1BAA

LPDA-2 command cannot run in switched network backup.

CPF1BAB

LPDA-2 command not supported by target DCE.

CPF1BAC

Required feature not installed.

CPF1BAD

LPDA-2 command not compatible with DCE configuration.

CPF1BA1

Line description &1 does not exist.

CPF1BA2

Line &1 not active or not varied on.

CPF1BA3

Line &1 is active secondary.

CPF1BA4

Line &1 failed while processing RUNLPDA.

CPF1BA5

No response received for LPDA-2 command.

CPF1BA6

Cannot run LPDA-2 command on line &1.

CPF1BA7

No buffers available for RUNLPDA data.

CPF1BA8
LPDA-2 command already active on line &1.

CPF1BBA
Continuous answer tone received for first call.

CPF1BBB
Continuous answer tone received for second call.

CPF1BBC
No line signal on switched connection.

CPF1BBD
No initial dial tone.

CPF1BBE
No dial tone after pause.

CPF1BBF
Busy tone received from remote DCE.

CPF1BB0
Coupler not installed in local DCE.

CPF1BB1
Coupler not operational.

CPF1BB2
LPDA-2 command not valid with configuration options.

CPF1BB3
Telephone number not valid.

CPF1BB5
Switched connection already active.

CPF1BB6
Only single telephone number allowed.

CPF1BB7
Two telephone numbers required.

CPF1BB8
No answer from remote DCE on first call.

CPF1BB9
No answer from remote DCE on second call.

CPF1BDA
Error occurred processing RUNLPDA command.

CPF1BDB
Internal processing error.

CPF1BDC
Required feature not operational.

CPF1BDD
Ring back limit exceeded.

CPF1BDE
DCE busy. LPDA-2 command cannot be run.

CPF1BDF
Line not in proper state for LPDA-2 call out command.

CPF1BD3
DCE currently processing previous LPDA-2 command.

CPF1BD5
Cannot send command to remote DCE.

CPF1BD6
Line description &1 not nonswitched *SDLC.

CPF1BD8
Request failed. System service tools active.

CPF1BD9
RUNLPDA command does not support switched lines.

CPF1B85
Two-wire coupler not installed.

CPF1B86
Coupler not installed.

CPF1B87
Transmit speed cannot be changed.

CPF1B88
DTE port selected not valid for configuration.

CPF1B94
Local DCE received no response from remote DCE.

CPF1B95
Not authorized to line description &1.

CPF1B97
Format of response received not valid.

CPF1B98
Received response with bad frame check sequence.

OFC8DB7
Internal processing failure.

Top

Run Query (RUNQRY)

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

Parameters
Examples
Error messages

The Run Query (RUNQRY) command runs an existing query or a default query if only a file name is specified for this command. The query gets information from the system database and produces a report of that information. The report is created in either detailed or summary form. The definition of the query can be printed when output to a printer or database file is specified. The output is shown, printed, or stored in a database file. The command is used in three ways: to run an existing query (one that has already been created), to run an existing query with some of its values changed by values specified on this command, or to run a default query based only on the defaults and values specified in this command.

- To run an existing query without changing the file or files to query, use the QRY parameter (without the QRYFILE parameter) to specify the name of the query.
- To run a changed version of an existing query, use the QRY parameter and the appropriate parameters to change the definition as desired. The parameter values specified on this command override the corresponding values in the existing query definition, but only when the command is processing. For example, use the QRYFILE parameter to indicate a different file or list of files to use in the query.
- To query a file without a previously defined query definition, use the QRYFILE parameter to specify which file to query. Only one file name can be specified for a default query.

If you specify both the QRY and QRYFILE parameters, the files specified in the QRYFILE parameter override the file names specified in the query. Therefore, if multiple files (and members) are defined in the query definition and you want to change one or two of them, specify *SAME for the file selections that do not change, and specify the values for the files to override.

Notes:

1. When a changed version of an existing query is run, the changes specified on the RUNQRY command do **not** change any of the values in the query definition itself; they affect only the results of the report being run.
2. If an existing query (identified on the QRY parameter) is used, *RUNOPT is the default value for any unspecified parameters; that is, the same value specified (or assumed) in the definition of the query is used as the default. If this is a default query, the default value is not *RUNOPT, but is the next predefined value listed in the syntax diagram.

Top

Parameters

Keyword	Description	Choices	Notes
QRY	Query	Single values: *NONE Other values: <i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Query	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	

Keyword	Description	Choices	Notes
QRYFILE	Query file	Values (up to 32 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i> , *SAME	
	Qualifier 2: Library	<i>Name</i> , *RUNOPT, *LIBL, *CURLIB	
	Element 2: Member	<i>Name</i> , *RUNOPT, *FIRST, *LAST, *ALL	
OUTTYPE	Report output type	*RUNOPT, *DISPLAY, *PRINTER, *OUTFILE	Optional, Positional 3
OUTFORM	Output form	*RUNOPT, *DETAIL, *SUMMARY	Optional
RCDSLIT	Record selection	*NO, *YES	Optional
PRTDFN	Print query definition	*RUNOPT, *NO, *YES	Optional
PRTDEV	Print device	<i>Name</i> , *RUNOPT, *PRINT	Optional
FORMSIZE	Form size	<i>Element list</i>	Optional
	Element 1: Form length	1-255, *RUNOPT	
	Element 2: Form width	1-378, *RUNOPT	
FORMTYPE	Form type	<i>Character value</i> , *RUNOPT, *STD	Optional
COPIES	Copies	1-255, *RUNOPT	Optional
LINESPACE	Line spacing	*RUNOPT, 1, 2, 3	Optional
OUTFILE	Output file	<i>Element list</i>	Optional
	Element 1: File	<i>Qualified object name</i>	
	Qualifier 1: File	<i>Name</i> , *RUNOPT	
	Qualifier 2: Library	<i>Name</i> , *RUNOPT, *CURLIB	
	Element 2: Member	<i>Name</i> , *RUNOPT, *FIRST, *LAST, *ALL	
	Element 3: Option	*RUNOPT, *NEWFILE, *RPLFILE, *NEWMBR, *RPLMBR, *ADDMBR	
AUT	Authority	<i>Name</i> , *RUNOPT, *LIBCRTAUT, *USE, *CHANGE, *ALL, *EXCLUDE	Optional

Top

Query (QRY Parameter)

Specifies the name of an existing query to be run. If QRY is not specified, QRYFILE must be specified.

The possible values are:

*NONE

No existing query definition is used. Instead, a default query (or quick query) is used to get information from the file specified on the QRYFILE parameter.

query-name

Specify the name of the query to run.

If no value is specified, the file or files that were specified when the query was defined are the files to be queried.

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

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

*CURLIB

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

library-name

Specify the library where the query is located.

Top

Query file (QRYFILE Parameter)

Specifies the database file or files to be queried for information. If the QRY parameter is specified, as many as 32 files can be specified on this parameter by using the file names and/or using the default value *SAME for one or more of the files. If the QRY parameter is not specified, only one file name can be specified on this parameter. If QRYFILE is not specified, QRY must be specified.

If no value is specified, the file or files that were specified when the query was defined are the files used to run the query.

You can enter multiple values for this parameter.

There are two parts to this parameter.

Element 1: Database File Name

The possible values are:

data-base-file-name

Specify the names of one or more database files that contain the data from which the system gets information to produce the output. Up to 32 files can be specified.

***SAME**

The list of files remains the same as defined in the query definition.

The name of a database file can be qualified by one of the following library values:

***SAME**

The value of the library, file, and member do not change for the file in this position within the list. The values are the same as defined in the query. Any values specified for the library, file, and member are ignored.

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

***CURLIB**

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

***RUNOPT**

The library specified for the file selection is used.

library-name

Specify the library where the database file is located.

Element 2: Database File Member

The possible values for member name are:

member-name

Specify the name of the file member to be queried.

***RUNOPT**

The member specified for this file selection is used.

***FIRST**

The first member in the file is the member to be queried.

***LAST**

The last member in the file is the member to be queried.

***ALL** The input file is a partitioned table, where all of the members in the file are to be queried.

Top

Report output type (OUTTYPE Parameter)

Specifies where the report or output produced by the query is sent. If a value is not specified in the query, and is not entered on the command, or if a query name is not specified, *DISPLAY is assumed.

The possible values are:

***RUNOPT**

If a query definition is being used, the type of output specified in the query definition is the type of output produced when this query is run.

***DISPLAY**

The output produced by the query is sent to the display station that runs the command. If run in batch, the output is sent to the printer instead of the display.

***PRINTER**

The output produced by the query is printed.

***OUTFILE**

The output is directed to the database file specified on the OUTFILE parameter.

Top

Output form (OUTFORM Parameter)

Specifies the form of output produced by the query. If no value was specified in the query and no value was entered on the command, or if a query name is not specified, *DETAIL is assumed.

The possible values are:

***RUNOPT**

If a query definition is used, the output form specified in the query definition is used when this query is run.

***DETAIL**

The output form produced by the query is a report containing detail records and summary records if any exist.

***SUMMARY**

The output form produced by the query is a report containing summary records only.

Top

Record selection (RCDSLT Parameter)

Specifies whether or not the query is run with a run time selection test.

Note: The Query for i5/OS licensed program must be installed and the query must be run interactively to specify *YES.

The possible values are:

***NO** The query is run without showing the record selection display in Query.

***YES** Record selection definition is allowed for this run only. A display is shown on which you can change the record selection tests defined in the query or specify record selection tests if a query name was not specified.

Note: For output to display, query can be run repeatedly.

Top

Print query definition (PRTDFN Parameter)

Specifies whether or not the query definition is printed with the report when the query is run. The definition can be printed when the output of the query is printed or is stored in a database file, as determined by the OUTTYPE parameter. If a value is not specified in the query, or in this parameter, or if a query name is not specified, the value *NO is assumed.

The possible values are:

***RUNOPT**

If a query definition is being used when the query is run, the print option specified in the query definition is used.

***NO** The query definition is not printed when the query is run.

***YES** The query definition is printed in the report. *YES cannot be specified if OUTTYPE(*DISPLAY) is specified or assumed.

Top

Print device (PRTDEV Parameter)

Specifies the printer device on which the report is printed. If no value is specified, the printer that was specified when the query was defined is assumed. If no printer is specified in the query or in this parameter, or if a query name is not specified, the value *PRINT is assumed.

Note: If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

***RUNOPT**

If a query definition is being used, the printer specified in the query definition is used to print the output when this query is run.

***PRINT**

The default printer, as defined by QPQUPRFIL, is used to print the output when this query is run.

printer-device-name

Specify the name of the printer that is used to print the output when this query is run.

Top

Form size (FORMSIZE Parameter)

Specifies the length and the width of the forms on which the report is printed. If a form width greater than 132 is specified, Query will open the file with CPI(15) and FONT(*CPI). If no value is specified in the query or in this parameter, or if a query name is not specified, 132 is the assumed form width, and the value from the file QPQUPRFIL is the assumed form length.

Note: If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

There are two parts to this parameter.

Element 1: Form Length

*RUNOPT

If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-length

Specify the form length used when this query is run. Valid values range from 1 through 255.

Element 2: Form Width

*RUNOPT

If a query definition is being used when the query is run, the form size specified in the query definition is used. If the form size specified in the query definition is blank, the value from QPQUPRFIL is assumed.

form-width

Specify the form width used when this query is run. Valid values range from 1 through 378.

Top

Form type (FORMTYPE Parameter)

Specifies the type of form on which the output is printed. The identifiers used to indicate the type of forms are user-defined and can be a maximum of 10 characters in length.

Note: If a value is not specified in the query or on this parameter, or if a query name is not specified, the value in QPQUPRFIL is assumed. If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

*RUNOPT

If a query definition is being used, the form type specified in the query definition is used when this query is run.

form-type

Specify the form type that is used when this query is run.

Top

Copies (COPIES Parameter)

Specifies the number of copies being printed.

Note: If a value is not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed number of copies. If an override is in effect for the printer file QPQUPRFIL, this parameter uses the value specified by the override.

The possible values are:

***RUNOPT**

If a query definition is being used, the number of copies specified in the query definition is used when this query is run. If the number of copies specified in the query definition is blank, the number of copies from QPQUPRFIL is assumed.

number-of-copies

Specify the number of copies to print when this query is run. Specify a number ranging from 1 through 255.

Top

Line spacing (LINESPACE Parameter)

Specifies the number of blank lines to leave between lines in the report. The numbers range from 1 through 3. If a value was not specified in the query or on this parameter, or if a query name is not specified, 1 is the assumed value.

The possible values are:

***RUNOPT**

If a query definition is being used, the number of lines specified in the query definition is used when this query is run.

- 1 Indicates that single spacing (no blank lines) is used when the query output is printed.
- 2 Indicates that double spacing (1 blank line) is used when the query output is printed.
- 3 Indicates that triple spacing (2 blank lines) is used when the query output is printed.

Top

Output file (OUTFILE Parameter)

Specifies the database file (if any) that receives the query output. If no value is specified for this parameter, the library, file, member, and option specified in the query are assumed. If a query is not specified, the file QQUERYOUT is created in the default output file library. This library is defined in the query profile or the current library (*CURLIB) if no profile exists. The first member (*FIRST) of this new file is used for the output.

Note: If you did not specify a current library, the QGPL library is used.

If the name specified by the OUTFILE parameter does not exist, the system creates it in the specified library.

There are three parts to this parameter.

Element 1: Database File Name The possible values are:

data-base-file-name

Specify the name of the database file that receives the output of the query.

***RUNOPT**

The database file specified in the query is used to receive the output of the query. The file, library, member, and option do not change. Any values specified for the library, member, and option are ignored.

The name of the database file can be qualified by one of the following library values:

***RUNOPT**

If specified in the query definition, the output is directed to the library named in the query definition.

***CURLIB**

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

library-name

Specify the name of the library to be used.

Element 2: Database File Member

The possible member values are:

***FIRST**

The first member in the file is used to receive the query output.

***LAST**

The last member in the file is used to receive the query output.

***RUNOPT**

The member specified in the query is used to receive the query output.

***ALL** The output file is a partitioned table, where all members in the file are used to receive the query output. When ***ALL** is specified for the member, the **Element 3 Data** option can only be set to ***RPLMBR** or ***ADDMBR**. The partitioned table must already exist when the query is run.

member-name

The named file member is used to receive the query output.

Element 3: Adding or Replacing Data

The last option specifies whether to put the data in a new database file, replace an existing database file, add a new member, replace an existing member, or add data to an existing member. If no value is specified in the query or in this parameter, or if a query name is not specified, the value ***NEWFILE** is assumed.

***RUNOPT**

If a query definition is used, the member option specified in the query definition is the type used when this query is run.

***NEWFILE**

The output is written to a new database file. This option is not valid when the member name is set to ***ALL**.

***RPLFILE**

The output deletes the old file and creates a new file. This option is not valid when the member name is set to ***ALL**.

***NEWMBR**

The output is added as a new member. This option is not valid when the member name is set to ***ALL**.

***RPLMBR**

The existing member is cleared and the output is then added.

***ADDMBR**

The output is added to the end of an existing member.

Top

Authority (AUT Parameter)

Specifies the authority given to users who do not have specific authority to the output file, who are not on an authorization list, and whose user group has no specific authority to the output file. More information on this parameter is in the CL Reference book, Appendix A.

Note: The authority is assigned only when the output file is created or replaced.

The possible values are:

***RUNOPT**

If specified in the query definition, the authority named in the query definition is used.

***LIBCRTAUT**

The public authority for the output file is taken from the value on the CRTAUT parameter of the target library (the library that is to contain the output file). The public authority is determined when the output file is created. If the CRTAUT value for the library changes after the output file is created, the new value does not affect any existing objects.

***CHANGE**

You can perform all operations on the output file except those limited to the owner or controlled by object existence authority and object management authority. You can change and perform basic functions on the output file. Change authority provides object operational authority and all data authority.

***USE** You can perform basic operations on the output file, such as running a program or reading a file. You cannot change the output file. *USE authority provides object operational authority, read authority, and execute authority.

***ALL** You perform all operations on the output file except those limited to the owner or controlled by authorization list management authority.

***EXCLUDE**

You cannot access the output file.

authorization-list-name

Specify the name of the authorization list used.

Top

Examples

Example 1: Printing Summary Records Only

```
RUNQRY QRY(LIBX/QRY1) OUTTYPE(*PRINTER)
        OUTFORM(*SUMMARY) COPIES(4)
```

This command runs the query QRY1 located in library LIBX. The report that is produced and printed contains summary records only. Four copies of the report are printed.

Example 2: Running a Default Query

```
RUNQRY  QRYFILE((LIBX/FILE2 *FIRST))  
        OUTTYPE(*OUTFILE)  OUTFORM(*DETAIL)  
        RCDSL(*YES)  OUTFILE(LIB2/OUT1 MBR4 *NEWMBR)
```

This command runs a default query and gets the data from the first member of file FILE2 located in library LIBX. Member MBR4 is created as a new member to file OUT1 in library LIB2. Member MBR4 contains the output from the default query. The record selection display is shown to allow you to specify which records from file FILE2 in library LIBX are written to new member MBR4 in file OUT1 in library LIB2. The output contains detail records only.

[Top](#)

Error messages

None

[Top](#)

Run Remote Command (RUNRMTCMD)

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

Parameters
Examples
Error messages

The Run Remote Command (RUNRMTCMD) command, also known as AREXEC when an SNA address is specified for the remote location name, allows users to run a command on a remote system that is running the target portion of this function.

The target portion of this function can be:

- a REXECD (remote execution) daemon, if you specify *IP for the address type
- an AREXECD (APPC remote execution) daemon, if you specify *SNA for the address type.

When the command is sent to the remote system, the local system waits for the command to complete and the output from the remote command will be placed in a spooled file.

Top

Parameters

Keyword	Description	Choices	Notes
CMD	Command	Character value	Required, Positional 1
RMTLOCNAME	Remote location	Element list	Required, Positional 2
	Element 1: Name or address	Character value	
	Element 2: Type	*SNA, *IP	
RMTUSER	Remote user ID	Character value, *NONE, *CURRENT	Optional
RMPWD	Remote password	Character value, *NONE	Optional
MODE	Mode	Communications name, *NETATR	Optional
CCSID	Coded character set ID	1-65533, *CALC	Optional
WAITTIME	Wait time (in seconds)	2-3600, *NOMAX, *NOWAIT	Optional

Top

Command (CMD)

Specifies a character string of up to 2000 characters that represents a command to be run on the target system. The maximum length supported by the target system may be less than 2000 characters. If you specify a command string that exceeds the maximum length supported by the target system, the command will fail.

The command must be enclosed in single quotation marks if it contains embedded blanks or special characters.

This is a required parameter.

Note: The normal rule of pairing single quotation marks in quoted strings on the local system must be doubled when the same string is submitted to a remote system on this CMD parameter; this is required

because you are coding a quoted string within another quoted string. Therefore, when this parameter is being coded, wherever a single quotation mark would normally be paired with another single quotation mark, each occurrence in the inside set of single quotation marks must be doubled to produce the same results at the target system.

Top

Remote location (RMTLOCNAME)

Specifies the target system and the type of connection used to connect to the target system.

This is a required parameter.

Element 1: Name or address

character-value

The syntax for this element is dependent on the value specified for the second element:

- When the address type is *SNA, specify the name of the target system using the format nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name. If no network ID is specified, the network attributes are used to determine the default network ID.
- When the address type is *IP, specify a host name or an internet address to identify the target system. Host names must follow these conventions:
 - The first character must be either A through Z or 0 through 9.
 - The last character must be either A through Z, 0 through 9, or a period (.).
 - Blanks () are not allowed.
 - The special characters, period (.), underscore (_), and minus (-) are allowed.
 - Parts of the name separated by periods (.) cannot exceed 63 characters in length.
Note: Each part of the name separated by periods (.) must begin and end with an English alphabetic character or a numeric character.
 - Names must be from 1 to 255 characters in length.

The IPv4 address is specified in the form nnn.nnn.nnn.nnn where nnn is a decimal number ranging from 0 through 255. An IPv4 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.

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

If the Internet address is entered from a command line, the address must be enclosed in single quotation marks.

Element 2: Type

***SNA** The target system is accessed using a Systems Network Architecture (SNA) address and protocol.

***IP** The target system is found using a host name or an internet address over a TCP/IP connection.

Top

Remote user ID (RMTUSER)

Specifies the user identifier (ID) for the target system. If a user ID is specified for this parameter and password security is active on the target system, RMPWD(*NONE) is not valid.

*NONE

No user ID is sent. If security on the target system is configured to require a user ID, the command will fail.

*CURRENT

The user ID of the job (signed-on user) using this command is sent.

character-value

Specify a user ID to use that exists on the target system. If a user ID is specified and password security is active on the target system, a password must be specified.

Top

Remote password (RMPWD)

Specifies the password sent to the target system.

*NONE

The system does not send a password. If a user identifier (ID) is specified on the RMTUSER parameter and password security is active on the target system, the command will fail.

character-value

Specify a password sent to the target system to verify the sign-on of the user ID specified in the RMTUSER parameter. If an address type of *IP is specified, the password sent is not substituted across the communication line; if *SNA is specified, the password may or may not be substituted, depending on whether the remote system supports password substitution.

Top

Mode (MODE)

Specifies the name of the mode to be used when sending the command to the target system. This parameter is only allowed when the address type value is *SNA.

*NETATR

The mode in the network attributes is used.

communications-name

Specify the name of the mode object to use. Specify BLANK for a mode name consisting of eight blank characters.

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

Top

Coded character set ID (CCSID)

Specifies the coded character set identifier (CCSID) that will be used for conversions. When data is sent to the remote system, the data is converted using the CCSID of the job to the CCSID specified on the CCSID parameter. Similarly, when data is received from the remote system, the data is converted from the CCSID that was specified on the CCSID parameter to the CCSID of the job.

Restrictions:

1. If the CCSID of the job is 65535 (indicating no conversion), the default CCSID of the job will be used.
2. This command uses the round-trip conversion method when converting data from the source CCSID to the target CCSID and back. For more information about CCSID conversion methods, see the National Language Support book.

*CALC

The CCSID value sent from the target system is used if it is available. If a value is not provided from the target system, a default value of 00819 (ISO 8859-1 8-bit ASCII) will be used.

1-65533

Specify the CCSID value to use. This value is validated to ensure a valid CCSID has been requested.

Top

Wait time (in seconds) (WAITTIME)

Specifies the time in seconds to wait for the return (echo) before declaring the remote location to be unreachable.

*NOMAX

The system waits forever.

*NOWAIT

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

2-3600 Specify the number of seconds to wait.

Top

Examples

Example 1: Run Remote Command (RUNRMTCMD) Using SNA Address Type

```
RUNRMTCMD  CMD('dir')  RMTLOCNAME(APPN.RMTSYS *SNA)
```

This command runs the 'dir' command on the system specified by 'RMTLOCNAME' using an SNA address type.

Example 2: Run Remote Command (AREXEC) Using SNA Address Type

```
AREXEC  CMD('dir')  RMTLOCNAME(APPN.RMTSYS)
```

This command is equivalent to the command in example 1.

Example 3: Run Remote Command Using IP Address Type and Host Name to Specify Target System

```
RUNRMTCMD  CMD('ls')  RMTLOCNAME(MYSYS.NET1.LOCAL *IP)
```

This command runs the 'ls' command on the system specified by host name MYSYS.NET1.LOCAL.

Example 4: Run Remote Command Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD  CMD('ls')  RMTLOCNAME('9.5.1.94' *IP)
```

This command runs the 'ls' command on the system specified by internet address '9.5.1.94'.

Example 5: Run Remote Command With Multiple Commands Using IP Address Type and Internet Address to Specify Target System

```
RUNRMTCMD  CMD('ls; cat myfile; date')  
           RMTLOCNAME('9.5.1.94' *IP)
```

This command runs multiple commands, first 'ls', then 'cat myfile', then 'date' on system specified by internet address '9.5.1.94'.

Example 6: Run Remote Command With a Wait Time Expiration Value

```
RUNRMTCMD  CMD('dir')  RMTLOCNAME(APPN.RMTSYS *SNA)  
           WAITTIME(15)
```

This command runs the 'dir' command on the specified system. The maximum wait time for the remote location to respond is 15 seconds. A wait time is only allowed when using *SNA address type value.

Example 7: Run Remote Command Using IP Address Type and IPv6 Address to Specify Target System

```
RUNRMTCMD  CMD('ls')  RMTLOCNAME('2001:D88::1' *IP)
```

This command runs the 'ls' command on the system specified by IPv6 address '2001:D88::1'.

Top

Error messages

***ESCAPE Messages**

CPF91CB

Problems occurred on the command, but the command completed.

CPF91CC

Command did not complete successfully.

CPF91CF

Command failed on remote system.

CPF91C9

MODE value not allowed when *IP specified for RMTLOCNAME address type.

Top

Run RNDC Command (RUNRNDCCMD)

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

Parameters
Examples
Error messages

The Run RNDC Command (RUNRNDCCMD) command, or its alias RNDC, starts the Remote Name Daemon Control utility. This command allows you to control a Domain Name System (DNS) running on your local system.

Restrictions:

- You must have execute (*X) authority to the directories in the path `/QIBM/UserData/OS400/DNS/_DYN`
- You must have read (*R) authority to the `/QIBM/UserData/OS400/DNS/_DYN` directory files.
- You must have execute (*X) authority to the directories in the path of the RNDC configuration file.
- You must have read (*R) authority to the RNDC configuration file.
- You must have execute (*X) authority to the directories in the path of the RNDC key file.
- You must have read (*R) authority to the key file.
- You must have execute (*X) authority to the directories in the path of the output file.
- You must have write (*W) authority to the output file if it already exists.
- You must have read, write and execute (*RWX) authority to the output file's parent directory if the output file does not already exist.

Top

Parameters

Keyword	Description	Choices	Notes
RNDCCMD	RNDC command	<i>Character value</i>	Required, Positional 1
DMNNAMSVR	Domain name server	<i>Character value, *DFT</i>	Optional, Positional 2
PORT	Domain name server port	1-65535, <u>953</u>	Optional, Positional 3
RNDCCFGF	RNDC configuration file	<i>Path name, *DFT</i>	Optional
SRCADR	Source address	<i>Character value, *DFT, *ANY4, *LOOPBACK4, *ANY6, *LOOPBACK6</i>	Optional
KEYFILE	Key file	<i>Path name, *DFT</i>	Optional
KEYNAME	Key name	<i>Character value, *DFT</i>	Optional
DEBUG	Show debug information	<u>*NO</u> , *YES	Optional
TOSTMF	Output file	<i>Path name, *STDOUT</i>	Optional

Top

RNDC command (RNDCCMD)

Specifies the control command to be sent to the DNS server.

This is a required parameter.

character-value

Specify the control command to run. The command must be one of the following:

```
reload
  Reload configuration file and zones
reload zone [class [view]]
  Reload a single zone
refresh zone [class [view]]
  Schedule immediate maintenance for a zone
retransfer zone [class [view]]
  Re-transfer a single zone without checking
  serial number
freeze zone [class [view]]
  Suspend updates to a dynamic zone
thaw zone [class [view]]
  Enable updates to a frozen dynamic zone and
  reload it
reconfig
  Reload configuration file and new zones only
stats
  Write server statistics to the statistics file
querylog
  Toggle query logging
dumpdb [-all|-cache|-zones] [view ...]
  Dump cache(s) to the dump file (named_dump.db)
stop
  Save pending updates to master files and stop
  the server
stop -p
  Save pending updates to master files and stop
  the server reporting process id
halt
  Stop the server without saving pending updates
halt -p
  Stop the server without saving pending updates
  reporting process id
trace
  Increment debugging level by one
trace level
  Change the debugging level
notrace
  Set debugging level to 0
flush
  Flushes all of the servers caches
flush [view]
  Flushes the servers cache for a view
flushname name [view]
  Flush the given name from the servers cache(s)
status
  Display status of the server
recurring
  Dump the queries that are currently recurring
  (named.recurring)
```

Top

Domain name server (DMNNSVR)

Specifies the name or the IP address of the DNS server that RNDC will use as its current server for the query session. You can specify any DNS server to which your TCP/IP network has access.

Note: RNDC sends information to DNS servers and it needs an active DNS server to send its commands. If you do not specify a DNS server with DMNNSVR when you start the tool, it will attempt to use the default server, port and key defined in the `rndc.conf` file. This is usually the loopback interface 127.0.0.1 and port 953.

***DFT** Use the default DNS server defined in the `/QIBM/UserData/OS400/DNS/_DYN/rndc.conf` file.

server-domain-name

Specify the name of a DNS server. This is a domain name like 'myserver.i5os.ibm.com'.

server-internet-address

Specify the IP address of a DNS server.

RNDC-server-name

Specify the name of a DNS server in the `rndc.conf` file. This can be the name like 'NS'.

Top

Domain name server port (PORT)

Specifies the default server port to use.

953 Use control channel port 953.

1-65535

Specify a valid port number.

Top

RNDC configuration file (RNDCCFGF)

Specifies the RNDC configuration file to be used for this session. This file contains both access key and option statements that define the default server and the access key for that server. It is possible to use RNDC to control DNS servers located on other systems if this configuration file contains the access key and server statements that apply to the remote servers.

***DFT** Use `/QIBM/UserData/OS400/DNS/_DYN/rndc.conf` as the default configuration file. The default server in this file is 'localhost'.

path-name

Specify the path name for a stream file containing RNDC configuration information. For example, `/home/myprofile/my-rndc-conf-file`.

Top

Source address (SRCADR)

Specifies the source address for the connection to the server.

***DFT** Uses the default supplied by the stack.

***ANY4**

Use the IPv4 wildcard address ('0.0.0.0').

***LOOPBACK4**

Use the IPv4 loopback address ('127.0.0.1').

***ANY6**

Use the IPv6 wildcard address ('::').

***LOOPBACK6**

Use the IPv6 loopback address ('::1').

character-value

Specify a valid IPv4 or IPv6 internet address.

Top

Key file (KEYFILE)

Specifies the access key file to use for this session. The key in this file will be used to authenticate commands sent to the server.

***DFT** Use /QIBM/UserData/OS400/DNS/_DYN/rndc.key as the default key file.

path-name

Specify the path name for a stream file containing key information. For example, '/home/myprofile/my-rndc-key-file'.

Top

Key name (KEYNAME)

Specifies the access key name to be used for this session. This key name must be known by the server that is being used for this session. If no key name is specified, RNDC will use the defaults from the /QIBM/UserData/OS400/DNS/_DYN/rndc.conf file.

***DFT** Use key name *rndc-key*.

character-value

Specify the RNDC key name to use.

The DNS server being queried needs to include this key and algorithm in its named.conf configuration file in order to allow RNDC clients with this key.

Top

Show debug information (DEBUG)

Specifies whether or not to turn debugging mode on. More information is displayed about the packet sent to the server and the resulting answer when debugging mode is on.

***NO** Turn off debugging messages.

***YES** Turn on debugging messages.

Top

Output file (TOSTMF)

Specifies the name of a stream file where all command output is written.

***STDOUT**

All command output goes to the standard output device (normally the display).

path-name

Specify the path for a stream file where output should be written.

Top

Examples

Example 1: Reload All DNS Server Configuration and Static Zones

```
RUNRNDCCMD  RNDCCMD('reload')
```

This command illustrates a simple reload of any changes to a DNS server configuration and any static zones.

The output from this type of query might look like this:

```
server reload successful
```

Similarly, if your RNDCC key from the rndc.conf file is not valid, the output from this type of query might look like this:

```
rndc: connection to remote host closed
This may indicate that
* the remote server is using an older version of the
  command protocol,
* this host is not authorized to connect,
* the clocks are not synchronized, or
* the key is invalid.
```

Example 2: Reload a Single Zone

```
RUNRNDCCMD  RNDCCMD('reload i5os.ibm.com')
```

This command illustrates a simple reload of a single static zone called 'i5os.ibm.com'.

The output from this type of query might look like this:

```
zone reload up-to-date
```

If the zone is not a static zone, the output from this type of query might look like this:

```
rndc: 'reload i5os.ibm.com' failed: dynamic zone
```

Example 3: Dumping the Cache

```
RUNRNDCCMD  RNDCCMD('dumpdb -cache')
```

This command illustrates how to dump the active cache on the server. The output from this type of query will be empty, because it goes to file named_dump.db in the server directory, i.e., for server NS the file would be /QIBM/UserData/OS400/DNS/NS/named_dump.db.

Top

Error messages

*ESCAPE Messages

DNS0013

Error processing command parameters.

DNS0065

Option 33 of i5/OS is required, but is not installed.

TCP7124

Program &1 in library &2 type *PGM ended abnormally.

Top

Run SQL Statements (RUNSQLSTM)

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

Parameters
 Examples
 Error messages

The Run SQL Statements (RUNSQLSTM) command processes a source file of Structure Query Language (SQL) statements.

Top

Parameters

Keyword	Description	Choices	Notes
SRCFILE	Source file	<i>Qualified object name</i>	Optional, Positional 1
	Qualifier 1: Source file	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
SRCMBR	Source member	<i>Name</i>	Optional, Positional 2
SRCSTMF	Source stream file	<i>Path name</i>	Optional
COMMIT	Commitment control	<i>*CHG, *UR, *CS, *ALL, *RS, *NONE, *NC, *RR</i>	Optional, Positional 3
NAMING	Naming	<i>*SYS, *SQL</i>	Optional, Positional 4
ERRLVL	Severity level	0-40, <u>10</u>	Optional
DATFMT	Date format	<i>*JOB, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL</i>	Optional
DATSEP	Date separator character	<i>*JOB, '/', '.', ',', '-', ' ', *BLANK</i>	Optional
TIMFMT	Time format	<i>*HMS, *USA, *ISO, *EUR, *JIS</i>	Optional
TIMSEP	Time separator character	<i>*JOB, ':', '.', ',', '-', ' ', *BLANK</i>	Optional
MARGINS	Source margins	<i>Element list</i>	Optional
	Element 1: Right margin	1-32754, <u>80</u>	
DFTRDBCOL	Default collection	<i>Name, *NONE</i>	Optional
SAALFLAG	IBM SQL flagging	<i>*NOFLAG, *FLAG</i>	Optional
FLAGSTD	ANS flagging	<i>*NONE, *ANS</i>	Optional
DECMPT	Decimal Point	<i>*JOB, *SYSVAL, *PERIOD, *COMMA</i>	Optional
SRTSEQ	Sort sequence	Single values: <i>*JOB, *LANGIDUNQ, *LANGIDSHR, *HEX</i> Other values: <i>Qualified object name</i>	Optional
	Qualifier 1: Sort sequence	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
LANGID	Language id	<i>Character value, *JOB</i>	Optional
PRTFILE	Print file	<i>Qualified object name</i>	Optional
	Qualifier 1: Print file	<i>Name, QSYSPRT</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
PROCESS	Statement processing	<i>*RUN, *SYN</i>	Optional
SECLVLTXT	Second level text	<i>*NO, *YES</i>	Optional
ALWCPYDTA	Allow copy of data	<i>*OPTIMIZE, *YES, *NO</i>	Optional

Keyword	Description	Choices	Notes
ALWBLK	Allow blocking	<u>*ALLREAD</u> , *NONE, *READ	Optional
SQLCURRULE	SQL rules	<u>*DB2</u> , *STD	Optional
DECRESULT	Decimal result options	<i>Element list</i>	Optional
	Element 1: Maximum precision	<u>31</u> , 63	
	Element 2: Maximum scale	0-63, <u>31</u>	
	Element 3: Minimum divide scale	0-9, <u>0</u>	
OUTPUT	Listing output	<u>*NONE</u> , *PRINT	Optional
TGTRLS	Target release	<i>Simple name</i> , <u>*CURRENT</u>	Optional
DBGVIEW	Debugging view	<u>*NONE</u> , *SOURCE, *STMT, *LIST	Optional
CLOSQCSR	Close SQL cursor	<u>*ENDMOD</u> , *ENDACTGRP	Optional
DLYPRP	Delay PREPARE	<u>*NO</u> , *YES	Optional
USRPRF	User profile	<u>*NAMING</u> , *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	<u>*USER</u> , *OWNER	Optional

Top

Source file (SRCFILE)

Specifies the source file that contains the Structured Query Language (SQL) statements to be run.

Qualifier 1: Run SQL Statements

name Specify the name of the source file that contains the SQL statements to be run. The source file can be a database file or an inline data file.

Qualifier 2: Library

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

*CURLIB

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

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the Structured Query Language (SQL) statements to be run.

name Specify the name of the source file member that contains the SQL statements to be run.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the SQL statements. The path name can be either absolute or relative.

Top

Commitment control (COMMIT)

Specifies whether SQL statements are run under commitment control.

***CHG or *UR**

Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL or *RS**

Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE or *NC**

Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Naming convention (NAMING)

Specifies the naming convention used for objects in SQL statements.

***SYS** The system naming convention (library-name/file-name) is used.

***SQL** The SQL naming convention (schema-name.table-name) is used.

Top

Severity level (ERRLVL)

Specifies whether the processing is successful, based on the severity of the messages generated by the processing of the SQL statements. If errors that are greater than the value specified for this parameter occur during processing, no more statements are run and the statements are rolled back if they are running under commitment control. Syntax checking will be attempted for the remaining statements in the source.

10 Statement processing is stopped when error messages with a severity level greater than 10 are received.

0-40 Specify the severity level to be used.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. For input date strings, the specified value is used to determine whether the date is specified in a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

***JOB** The format specified for the job is used. Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

/' A slash is used as the date separator.

.' A period is used as the date separator.

-' A dash is used as the date separator.

',' A comma is used as the date separator.

'' or *BLANK

A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. For input time strings, the specified value is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when ***HMS** is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

'/' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' or ***BLANK**

A blank is used as the time separator.

Top

Source margins (MARGINS)

Specifies the part of the source input record that contains source text. The left margin is always position 1 of the input record. The right margin defaults to 80.

If the **Source stream file (SRCSTMF)** parameter is specified, margins are ignored.

1-32754

Specify the ending position to be used for each input record.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**

The naming convention specified for the **Naming convention (NAMING)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the NAMING parameter.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

*NOFLAG

No checks are made to see whether SQL statements conform to IBM SQL syntax.

*FLAG

Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

*NONE

No checks are made to see whether SQL statements conform to ANSI standards.

*ANS Checks are made to see whether SQL statements conform to standards.

Top

Decimal Point (DECMPT)

Specifies the decimal point value used for numeric constants in SQL statements. This value is also used as the decimal point character when casting between character and numeric values.

*JOB The representation for the decimal point is the value used by the job running the statement.

*SYSVAL

The QDECFMT system value is used as the decimal point.

*PERIOD

A period represents the decimal point.

*COMMA

A comma represents the decimal point.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Single values

*JOB The SRTSEQ value for the job is used.

*LANGIDUNQ

The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

*LANGIDSHR

The shared-weight sort table for the language specified for the LANGID parameter is used.

*HEX A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved.

language-identifier

Specify a language identifier.

Top

Print file (PRTFILE)

Specifies the printer device file to which the RUNSQLSTM printout is directed. The file must have a minimum length of 132 bytes. If a file with a record length of less than 132 bytes is specified, information is lost.

Qualifier 1: Print file

QSYSPRT

The RUNSQLSTM output file is directed to the IBM-supplied printer file, QSYSPRT.

name Specify the name of the printer device file to which the RUNSQLSTM output is directed.

Qualifier 2: Library

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

***CURLIB**

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

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

Top

Statement processing (PROCESS)

Specifies whether SQL statements in the source file member are executed or syntax-checked only.

***RUN** Statements are syntax-checked and run.

***SYN** Statements are syntax-checked only.

Second level text (SECLVLTXT)

Specifies whether the second-level message text descriptions should be written to the output listing.

- *NO** Second-level text is not included in the listing.
- *YES** Second-level text with replacement data is added to the listing for all messages.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**

The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

- *YES** A copy of the data is used only when necessary.
- *NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**

Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
 - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
 - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.

- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

DB2 The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NONE

The precompiler listing is not generated.

*PRINT

The precompiler listing is generated.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release.

*CURRENT

The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release

Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by SQL.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NONE

No debug view information is generated.

*SOURCE

Generates a source view of the C source generated by RUNSQLSTM for the SQL procedures, functions, or triggers in the input SQL source member. The C source member is passed to the SQL precompiler by invoking the CRTSQLCI (Create SQL ILE C object) command. A source view is also generated by the SQL precompiler for the C source member which is produced by the precompiler.

***STMT**

Allows the compiled object to be debugged using program statement numbers and symbolic identifiers.

***LIST** Generates the listing view for debugging the compiled object.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements. This option is ignored for SQL table functions.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

***ENDMOD**

SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

***ENDACTGRP**

SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES on this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*NAMING

The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

*USER

The profile of the user running the program or SQL package is used.

*OWNER

The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

Note: This parameter applies only to CREATE and ALTER statements for SQL procedures, functions, or triggers in the source file. This value will be used when creating the program for the SQL routine.

*USER

Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

*OWNER

Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Examples

```
RUNSQLSTM SRCFILE(MYLIB/MYFILE) SRCMBR(MYMBR)
```

This command processes the SQL statements in member MYMBR found in file MYFILE in library MYLIB.

Top

Error messages

*ESCAPE Messages

SQL9006

DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

SQL9010

RUNSQLSTM command failed.

SQL9014
Remote Connection Active

[Top](#)

Revoke Access Code Authority (RVKACCAUT)

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

Parameters
Examples
Error messages

The Revoke Access Code Authority (RVKACCAUT) command allows you to revoke the access code authority for an individual user or a group of users.

Restrictions:

You must have all object (*ALLOBJ) special authority to revoke access code authority for other users.

Top

Parameters

Keyword	Description	Choices	Notes
ACC	Document access code	Single values: *ALL Other values (up to 300 repetitions): 1-2047	Required, Positional 1
USER	User profile	Single values: *CURRENT, *ALL Other values (up to 300 repetitions): <i>Name</i>	Required, Positional 2

Top

Document access code (ACC)

Specifies the access code for which authority is revoked.

This is a required parameter.

***ALL** All access code authority for the user is revoked.

1-2047 Specify the access code for which authority is revoked. A maximum of 300 access codes can be specified here.

Top

User profile (USER)

Specifies the names of the user profile for which access code authority is revoked.

This is a required parameter.

***CURRENT**

Specifies that the access code authority of the user currently running this command is revoked. This is the only valid option you can specify unless you have all object (*ALLOBJ) or security administrator (*SECADM) special authority.

Note: You can specify *CURRENT on the USER parameter to remove authority from any access codes to which you have authority.

***ALL** Access code authority is revoked from all users.

name Specify the name of the user profile for which access code authority is revoked.

Top

Examples

Example 1: Revoking Authority of Current User

```
RVKACCAUT ACC(250) USER(*CURRENT)
```

This command takes away the access code authority of access code 250 from the user currently running this command.

Example 2: Revoking Authority of Specific User

```
RVKACCAUT ACC(300) USER(BILLY)
```

This command takes away the access code authority of access code 300 from user BILLY. This command must be run by someone with *ALLOBJ or *SECADM special authority, or by user BILLY. A user who runs this command for himself can enter USER(*CURRENT) or his own user profile name; they are the same.

Top

Error messages

*ESCAPE Messages

CPF9009

System requires file &1 in &2 be journaled.

CPF9014

Access code authority removed from &1 users, not revoked from &2 users.

CPF9024

System cannot get correct record to finish operation.

CPF9037

Not allowed to specify USER(*ALL).

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

Revoke Object Authority (RVKOBJAUT)

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

Parameters
Examples
Error messages

The Revoke Object Authority (RVKOBJAUT) command is used to take away specific (or all) authority for the named object(s) from one or more users named in the command, or to remove the authority of an authorization list for the named object(s). This command can be run by the security officer, by an object's owner, or by a user who has object management authority for the object to be revoked. Users who have object management authority can revoke only the explicit authority that they have. A user may not be able to grant or revoke authority for an object that has been allocated (locked) by another job. Authority cannot be revoked for an object that is currently in use.

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 System i Security Reference, SC41-5302.

Restrictions:

- Before this command is used to remove authorities to use a device, control unit, or line description, its associated device, control unit, or line must be varied on.
- Authority to use a device cannot be revoked if a user is currently signed on to the device.
Note: Users can revoke their own authority to a device if they are currently signed onto that device. However, doing so may produce unpredictable results and is not advisable.
- For display stations or for work station message queues associated with the display station, if this command is not run from the device for which authorities are to be revoked, it should be preceded by the Allocate Object (ALCOBJ) command and followed by the Deallocate Object (DLCOBJ) command.
- Object type *DOC or *FLR cannot be specified.
- Document interchange support must be used.
- Object type *AUTL cannot be specified. The Change Authorization List Entry (CHGAUTLE) or Remove Authorization List Entry (RMVAUTLE) commands must be used. AUT (*AUTL) can be specified only with USER (*PUBLIC).
- Only a user with *ALL authority or the owner can remove the authorization list.
- You must have *USE authority to the auxiliary storage pool device if one is specified.

*** Security Risk ***

Revoking all authorities specifically given to a user for an object can result in the user having more authority than before the revoke operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	<i>Qualified object name</i>	Required, Positional 1
	Qualifier 1: Object	<i>Generic name, name, *ALL</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB, *ALL, *ALLUSR, *USRLIBL, *ALLAVL, *ALLUSRAVL</i>	
OBJTYPE	Object type	<i>*ALL, *ALRTBL, *BNDDIR, *CFGL, *CHTFMT, *CLD, *CLS, *CMD, *CNL, *COSD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *CTLD, *DEVD, *DTAARA, *DTADCT, *DTAQ, *EDTD, *FCT, *FILE, *FNTRSC, *FNNTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *IPXD, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LIB, *LIND, *LOCALE, *M36, *M36CFG, *MEDDFN, *MENU, *MGTCOL, *MODD, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *NTBD, *NWID, *NWSCFG, *NWS, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRDDFN, *PRDLOD, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *S36, *SBS, *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: <i>*ALL, *PUBLIC</i> 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</i>	Optional

Top

Object (OBJ)

Specifies the objects to have specific authority revoked. If **ALL* is specified for the object name, a library name must be specified.

This is a required parameter.

Qualifier 1: Object

***ALL** All objects of the specified type (OBJTYPE) found in the search have specific authorities revoked. You must specify the name of a library when **ALL* is specified for the object name.

generic-name

Specify the generic name of the objects that are to have specific authorities revoked.

A generic name is a character string of one or more characters followed by an asterisk (*); for example ABC*. The asterisk substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

name Specify the name of the object that is to have specific authorities revoked.

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. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***USRLIBL**

If a current library entry exists in the library list for the current thread, the current library and the libraries in the user portion of the library list are searched. If there is no current library entry, only the libraries in the user portion of the library list are searched. If the **ASP device (ASPDEV)** parameter is specified when this value is used, ASPDEV(*) is the only valid value.

***ALL** All the libraries in the auxiliary storage pools (ASPs) specified for the **ASP device (ASPDEV)** parameter are searched.

***ALLUSR**

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

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

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

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

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFERDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

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

***ALLAVL**

All libraries in all available ASPs are searched.

***ALLUSRAVL**

All user libraries in all available ASPs are searched. Refer to *ALLUSR for a definition of user libraries.

name Specify the name of the library to be searched.

Top

Object type (OBJTYPE)

Specifies the object type of the object that has specific authorities revoked. For a complete list of supported object types, position the cursor on this parameter while prompting the command and press F4.

This is a required parameter.

***ALL** All supported object types have specific authorities revoked.

object-type

Specify the object type that is to have specific authorities revoked.

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 whose specific authorities to the named object are to be revoked.

Note: Either this parameter or the **Authorization list (AUTL)** parameter must be specified.

Authorities revoked by this command are related to those given by the Grant Object Authority (GRTOBJAUT) command. If users have public authority to an object because USER(*PUBLIC) was specified on the GRTOBJAUT command, that public authority is revoked when *PUBLIC is specified on this parameter. If users have specific authorities to an object because their names were specified on the GRTOBJAUT command, their names must be specified on this parameter to revoke the same authorities.

The authorities to be revoked are specified on the **Authority (AUT)** parameter.

Single values

***ALL** The authorities specified are to be taken away from all enrolled users of the system except the owner, whether they were publicly or explicitly authorized.

***PUBLIC**

The specified authorities are taken away from users who do not have specific authority for the

object, are not on the authorization list, and whose group has no authority. Any users who have specific authorities still keep their authorities to the object.

Other values (up to 50 repetitions)

name Specify the name of the user profile of the user that is to have the specified authorities revoked. This parameter cannot be used to revoke public authority from specific users; only authorities that were specifically given to a user can be specifically revoked. A maximum of 50 user profile names can be specified.

Top

Authority (AUT)

Specifies the authorities to be revoked from the users who do not have specific authority to the object, who are not on an authorization list, and whose user group does not have specific authority to the object.

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 that is revoked from the object specified for the **Object (OBJ)** parameter. If public authority in the object is *AUTL, it is changed to *EXCLUDE.

Note: Either this parameter or the **Users (USER)** parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

name Specify the name of the authorization list.

Top

Examples

Example 1: Removing Authority From All Users Except Program Owner

```
RVKOBJAUT OBJ(ARLIB/PROG1) OBJTYPE(*PGM) USER(*ALL)
```

This command removes the authorities (AUT was not specified; *CHANGE is assumed) from all users who were either explicitly or publicly authorized, except the owner, for the program (*PGM) named PROG1 located in the library named ARLIB.

Example 2: Removing Object Owner's Authority to Delete a Program

```
RVKOBJAUT OBJ(TSMITHPGM/MITHLIB) OBJTYPE(*PGM)  
USER(TSMITH) AUT(*OBJEXIST)
```

This command removes the object owner's (TSMITH) authority to delete a program (TSMITHPGM) in his library (SMITHLIB). The object owner might do this to ensure that the object is not deleted by mistake. If the owner ever wants to delete the object, object existence authority for the object can be granted by using the Grant Object Authority (GRTOBJAUT) command).

Example 3: Removing *DLT and *UPD Authorities

```
RVKOBJAUT  OBJ(FILEX) OBJTYPE(*FILE)
           USER(HEANDERSON) AUT(*DLT *UPD)
```

This command removes delete and update authorities for the file named FILEX from the user HEANDERSON.

Example 4: Removing *OBJEXIST Authority

```
RVKOBJAUT  OBJ(ARLIB/ARJOB) OBJTYPE(*JOB) USER(RLJOHNSON)
           AUT(*OBJEXIST)
```

This command removes the object existence authority for the object named ARJOB from the user RLJOHNSON. ARJOB is a job description that is located in the library named ARLIB.

Example 5: Removing Specific Authorities

```
RVKOBJAUT  OBJ(FILEX) OBJTYPE(*FILE) AUTL(FILEUSERS)
```

This command removes specific authorities for the file named FILEX from the users in the authorization list FILEUSERS.

Top

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.

CPF22A4

*EXCLUDE cannot be revoked from *PUBLIC.

CPF22A5

Object &1 in &3 type *&2 not secured by authorization list &4.

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.

CPF2224
Not authorized to revoke authority for 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).

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.

CPF9804
Object &2 in library &3 damaged.

***STATUS Messages**

CPF2256
Authority not revoked from all objects or all users.

Top

Revoke Public Authority (RVKPUBAUT)

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

Parameters
Examples
Error messages

The Revoke Public Authority (RVKPUBAUT) limits the use of a set of IBM-supplied commands and programs by changing the public authority to *EXCLUDE. To determine what commands and programs are being restricted, issue the Retrieve CL Source (RTVCLSRC) command against the program QSECRVKP and examine the source file created by the RTVCLSRC command.

Restriction: You must have all object (*ALLOBJ) special authority to run this command.

This command can be customized by the security administrator by following the steps below:

1. Issue the Retrieve CL Source (RTVCLSRC) command against the program QSECRVKP.
2. Edit the source code produced from the RTVCLSRC command and compile the new program. Make sure that the program is given a new name, is created into a library other than QSYS, and that the *PUBLIC authority is set to *EXCLUDE.
3. Issue the Change Command (CHGCMD) against the Revoke Public Authority command and specify your new program for the PGM parameter. An example is listed below:

```
CHGCMD CMD(QSYS/RVKPUBAUT) PGM(library_name/new_pgm_name)
```

Note: If a product upgrade is done, the RVKPUBAUT command is reinstalled, or maintenance is applied to the RVKPUBAUT command, the CHGCMD will have to be issued again to customize the command.

DISCLAIMER: IBM cannot guarantee or imply reliability, serviceability, performance or function of the retrieved QSECRVKP source code and any programs. THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Name, <u>QSYS</u>	Optional

Top

Library (LIB)

The name of the library where the IBM-supplied commands are to be found. The library value is only used for commands because they can be located in secondary language libraries.

QSYS The commands are located in library QSYS.

library-name

The library where the commands are located.

Top

Examples

RVKPUBAUT LIB(QSYS)

This command sets the public authority for commands and programs in library QSYS to *EXCLUDE.

[Top](#)

Error messages

*ESCAPE Messages

CPF304

User does not have required special authorities.

[Top](#)

Revoke User Permission (RVKUSRPMN)

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

Parameters
Examples
Error messages

The Revoke User Permission (RVKUSRPMN) command allows you to revoke user permission from one user (or all users) to access documents or folders on behalf of another user.

Note: If work is being done on behalf of another user at the time this command is running, functions that have started are completed; however, additional functions are not accepted.

Restrictions:

You must have all object (*ALLOBJ) special authority to revoke document authority for other users.

Top

Parameters

Keyword	Description	Choices	Notes
FROMUSER	From user profile	<i>Name</i> , *ALL	Required, Positional 1
FORUSER	For user profile	Single values: *CURRENT Other values (up to 300 repetitions): <i>Name</i>	Optional, Positional 2

Top

From user profile (FROMUSER)

Specifies the user profile name of the user whose permission is revoked.

This is a required parameter.

name Specify the name of the user profile that is no longer permitted to work on behalf of the user specified on the **For user profile (FORUSER)** parameter.

***ALL** All users who are currently permitted to work on behalf of other users are no longer permitted to do so.

Top

For user profile (FORUSER)

Specifies the user profile name of the user on whose behalf the user specified on the **From user profile (FROMUSER)** parameter can no longer work.

***CURRENT**

The user specified on the **From user profile (FROMUSER)** parameter can no longer work on your behalf.

name Specify the name of the user profile on whose behalf other users are no longer permitted to work.

Examples

```
RVKUSRPMN FROMUSER(JOHNSON) FORUSER(ANDERSON)
```

This command takes away user permission from JOHNSON for ANDERSON. The user JOHNSON is no longer allowed to work on behalf of ANDERSON.

Error messages

*ESCAPE Messages

CPF9008

User permission revoked for &1 users, not revoked for &2.

CPF9009

System requires file &1 in &2 be journaled.

CPF9024

System cannot get correct record to finish operation.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

Revoke Workstation Object Aut (RVKWSOAUT)

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

Parameters
 Examples
 Error messages

The Revoke Workstation Object Authority (RVKWSOAUT) command is used to take away authority for a workstation object used by the i5/OS Graphical Operations program. Specific or all authority can be taken away from one or more users named in the command. You also can take away the authority of an authorization list for the named object.

This command can be issued by a security officer, by an object owner, or by a user who has object management authority to the object for which authority is to be revoked. If a specific authority (other than *ALL) is specified on the AUT parameter, and that authority is not revoked, a message is issued that indicates the authority that is not revoked.

*** Security Risk ***

Revoking all authorities given specifically to a user for an object can result in the user having more authority than before the operation. If a user has *USE authority for an object and *CHANGE authority on the authorization list that secures the object, revoking *USE authority results in the user having *CHANGE authority to the object.

Restrictions:

- If you have object management (*OBJMGT) authority, you can revoke only the explicit authority that you have.
- You might not be able to grant or revoke authority for an object that has been allocated (locked) to another job. Authority cannot be revoked for an object that is currently in use.

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, *TPLJOBL, *JOBL, *TPLJOBQ, *TPLJOBLOG, *JOBLOG, *TPLJOBQL, *JOBQL, *TPLMSG, *MSG, *TPLMSGQ, *TPLMSGSD, *MSGSD, *TPLSGNUSL, *SGNUSL, *TPOBJL, *OBJL, *TPLLIBSL, *LIBSL, *TPLLIB, *LAUNCH, *TPLLAUNCH, *PRSET	
USER	Users	Single values: *ALL, *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

Top

Workstation object type (WSOTYPE)

Specifies the workstation object for which specific authorities are to be revoked.

This is a required parameter.

***TPLWRKARA**

The authorities to the work area template are revoked.

***WRKARA**

The authorities to the work area objects are revoked.

***TPLPRTOL**

The authorities to the printer output list template are revoked.

***PRTOL**

The authorities to the printer output list objects are revoked.

***TPLPRTL**

The authorities to the printer list template are revoked.

***PRTL** The authorities to the printer list objects are revoked.

***TPLOUTQ**

The authorities to the output queue template are revoked.

***TPLOUTQL**

The authorities to the output queue list template are revoked.

***OUTQL**

The authorities to the output queue list objects are revoked.

***TPLJOBL**

The authorities to the job list template are revoked.

***JOBL** The authorities to the job list objects are revoked.

***TPLJOBQ**

The authorities to the job queue template are revoked.

***TPLJOBLOG**

The authorities to the job log template are revoked.

***JOBLOG**

The authorities to the job log objects are revoked.

***TPLJOBQL**

The authorities to the job queue list template are revoked.

***JOBQL**

The authorities to the job queue list objects are revoked.

***TPLMSG**

The authorities to the message list template are revoked.

***MSG**

The user authorities to the message list objects are revoked.

***TPLMSGQ**

The authorities to the message queue template are revoked.

***TPLMSGSEND**

The authorities to the message sender template are revoked.

- *MSGSEND**
The authorities to the message sender objects are revoked.
- *TPLSGNUSL**
The authorities to the signed-on user list template are revoked.
- *SGNUSL**
The authorities to the signed-on user list objects are revoked.
- *TPLOBJL**
The authorities to the object list template are revoked.
- *OBJL** The authorities to the object list objects are revoked.
- *TPLLBSL**
The authorities to the library list template are revoked.
- *LIBSL**
The user authorities to the library list objects are revoked.
- *TPLLIB**
The authorities to the library template are revoked.
- *TPLLAUNCH**
The authorities to the job submitter template are revoked.
- *LAUNCH**
The authorities to the job submitter objects are revoked.
- *PRSET**
The authorities to the personal setting objects are revoked.

Top

Users (USER)

Specifies one or more users whose specific authorities to the named object are to be revoked.

Authorities revoked by this command are related to those given by the Grant Workstation Object Authority (GRTWSOAUT) command. If users have public authority to an object because USER(*PUBLIC) was specified on the GRTWSOAUT command, that public authority is revoked when *PUBLIC is specified on this parameter. If users have specific authorities to an object because their names were specified on the GRTWSOAUT command, their names must be specified on this parameter to revoke the same authorities.

The authorities to be revoked are those specified for the **Authority (AUT)** parameter.

Note: Either this parameter or the AUTL parameter must be specified.

***ALL** The authorities specified on the AUT parameter are taken away from all enrolled users of the system except the owner, if they are publicly or explicitly authorized.

***PUBLIC**
The specified authorities are taken away from users who do not have specific authority for the object, are not on the authorization list, and whose group has no authority. Any users who have specific authorities still keep their authorities to the object.

name Specify the name of the user profile that is to have the specified authorities revoked. This parameter cannot be used to revoke public authority from specific users; only authorities that were specifically given to a user can be specifically revoked.

Top

Authority (AUT)

Specifies the authority to be revoked from the users who do not have specific authority to the object, who are not on an authorization list, and whose user group does not have specific authority to the object.

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 that is revoked from the object specified on the WSOTYPE parameter. If public authority for the object is *AUTL, it is changed to *EXCLUDE. The authorization list's authority is then removed.

Note: Either this parameter or the USER parameter must be specified. If this parameter is specified, the AUT parameter is ignored.

name Specify the name of the authorization list.

Top

Examples

```
RVKWSOAUT  WSOTYPE(*SGNUSL)  USER(HEANDERSON)  AUT(*DLT *UPD)
```

This command removes the delete and the update authorities for signed-on user list objects from the user profile name HEANDERSON.

Top

Error messages

Unknown

Top

Save Object (SAV)

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

Parameters
 Examples
 Error messages

The Save (SAV) command saves a copy of one or more objects that can be used in the integrated file system.

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

Restrictions:

- For detailed restrictions on using this command to save objects by using name patterns in the root directory, to save objects in libraries, or to save document library objects, see the Recovering your system book, SC41-5304.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Values (up to 4 repetitions): <i>Path name</i>	Required, Positional 1
OBJ	Objects	Values (up to 300 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: Name	<i>Path name, *</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
PATTERN	Name pattern	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value, *</i>	
	Element 2: Include or omit	<u>*INCLUDE</u> , *OMIT	
SUBTREE	Directory subtree	<u>*ALL</u> , *DIR, *NONE, *OBJ, *STG	Optional
SAVACT	Save active	<u>*NO</u> , *YES, *SYNC	Optional
SAVACTOPT	Save active option	<u>*NONE</u> , *ALL, *ALWCKPWRT, *NWSSTG	Optional
OUTPUT	Output	<i>Path name, *NONE</i> , *PRINT	Optional
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional
LABEL	Label	<i>Character value, *GEN</i>	Optional
OPTFILE	Optical file	<i>Path name, *</i>	Optional
SEQNBR	Sequence number	1-16777215, <u>*END</u>	Optional
EXPDATE	File expiration date	<i>Date, *PERM</i>	Optional
ENDOPT	End of media option	<u>*REWIND</u> , *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	<u>*YES</u> , *NO	Optional
SAVACTMSGQ	Save active message queue	<i>Path name, *NONE</i> , *WRKSTN	Optional
SYNCID	Synchronization ID	<i>Name, *NONE</i>	Optional
INFTYPE	Type of output information	<u>*ALL</u> , *ERR, *SUMMARY	Optional
SYSTEM	System	*ALL, <u>*LCL</u> , *RMT	Optional

Keyword	Description	Choices	Notes
CHGPERIOD	Time period for last change	<i>Element list</i>	Optional
	Element 1: Start date	<i>Date, *ALL, *LASTSAVE</i>	
	Element 2: Start time	<i>Time, *ALL</i>	
	Element 3: End date	<i>Date, *ALL</i>	
	Element 4: End time	<i>Time, *ALL</i>	
PRECHK	Object pre-check	<i>*NO, *YES</i>	Optional
TGTRLS	Target release	<i>*CURRENT, *PRV, V5R3M0, V5R4M0, V6R1M0</i>	Optional
UPDHST	Update history	Single values: <i>*NO, *YES</i> Other values (up to 2 repetitions): <i>*SYS, *PC</i>	Optional
CLEAR	Clear	<i>*NONE, *ALL, *AFTER, *REPLACE</i>	Optional
DTACPR	Data compression	<i>*YES, *NO, *DEV, *LOW, *MEDIUM, *HIGH</i>	Optional
COMPACT	Data compaction	<i>*DEV, *NO</i>	Optional
ASPDEV	ASP device	<i>Name, *DFT, *ALLAVL, *, *SYSBAS, *CURASPGRP</i>	Optional
SCAN	Scan objects	<i>Element list</i>	Optional
	Element 1: Scan during save	<i>*NO, *YES</i>	
	Element 2: Save failed objects	<i>*NOSAVFAILED, *SAVFAILED</i>	
PVTAUT	Private authorities	<i>*NO, *YES</i>	Optional

Top

Device (DEV)

Specifies the device to which the objects are to be saved.

For more information on specifying device path names, refer to "Specifying the device name" in the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

This is a required parameter.

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

'save-file-path-name'

Specify the path name of the save file used to save the objects.

'optical-device-path-name'

Specify the path name of the optical device used to save the objects.

'tape-media-library-device-path-name'

Specify the path name of the tape media library device used to save the objects.

'tape-device-path-name'

Specify the path name of the tape device used to save the objects. A maximum of four tape devices can be specified. If a virtual tape device is used, it must be the only device specified.

'media-definition-path-name'

Specify the path name of the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data.

For information about creating a media definition, see the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

For information about using a media definition, see the Recovering your system book, SC41-5304 and the Back up your server topic in the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Objects (OBJ)

Specifies the objects to be saved. You can specify an object name pattern for the path name to be used. When a path name is specified that could match many objects, you can specify a value for the **Name pattern (PATTERN)** parameter to subset the objects that are to be saved.

A maximum of 300 path names can be specified.

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

Additional information about object name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 1: Name

'**' The objects in the current directory are saved.

path-name

Specify an object path name or a pattern that can match many names.

Element 2: Include or omit

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 determines whether the subtrees are included or omitted.

*INCLUDE

The objects that match the object name pattern are to be saved, unless overridden by an *OMIT specification.

*OMIT

The objects that match the object name pattern are not saved. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Name pattern (PATTERN)

Specifies one or more object name patterns to be used to subset the objects to be saved. The **Objects (OBJ)** parameter defines the set of candidate objects. A maximum of 300 values can be specified for this parameter.

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

Element 1: Pattern

* All objects which qualify for the operation are included or omitted.

character-value

Specify an object name or a pattern that can match many names.

Element 2: Include or omit

Specifies whether names that match the pattern should be included or omitted from the operation.

Note: The SUBTREE parameter determines whether the subtrees are included or omitted.

*INCLUDE

Only objects which are included by the OBJ parameter and match the PATTERN parameter are included in the save, unless overridden by an *OMIT specification.

*OMIT

All objects which are included by the OBJ parameter are included in the save except those objects which match the PATTERN parameter. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the save operation.

*ALL The entire subtree of each directory that matches the object name pattern is included. The subtree includes all subdirectories and the objects within those subdirectories.

*DIR The objects in the first level of each directory that matches the object name pattern are included. The subdirectories of each matching directory are included, but the objects in the subdirectories are not included.

*NONE

No subtrees are included in the save operation. If a directory matches the object name pattern specified, the objects in the directory are included. If the directory has subdirectories, neither the subdirectories nor the objects in the subdirectories are included.

*OBJ Only the objects that exactly match the object name pattern will be processed. If the object name pattern specifies a directory, objects in the directory are not included.

*STG The objects that match the object name pattern are processed along with the storage for related objects. Objects that are saved using this value can only be restored using SUBTREE(*STG).

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

- *NO** Objects that are in use are not saved. Objects cannot be updated while being saved.
- *YES** Objects can be saved and used at the same time. The object checkpoints can occur at different times.
- *SYNC** Objects can be saved and used at the same time. All of the object checkpoints occur at the same time.

Top

Save active option (SAVACTOPT)

Specifies options to be used with the save while active parameter.

- *NONE** No special save while active options will be used.
- *ALL** All of the options listed below are used during the save operation. See the description of the individual values to determine how the options will be handled.
- *ALWCKPWRT** Enables objects to be saved while they are being updated if the corresponding system attribute for the object is set.

Note: This option should only be used by applications to save objects that are associated with the application and that have additional backup and recovery considerations. For more information, refer to the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.
- *NWSSTG** Allows network server storage spaces in directory '/QFPNWSSTG' to be saved when they are active.

Top

Output (OUTPUT)

Specifies whether a list of information about the saved objects is created. The information can be directed to a spooled file, a stream file, or a user space.

A stream file or user space is specified as a path name.

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

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

***NONE**

No output is created.

***PRINT**

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

'stream-file-path-name'

Specify the path name of the existing stream file to which the output of the command is directed.

'user-space-path-name'

Specify the path name of the existing user space to which the output of the command is directed.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

A maximum of 75 volume identifiers can be specified. After all specified volumes are filled, the save operation continues on whatever volumes are placed in the device.

Single values

***MOUNTED**

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Top

Label (LABEL)

Specifies the file identifier of the media to be used for the save operation.

***GEN** The file label is created by the system.

- For objects in libraries, this is the equivalent of LABEL(*LIB) on the Save Object (SAVOBJ) and Save Library (SAVLIB) commands.
- For document library objects, this is the equivalent of LABEL(*GEN) on the Save Document Library Object (SAVDLO) command.
- For objects in other file systems, the label is **SAVyyyymmdd**.

character-value

Specify the identifier (up to 17 characters) of the tape file used for the save operation.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

***** The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

***END** The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

***UNLOAD**

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

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

***NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing is complete.

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

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

***NONE**

No notification message is sent.

***WRKSTN**

The notification message is sent to the work station message queue.

path-name

Specify the path name of the message queue to be used.

Top

Synchronization ID (SYNCID)

Specifies the name of the synchronized checkpoint in which this save while active operation will participate. The synchronized checkpoint must already be started by the Start Save Synchronization (STRSAVSYNC) command.

*NONE

The checkpoint for this save while active operation is not synchronized with any other save while active operation.

name Specify the name of the synchronized checkpoint. If you specify a name, you must also specify a value of *SYNC for the **Save active (SAVACT)** parameter.

Top

Type of output information (INFTYPE)

Specifies the type of information that is directed to the spooled file, stream file, or user space.

*ALL The file will contain information about the command, an entry for each directory, an entry for each object that was successfully saved, and an entry for each object that was not successfully saved.

*ERR The file will contain information about the command, an entry for each directory, and an entry for each object that was not successfully saved.

*SUMMARY

The file will contain information about the command, and an entry for each directory.

Top

System (SYSTEM)

Specifies whether to process objects that exist on the local system or remote systems.

*LCL Only local objects are processed.

*RMT Only remote objects are processed.

*ALL Both local and remote objects are processed.

Top

Time period for last change (CHGPERIOD)

Specifies a date/time range. Objects that were last changed within that range will be saved.

Element 1: Start date

*ALL No starting date is specified. All objects last changed prior to the ending date will be saved.

*LASTSAVE

The objects that have changed since the last time they were saved with UPDHST(*YES) specified are saved. **Notes:**

1. If this value is specified, the value *ALL must be specified for all other elements of this parameter.
2. For local file systems, the system archive attribute is used. For remote file systems, the PC archive attribute is used.

date Specify the date after which objects that have changed are to be saved. The date must be specified in job date format.

Element 2: Start time

***ALL** All times of day are included in the range.

time Specify the time on the start date after which objects that have changed are to be saved.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the starting date is an explicit date.

Element 3: End date

***ALL** No ending date is specified. All objects changed since the starting date will be saved.

date Specify the date before which objects that have changed are to be saved. The date must be specified in the job date format.

Element 4: End time

***ALL** All times of day are included in the range.

time Specify a time on the end date before which objects that have changed are to be saved.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the ending date is an explicit date.

Top

Object pre-check (PRECHK)

Specifies whether the save operation ends if any of the selected objects cannot be saved.

***NO** The save operation does not end. Objects that can be saved are saved.

***YES** The save operation ends. Nothing is saved unless all of the selected objects can be saved.

Top

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Update history (UPDHST)

Specifies whether to update the save history on the objects saved with this save operation. The save history information is used when CHGPERIOD(*LASTSAVE) is specified on a later save operation.

Single values

***NO** This save operation will not be updated in the save history of the selected objects.

***YES** The save history is updated. For local file systems, the system save history is updated. For remote file systems, the PC save history is updated.

Other values (up to 2 repetitions)

***SYS** The system save history is updated.

***PC** The PC save history is updated.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.

3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE**

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

*MEDIUM

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

*HIGH

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation.

***DFT** The operation uses the ASPDEV value appropriate for the file system from which objects are being saved. For Integrated File System objects, *ALLAVL is used. For objects from the QSYS file system, the corresponding save command ASPDEV default is used.

*ALLAVL

The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs.

***** The operation includes the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Scan objects (SCAN)

Specifies whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points and whether objects that previously failed a scan should be saved.

The integrated file system scan-related exit points are:

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

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

Element 1: Scan during save

***NO** Objects will not be scanned by the scan-related exit programs.

***YES** Objects will be scanned according to the rules described in the scan-related exit programs.

Element 2: Save failed objects

***NOSAVFAILED**

Objects that have either previously failed a scan or that fail a scan by a QIBM_QP0L_SCAN_OPEN exit program during this save will not be saved.

***SAVFAILED**

Objects that have either previously failed a scan or that fail a scan during this save will be saved.

Top

Private authorities (PVTAUT)

Specifies whether to save private authorities with the objects that are saved. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the recovery of an object or a group of objects. It will not simplify the recovery of an entire system.

***NO** No private authorities are saved.

***YES** Private authorities are saved for each object that is saved.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

Top

Examples

Example 1: Saving All Data Not in Libraries and Not Document Library Objects

```
SAV DEV('/QSYS.LIB/TAP01.DEVD')
    OBJ('//*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT)
```

This command saves all objects that are not in libraries and are not document library objects.

Example 2: Saving Changes Since the Last Time the Objects Were Saved

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
      CHGPERIOD(*LASTSAVE)
```

This command saves all objects that are not in libraries, that are not document library objects, and that have changed since the last time they were saved with UPDHST(*YES) specified.

Example 3: Saving Data That Has Not Changed Since 1999

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
      CHGPERIOD((*ALL *ALL '12/31/99'))
```

This command saves all objects that are not in libraries, that are not document library objects, and that have not changed since December 31, 1999.

Example 4: Saving All Objects in the Current Directory

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD')
```

This command uses the default value `**` on the OBJ parameter to save all objects in the current directory and its subdirectories. This example is not valid if the current directory is the root directory or if the current directory is in the QDLS file system.

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD') OBJ('*') SUBTREE(*NONE)
```

This command saves all objects in the current directory but not in the subdirectories.

Example 5: Omitting Objects During a Save Operation

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/*') ('**.*BACKUP' *OMIT) ('**.*TEMP' *OMIT))
```

This command saves all objects in the current directory except those with extensions of `.BACKUP` and `.TEMP` (the entire subtrees of directories with these extensions are omitted).

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD')
      OBJ('/A') ('/A/B/C' *OMIT))
```

This command saves all objects in directory `/A` and its subdirectories, except those in directory `/A/B/C`.

Example 6: Saving a Library

```
SAV  DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QSYS.LIB/A.LIB')
```

This command saves library A on the tape device named TAP01.

Example 7: Saving Two Libraries

```
SAV DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QSYS.LIB/A.LIB')
SAV DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QSYS.LIB/B.LIB')
```

These two commands save two libraries. The first command saves library A and the second command saves library B.

Example 8: Saving All Files in a Library

```
SAV DEV('/QSYS.LIB/TAP01.DEVD')
   OBJ('/QSYS.LIB/MYLIB.LIB/*.FILE')
```

This command saves all files in the library MYLIB on the tape device named TAP01.

Example 9: Saving Two Objects in a Library

```
SAV DEV('/QSYS.LIB/TAP01.DEVD')
   OBJ('/QSYS.LIB/MYLIB.LIB/MYPGM.PGM')
SAV DEV('/QSYS.LIB/TAP01.DEVD')
   OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
```

These two commands save two objects in the same library. This first command saves the program MYPGM from library MYLIB. The second command saves the file MYFILE from library MYLIB.

Example 10: Saving a Stream File, a Database File, and a Document

```
SAV DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/MyDir/MyFile')
SAV DEV('/QSYS.LIB/TAP01.DEVD')
   OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
SAV DEV('/QSYS.LIB/TAP01.DEVD')
   OBJ('/QDLS/MYFLR/MYDOC') SUBTREE(*OBJ) UPDHST(*YES)
```

Three commands are used in this example to save three objects. The first command saves the stream file MyFile in the directory MyDir on the tape device named TAP01. The second command saves the database file MYFILE in the library named MYLIB on the tape device named TAP01. The third command saves the document MYDOC in a folder named MYFLR on the tape device named TAP01.

Example 11: Saving to a Save File

```
SAV DEV('/QSYS.LIB/MYLIB.LIB/MYSAVF.FILE') OBJ(MYDIR)
```

This command saves the directory MYDIR to a save file named MYSAVF.

Example 12: Using Symbolic Links for the Save Operation

```
SAV DEV('DevLink')
   OBJ(('DirLink') ('FileLink') ('DirLink/*'))
   SAVACT(*YES) SAVACTMSGQ('MsgqLink')
```

This command assumes that the current directory contains the following symbolic links:

- DevLink = /QSYS.LIB/TAP01.DEVD

- DirLink = /SomeDirectory
- FileLink = /SomeDirectory/SomeFile
- MsgqLink = /QSYS.LIB/LIB1.LIB/MSGQ1.MSGQ

This command saves the names associated with DirLink and FileLink, and the objects in SomeDirectory, to device TAP01. A message is sent when the save-while-active checkpoint is complete.

Symbolic links can be used to specify a device, a save-while-active message queue, and an output file. When symbolic links are specified to be saved, only the names of the associated objects are saved, not the content of the associated objects. A symbolic link to a directory can be used to save objects in the directory. Additional information about symbolic links is in the **Integrated file system** topic in the **File systems and management** category of the Information Center.

Example 13: Saving a Storage Space While it is Active

```
SAV  DEV('/QSYS.LIB/MYLIB.LIB/MYSAVF.FILE')
      OBJ('/QFPNWSSTG/MYDISK')
      SAVACT(*YES) SAVACTOPT(*NWSSTG)
```

This command saves the objects associated with storage space '/QFPNWSSTG/MYDISK' even if the associated *NWSSTG is varied on.

Example 14: Saving Shares from a Windows, Linux or AIX Server

```
SAV  DEV('/QSYS.LIB/MYLIB.LIB/MYSAVF.FILE')
      OBJ('/QNTC/MYSERVER/MYSHARE')
```

This command saves share MYSHARE and all of the objects the share contains from a server named '/QNTC/MYSERVER'. The example assumes that server MYSERVER was properly configured prior to running the SAV command. For more information on file level backup for Windows, Linux and AIX servers, see the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Error messages

*ESCAPE Messages

CPFA0DB

Object not a QSYS.LIB object. Object is &1.

CPFA0DC

Object not a QDLS object. Object is &1.

CPF3708

Save file &1 in &2 too small.

CPF3727

Duplicate device &1 specified on device name list.

CPF3735

Storage limit exceeded for user profile &1.

CPF3738

Device &1 used for save or restore is damaged.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF3782
File &1 in &2 not a save file.

CPF3794
Save or restore operation ended unsuccessfully.

CPF37B9
Synchronization ID &1 in use.

CPF37BC
Synchronization ID &1 ended. Wait time exceeded.

CPF37BD
Synchronization ID &1 ended. Save ended before checkpoint.

CPF37BE
Synchronization ID &1 not started.

CPF380D
Save or restore of entire system completed unsuccessfully.

CPF3808
Save file &2 in &3 not complete.

CPF381D
Values on CHGPERIOD parameter not valid.

CPF3812
Save file &1 in &2 in use.

CPF382B
Parameters not valid with multiple file systems.

CPF382C
OBJ parameter value not valid for QSYS file system.

CPF382E
Specified parameter not valid for QDLS file system.

CPF382F
OBJ parameter value not valid for QDLS file system.

CPF3823
No objects saved or restored.

CPF3826
*INCLUDE object required on OBJ parameter.

CPF3828
Error occurred while attempting to use &1.

CPF3829
Specified parameter not valid for QSYS file system.

CPF383A
Save or restore ended unsuccessfully.

CPF383B
End of file &1.

CPF383D
Cannot use &1.

CPF3833
Specified value on DEV parameter not valid.

CPF3834
Too many values specified on the DEV parameter.

CPF3835
Tape devices do not support same densities.

CPF3837
&1 objects saved. &2 not saved.

CPF3838
&1 objects saved. &2 objects not saved.

CPF384A
Volume identifier &1 not valid.

CPF384B
Optical file specified not valid.

CPF384C
Error occurred during CCSID conversion.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF386B
Not able to save &1.

CPF386C
Not able to save &1.

CPF3894
Cancel reply received for message &1.

CPF38A5
Error on the PATTERN parameter.

CPF5729
Not able to allocate object &1.

CPF9802
Not authorized to object &2 in &3.

CPF9825
Not authorized to device &1.

Top

Save APAR Data (SAVAPARDTA)

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

Parameters
Examples
Error messages

The Save APAR Data (SAVAPARDTA) command allows the user to save information required for an Authorized Problem Analysis Report (APAR).

Restrictions:

- The following user profiles have private authorities to use the command:
 - QPGMR
 - QSYSOPR
 - QSRV
 - QSRVBAS

Top

Parameters

Keyword	Description	Choices	Notes
PRBID	Problem identifier	<i>Character value</i> , *NEW	Required, Positional 1

Top

Problem identifier (PRBID)

Specifies the identifier (ID) of the problem for which APAR data is to be saved.

***NEW** An open problem log record is created to track this APAR.

character-value

Specify the ID of the problem for which APAR data is to be saved.

Top

Examples

```
SAVAPARDTA PRBID(*NEW)
```

This command creates an open problem log for which APAR data is saved. The user selects the data to be saved by indicating the choices on a list display. This data is saved in an APAR library.

Top

Error messages

*ESCAPE Messages

CPF2182

Not authorized to library &1.

CPF39FA

Problem &1 &2 &3 not found

CPF39FF

SAVAPARDTA command can not be run

CPF39F2

Cannot allocate library &1

CPF39F4

No more APAR data can be saved for this problem

CPF39F5

Query of problem &1 failed

CPF39F6

Problem record could not be created

CPF39F7

Could not create user space in APAR library

CPF39F8

Problem &1 could not be associated with an APAR library

CPF39F9

Problem &1 &2 &3 in use

Top

Save Configuration (SAVCFG)

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

Parameters
Examples
Error messages

The Save Configuration (SAVCFG) command saves all configuration and system resource management (SRM) objects without requiring a system in a restricted state. The information saved includes the following:

- Line descriptions
- Controller descriptions
- Device descriptions
- Mode descriptions
- Class-of-service descriptions
- Network interface descriptions
- Network server descriptions
- NetBIOS descriptions
- Connection lists
- Configuration lists
- Hardware resource data
- Token-ring adaptor data

Information saved can be restored with the Restore Configuration (RSTCFG) command.

Restrictions:

- You must have save system (*SAVSYS) special authority to run this command.
- System resource management (SRM) objects are not saved if a Work with Hardware Products (WRKHDWPRD) job is running at the same time.

Top

Parameters

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 1
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	1-16777215, *END	Optional
EXPDATE	File expiration date	<i>Date</i> , *PERM	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
USEOPTBLK	Use optimum block	*YES, *NO	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	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional

Keyword	Description	Choices	Notes
PRECHK	Object pre-check	*NO, *YES	Optional
DTACPR	Data compression	*DEV, *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	*DEV, *NO	Optional
OUTPUT	Output	*NONE, *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<i>*REPLACE, *ADD</i>	

Top

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

***MOUNTED**

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

***END** The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

*LEAVE

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

*UNLOAD

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

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

***NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

* The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name!'

 The system generates an optical file name in the specified directory of the optical volume.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save configuration operation ends if any of the objects satisfy the following conditions:

- The objects were previously found to be damaged.
- The objects are locked by another job.
- The user does not have authority to save the objects.

***NO** The save operation continues, saving only configuration and system resource management (SRM) objects that can be saved.

***YES** The save operation ends before any data is written to the media if any configuration objects or system resource manager objects cannot be saved.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

*MEDIUM

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

*HIGH

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

***NONE**
No output listing is created.

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

***OUTFILE**
The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASAVOBJ with format name QSRSAV as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Examples

Example 1: Saving Objects

```
SAVCFG DEV(TAP01) CLEAR(*ALL)
```

This command saves system resource management objects (hardware resource data and token-ring adaptor data) and all configuration objects (including all line, controller, device, mode, class-of-service, and network descriptions, configuration lists, and connection lists). They are saved on the TAP01 tape drive. CLEAR(*ALL) automatically clears all uncleared tapes when they are encountered.

Example 2: Saving Objects to a Specific Tape

```
SAVCFG DEV(TAP01) VOL(ABC)
```

This command saves the SRM and configuration objects on the TAP01 tape drive, starting on the tape volume labeled ABC. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be put on the TAP01 tape drive is issued.

Error messages

*ESCAPE Messages

CPF2206

User needs authority to do requested function on object.

CPF222E

&1 special authority is required.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3737

Save and restore data area &1 not found.

CPF376D

Not all configuration objects saved to save file &3.

CPF376E

Not all configuration objects saved.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF3782

File &1 in &2 not a save file.

CPF3793

Machine or ASP storage limit reached.

CPF3794

Save or restore operation ended unsuccessfully.

CPF3812

Save file &1 in &2 in use.

CPF384E

USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF388B

Optical file path name not valid.

CPF3894

Cancel reply received for message &1.

CPF5729

Not able to allocate object &1.

CPF9809

Library &1 cannot be accessed.

CPF9812

File &1 in library &2 not found.

CPF9845

Error occurred while opening file &1.

CPF9846

Error while processing file &1 in library &2.

CPF9847

Error occurred while closing file &1 in library &2.

CPF9850

Override of printer file &1 not allowed.

CPF9851

Overflow value for file &1 in &2 too small.

CPF9860

Error occurred during output file processing.

Top

Save Changed Objects (SAVCHGOBJ)

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

Parameters
Examples
Error messages

The Save Changed Object (SAVCHGOBJ) command saves a copy of each changed object or group of objects located in the same library. When *ALL is specified for the **Objects (OBJ)** parameter, objects can be saved from all user libraries or from a list of libraries. When saving to a save file, only one library can be specified. For database files, only the changed members are saved.

Objects changed since the specified date and time are saved with the following exceptions:

- If OBJJRN(*NO) is specified, objects currently being journaled are not saved, unless journaling was started after the specified date and time. This ensures that changes made to an object before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, and logical file definitions are saved, but the contents of those objects are not saved. Logical file access paths are saved if ACCPTH(*YES) is specified. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter.

Specified objects that were changed and the libraries where they reside remain locked during the save operation.

Saved objects can be restored with the Restore Object (RSTOBJ) command.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved by this command are listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. The system saves the changed objects by writing a copy of each one on tapes, optical volumes, or a save file. The description of each object is changed with the date, time, and place when it was last saved and is controlled by the UPDHST parameter.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- You must either have save system (*SAVSYS) special authority or the following object authorities:
 - object existence (*OBJEXIST) authority for each object to be saved
 - execute (*EXECUTE) authority to each library objects are saved from
- If you do not have *SAVSYS special authority, only those changed objects that you have authority for are saved.
- When saving to a tape or optical device, you must have use (*USE) authority to the device description and device file. When saving to a save file, you must have object operational (*OBJOPR) and add (*ADD) authorities to the save file, and *EXECUTE authority to the library where the save file is located.
- When using a media definition, you must have *USE authority to the media definition and *EXECUTE authority to the media definition library.

- If tape is used, a standard labeled volume designation must be used.
- No changed object that is being saved can be changed by another job that is running when the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file, by specifying SAVFDTA(*YES), only the description of the save file is saved.
- When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the library where the file is located.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
LIB	Library	Single values: *ALLUSR, *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
DEV	Device	Single values: *SAVE, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
OBJTYPE	Object types	Single values: * <u>ALL</u> Other values (up to 300 repetitions): <i>Character value</i>	Optional
OBJJRN	Journalized objects	* <u>NO</u> , *YES	Optional
REFDATE	Reference date	<i>Date</i> , * <u>SAVLIB</u>	Optional
REFTIME	Reference time	<i>Time</i> , * <u>NONE</u>	Optional
VOL	Volume identifier	Single values: * <u>MOUNTED</u> Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	1-16777215, * <u>END</u>	Optional
LABEL	Label	<i>Character value</i> , * <u>LIB</u>	Optional
EXPDATE	File expiration date	<i>Date</i> , * <u>PERM</u>	Optional
ENDOPT	End of media option	* <u>REWIND</u> , *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	<i>Name</i> , * <u>FIRST</u>	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
USEOPTBLK	Use optimum block	* <u>YES</u> , *NO	Optional
TGTRLS	Target release	* <u>CURRENT</u> , *PRV, V5R3M0, V5R4M0, V6R1M0	Optional
UPDHST	Update history	* <u>YES</u> , *NO	Optional
CLEAR	Clear	* <u>NONE</u> , *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	* <u>NO</u> , *YES	Optional
SAVACT	Save active	* <u>NO</u> , *LIB, *SYNCLIB, *SYSDFN	Optional

Keyword	Description	Choices	Notes
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, 120 , *NOMAX	
	Element 2: Pending record changes	0-99999, * LOCKWAIT , *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, * LOCKWAIT , *NOMAX	
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	Name, * NONE , *WRKSTN	
	Qualifier 2: Library	Name, * LIBL , *CURLIB	
SYNCID	Synchronization ID	Name, * NONE	Optional
ACCPATH	Save access paths	* SYSVAL , *NO, *YES	Optional
SAVFDTA	Save file data	* YES , *NO	Optional
QDTA	Queue data	* NONE , *DTAQ	Optional
PVTAUT	Private authorities	* NO , *YES	Optional
DTACPR	Data compression	* DEV , *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	* DEV , *NO	Optional
OMITLIB	Libraries to omit	Single values: * NONE , *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name</i> , * NONE , *ALL	
	Qualifier 2: Library	<i>Generic name, name</i> , * ALL	
	Element 2: Object type	<i>Character value</i> , * ALL	
ASPDEV	ASP device	Name, *, *SYSBAS, *CURASPGRP	Optional
OUTPUT	Output	* NONE , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	Name	
	Qualifier 2: Library	Name, * LIBL , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	Name, * FIRST	
	Element 2: Replace or add records	* REPLACE , *ADD	
INFTYPE	Type of output information	* OBJ , *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	<i>Qualified object name</i>	Optional
	Qualifier 1: Command user space	Name	
	Qualifier 2: Library	Name, * LIBL , *CURLIB	

Top

Objects (OBJ)

Specifies the names of one or more objects, or the generic names of each group of objects, to check for changes and then to save those objects that have changed. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

***ALL** All changed objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of changed objects to save in the specified library. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name Specify the names of specific objects to save. Both generic names and specific names can be specified in the same command.

Top

Library (LIB)

Specifies the library that contains the changed objects to be saved.

Note: Libraries are saved alphabetically by ASP device name for each value specified. Libraries on independent ASPs are saved before libraries on the system and basic user ASPs. For value *ALLUSR, libraries QSYS2, QGPL, QUSRSYS, and QSYS2xxxxx (where xxxxx is an independent ASP number) are saved first, if they are located on the ASPs specified by the ASPDEV parameter.

This is a required parameter.

Single values

***ALLUSR**

All user libraries are saved. All libraries with names that do not begin with the letter Q are saved 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 saved:

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIJS	QUSRVRxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	

QMOMDATA	QUSRADSM	QUSRPOSSA
QMOMPROC	QUSRBRM	QUSRPYMSVR
QPFRDATA	QUSRDIRCF	QUSRDRARS
QRCL	QUSRDIRCL	QUSRSYS

Note: 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.

*USRSPC

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A generic library name cannot be specified when saving to a save file.

name If *ALL is specified for the **Objects (OBJ)** parameter, up to 300 library names can be specified. Only one library can be specified when saving to a save file.

Top

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

*MEDDFN

The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are

used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

Object types (OBJTYPE)

Specifies the types of system objects whose changes are saved. The object types saved are also the ones saved and restored by the Save Library (SAVLIB), Restore Object (RSTOBJ), and Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command.

Single values

***ALL** Changes to all object types that are specified by name, and which are in the specified library, are saved.

Other values (up to 300 repetitions)

object-type

Specify the value for each of the types of objects that are saved, such as command (*CMD), file (*FILE), or program (*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Journalled objects (OBJJRN)

Specifies whether to save changed objects that are currently being journaled and that have been journaled since the date and time specified for the **Reference date (REFDATE)** and **Reference time (REFTIME)** parameters.

***NO** Objects being journaled are not saved. If journaling was started after the specified date and time, the changed objects or changed database file members are saved. The date and time of the last journal start operation can be shown by using the Display Object Description (DSPOBJD) command.

***YES** Objects whose changes are entered in a journal are saved.

Top

Reference date (REFDATE)

Specifies the reference date. Objects that have been changed since this date are saved.

***SAVLIB**

The objects that have been changed since the date of the last running of the Save Library (SAVLIB) command are saved. If the specified library was never saved, a message is issued and the library is not saved, but the operation continues.

date Specify the reference date; objects that have been changed since this date are saved. If you specify a date later than the date of the running of this command, a message is issued and the operation ends. The date must be specified in the job date format.

Reference time (REFTIME)

Specifies the reference time. Objects that have been changed since this time on the specified date are saved.

*NONE

No explicit time is specified. Any objects changed since the date specified for the **Reference date (REFDATE)** parameter are saved.

time Specify the reference time; objects that have been changed since this time on the specified date are saved. If *SAVLIB is specified for the REFDATE parameter, no reference time can be specified. If you specify a time later than the time of the running of this command, a message is issued and the operation ends.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 6 digits (hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

Specify the sequence number of the file to be used for the save operation.

Top

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

***LIB** The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value

Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

*UNLOAD

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

Top

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

If an unrecoverable media error occurs during the save operation, this parameter can be used to restart the operation.

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

```
STRLIB(library-name) OMITLIB(library-name)
```

where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

*FIRST

The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name Specify the name of the library with which to begin the save operation.

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Recovering your system book, SC41-5304, and the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

Qualifier 1: Media definition

name Specify the name of the media definition to be used.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

***** The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.

- The value for the DTACPR parameter is ignored.
- *NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:

- If the program object was created at a release level more current than the targeted earlier release, you must:
 1. create the program object again specifying the targeted earlier release
 2. save the program object specifying the targeted earlier release
 3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
 1. save the program object specifying the targeted earlier release
 2. restore the program object on the target system.

For non-program objects:

You can:

1. save the object specifying the targeted earlier release
2. restore the object on the target system.

*CURRENT

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

- *PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Notes:

1. If LIB(*ALLUSR) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the chart in the Recovering your system book, SC41-5304.

Top

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journalized Changes (APYJRNCHG) command.

***YES** The last save date, time, and location is updated in each object saved.

***NO** The save history information contained in the description of each object saved is not updated.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE**

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the

save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

***NO** The save operation for a library continues, saving only those objects that can be saved.

***YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYNCLIB**

Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: If you specify this value and you are saving many libraries, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.

***SYSDFN**

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object lock before continuing the save operation.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value ***NOCMTBDY** is specified.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

***NOCMTBDY**

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify ***NOCMTBDY**.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.

- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Top

Synchronization ID (SYNCID)

Specifies the name of the synchronized checkpoint in which this save while active operation will participate. The synchronized checkpoint must already be started by the Start Save Synchronization (STRSAVSYNC) command.

*NONE

The checkpoint for this save while active operation is not synchronized with any other save while active operation.

name Specify the name of the synchronized checkpoint. If you specify a name, you must also specify a value of *SYNCLIB for the **Save active (SAVACT)** parameter.

Note: If you specify a name, the value used for the **Save active wait time (SAVACTWAIT)** parameter **Element 2: Pending record changes** is the largest value specified among all of the participating save operations. However, if any participating save operation specifies *NOCMTBDY, then all participating save operations must specify *NOCMTBDY.

Top

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Top

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

***NONE**

Only the description of a queue is saved.

***DTAQ**

The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save private authorities with the objects that are saved. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the recovery of an object or a group of objects. It will not simplify the recovery of an entire system.

***NO** No private authorities are saved.

***YES** Private authorities are saved for each object that is saved.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

***MEDIUM**

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

***HIGH**

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

***NONE**

No libraries are excluded from the save operation.

***USRSPC**

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the

generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Single values

***USRSPC**

The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

***ALL** The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

***ALL** All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

*NONE

No output listing is created.

*PRINT

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASAVOBJ with format name QRSABV as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Type of information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

***OBJ** The list contains an entry for each object requested to be saved.

***ERR** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

***LIB** The list contains a library entry for each library requested to be saved

***MBR** The list contains an entry for each object, database file member, and spooled file requested to be saved.

Top

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library where the user space is located.

Top

Examples

Example 1: Saving Changed Files

```
SAVCHGOBJ OBJ(ORD*) LIB(DSTPRODLB) DEV(TAP01)
          OBJTYPE(*FILE) REFDATE(122290)
```

This command saves all files with names that start with the characters ORD in the library named DSTPRODLB that were changed since December 22, 1990.

Example 2: Saving Description and Data for Save Files

```
SAVCHGOBJ OBJ(FILE*) LIB(MYLIB) DEV(TAP01) OBJTYPE(*FILE)
          REFDATE(122290) SAVFDATA(*YES)
```

This command saves all files with names that start with the characters FILE* in the library named MYLIB that were changed since December 22, 1990. It also saves the description and the data for all save files that match this selection criteria.

Top

Error messages

*ESCAPE Messages

CPF3702

&1 objects saved from &3. &2 not saved. &9 not included.

CPF3703

&2 &1 in &3 not saved.

CPF3708
Save file &1 in &2 too small.

CPF3709
Tape devices do not support same densities.

CPF3727
Duplicate device &1 specified on device name list.

CPF3728
Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3745
No record of SAVLIB operation exists for &1.

CPF3746
System date and time earlier than reference date and time.

CPF3747
Object names cannot be specified with more than one library.

CPF3749
Objects from library &2 not saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF377F
Save-while-active request prevented by pending record changes.

CPF3770
No objects saved or restored for library &1.

CPF3774
&1 objects saved from &3. &2 not saved. &8 not included.

CPF3778
Not all objects saved from all libraries.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF37AB
*NOCMTBDY not allowed with target release.

CPF37AC
Library not allowed with *NOCMTBDY.

CPF37B4
User space &1 in &2 not valid.

CPF37B5
PVTAUT not allowed with target release.

CPF37B7
Not authorized to save private authorities.

CPF37B9
Synchronization ID &1 in use.

CPF37BC
Synchronization ID &1 ended. Wait time exceeded.

CPF37BD
Synchronization ID &1 ended. Save ended before checkpoint.

CPF37BE
Synchronization ID &1 not started.

CPF37C7
Synchronization ID &1 ended. SAVACTWAIT error.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF3818
Starting library &1 not found.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF388B
Optical file path name not valid.

CPF3892
&2 &1 in &3 not saved.

CPF3894
Cancel reply received for message &1.

CPF38A2
ASP device &1 not correct.

CPF38A3
File &1 in &2 not valid with ASPDEV.

CPF38A4
ASP device &1 not correct.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9825
Not authorized to device &1.

CPF9833
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPFB8ED
Device description &1 not correct for operation.

***STATUS Messages**

CPF3770
No objects saved or restored for library &1.

CPF3774

&1 objects saved from &3. &2 not saved. &8 not included.

CPF3871

No objects saved or restored; &3 objects not included.

[Top](#)

Save Document Library Object (SAVDLO)

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

Parameters
Examples
Error messages

The Save Document Library Object (SAVDLO) command saves a copy of the specified documents, folders, or distribution objects (mail).

Notes:

- When a folder is saved, the folder object is saved along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved individually using DLO(*FLRLVL).
- Distribution objects (mail) cannot be saved or restored for individual users. Mail can be saved only for all users.
- SAVDLO does not require a dedicated system; however, individual objects in use when the save is issued cannot be saved. To ensure all document library objects are saved, run this command when no document or folder activity is occurring on the system.

Restrictions:

- You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority to use the following parameter combinations on this command:
 - DLO(*ALL) FLR(*ANY)
 - DLO(*CHG)
 - DLO(*MAIL)
 - DLO(*SEARCH) OWNER(*ALL)
 - DLO(*SEARCH) OWNER(*user-profile-name*)where the *user profile name* specified is not the *user profile name* of the user issuing the SAVDLO command.
- If you do not have *ALLOBJ or *SAVSYS special authority, you must:
 - Have all (*ALL) authority for each document or folder to be saved
 - Be enrolled as a Document Interchange Architecture (DIA) user
- Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.
- When using the OUTFILE parameter to save to an existing database file, you must have execute authority to the output file library.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SEARCH, *CHG, *SYSOBJNAM, *DOCL, *MAIL, *FLRLVL Other values (up to 300 repetitions): <i>Character value</i>	Required, Positional 1
FLR	Folder	Single values: *ANY, *NONE Other values (up to 300 repetitions): <i>Character value</i>	Optional

Keyword	Description	Choices	Notes
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
SRCHTYPE	Search type	* <u>DOC</u> , *ALL	Optional
CHKFORMRK	Check for mark	* <u>NO</u> , *YES	Optional
CHKEXP	Expiration date	<i>Date</i> , * <u>NO</u> , *CURRENT	Optional
CRTDATE	Creation date	<i>Element list</i>	Optional
	Element 1: Starting time and date	<i>Element list</i>	
	Element 1: Starting time	<i>Time</i> , * <u>AVAIL</u>	
	Element 2: Starting date	<i>Date</i> , * <u>BEGIN</u> , *CURRENT	
	Element 2: Ending time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time</i> , * <u>AVAIL</u>	
	Element 2: Ending date	<i>Date</i> , * <u>END</u>	
DOCCLS	Document class	<i>Character value</i> , * <u>ANY</u>	Optional
OWNER	Owner profile	<i>Name</i> , * <u>CURRENT</u> , *ALL	Optional
REFCHGDATE	Last changed date	<i>Date</i> , * <u>ANY</u> , *SAVDLOALL	Optional
REFCHGTIME	Last changed time	<i>Time</i> , * <u>ANY</u>	Optional
SYSOBJNAM	System object name	Values (up to 300 repetitions): <i>Name</i>	Optional
DOCL	Document list	<i>Character value</i> , * <u>NONE</u>	Optional
VOL	Volume identifier	Single values: * <u>MOUNTED</u> Other values (up to 75 repetitions): <i>Character value</i>	Optional
SEQNBR	Sequence number	1-16777215, * <u>END</u>	Optional
EXPDATE	File expiration date	<i>Date</i> , * <u>PERM</u>	Optional
ENDOPT	End of media option	* <u>REWIND</u> , *LEAVE, *UNLOAD	Optional
LABEL	Label	<i>Character value</i> , * <u>GEN</u>	Optional
USEOPTBLK	Use optimum block	* <u>YES</u> , *NO	Optional
OPTFILE	Optical file	<i>Path name</i> , _	Optional
SAVF	Save file	<i>Qualified object name</i>	Optional
	Qualifier 1: Save file	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
OMITFLR	Folders to omit	Single values: * <u>NONE</u> Other values (up to 300 repetitions): <i>Character value</i>	Optional
OUTPUT	Output	* <u>NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , * <u>LIBL</u> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name</i> , * <u>FIRST</u>	
	Element 2: Replace or add records	* <u>REPLACE</u> , *ADD	
CLEAR	Clear	* <u>NONE</u> , *AFTER, *ALL, *REPLACE	Optional
STG	Storage	* <u>KEEP</u> , *DELETE, *FREE	Optional
DTACPR	Data compression	* <u>DEV</u> , *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	* <u>DEV</u> , *NO	Optional

Keyword	Description	Choices	Notes
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEVVD Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	
TGTRLS	Target release	*CURRENT, *PRV, V5R3M0, V5R4M0, V6R1M0	Optional
SAVACT	Save active	*NO, *YES	Optional
SAVACTWAIT	Save active wait time	0-99999, 120, *NOMAX	Optional
ASP	ASP number	1-32, *ANY	Optional

Top

Document library object (DLO)

Specifies the document library objects to save. To save a folder, DLO(*ALL) must be specified.

This is a required parameter.

Single values

***ALL** All document library objects further qualified by the FLR parameter are to be saved. Specifying DLO(*ALL) FLR(*ANY) saves all document library objects.

*SEARCH

All document library objects that meet the specified search values are saved. Search values are specified by using the following parameters:

- **Folder (FLR)** parameter.
- **Check for mark (CHKFORMRK)** parameter.
- **Expiration date (CHKEXP)** parameter.
- **Creation date (CRTDATE)** parameter.
- **Document class (DOCCLS)** parameter.
- **Owner profile (OWNER)** parameter.
- **Last changed date (REFCHGDATE)** parameter.
- **Last changed time (REFCHGTIME)** parameter.

Note: Folders are saved only if SRCHTYPE(*ALL) is specified.

***CHG** All documents created or changed and all folders created since the last complete save operation and all mail is saved.

*SYSOBJNAM

The documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are saved.

*DOCL

The list of documents referred to in a document list specified for the **Document list (DOCL)** parameter is saved.

*MAIL

The distribution objects and documents referred to by a mail log are saved.

*FLRLVL

The folders specified for the **Folder (FLR)** parameter and documents in the folders are saved. Subfolders are not saved.

Other values (up to 300 repetitions)

document-name

Specify the user-assigned names of the documents that are saved. All documents specified must be in the same folder and that folder must be specified for the **Folder (FLR)** parameter.

Top

Folder (FLR)

Specifies the name of the folder to save.

Single values

***ANY** Document library objects can be saved from any folder. Consider the following when using the FLR parameter:

- FLR(*ANY) is not valid when one of the following is specified:
 - DLO(*DOCL)
 - DLO(*FLRLVL)
 - DLO(*document-name*)
- FLR(*ANY) is required when one of the following is specified:
 - DLO(*CHG)
 - DLO(*SYSOBJNAM)
 - DLO(*MAIL)
 - DLO(*SEARCH) SRCTYPE(*ALL)
- When SAVDLO DLO(*ALL) FLR(*ANY) is specified, the following are saved:
 - All documents
 - All folders
 - All distribution objects (mail)

***NONE**

The documents saved are not in any folder. FLR(*NONE) is valid only when one of the following is specified:

- DLO(*ALL)
- DLO(*SEARCH) SRCTYPE(*DOC)

Other values (up to 300 repetitions)

generic-folder-name

Specify a generic name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

folder-name

Specify the user-assigned name of the folder in which the documents to be saved are located. The folder name can be a maximum of 63 characters in length.

- Folder objects specified here are saved only when DLO(*ALL) or DLO(*FLRLVL) is specified.
- FLR(*folder-name*) is not valid when one of the following is specified:
 - DLO(*SYSOBJNAM)
 - DLO(*MAIL)
 - DLO(*SEARCH) SRCTYPE(*ALL)

- Only one folder name can be specified when one of the following is specified:
 - DLO(*DOCL)
 - DLO(*SEARCH) SRCHTYPE(*DOC)
 - DLO(*document-name*)

Top

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Top

Search type (SRCHTYPE)

Specifies the type of objects for which to search. This parameter is valid only if *SEARCH is specified for the **Document library object (DLO)** parameter.

***DOC** Only documents are to be searched and saved.

***ALL** Documents and folders are to be searched and saved.

Top

Check for mark (CHKFORMRK)

Specifies whether documents marked for storage are saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

***NO** Documents that meet the other search values for this save operation are saved regardless of whether they are marked for storage.

***YES** Only those documents that meet the other search values and are also marked for storage are saved. Documents may be marked:

- Keep
- Free
- Delete

Top

Expiration date (CHKEXP)

Specifies that all documents with an expiration date before the specified date are to be saved. The expiration date is assigned by the user when a document is created to specify when the document is no longer needed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCTYPE)** parameter.

***NO** The expiration date is ignored.

***CURRENT**

All documents with an expiration date before today's date are saved.

date Specify a document expiration date. All documents with an expiration date before this date are saved.

Top

Creation date (CRTDATE)

Specifies that documents and folders that have a creation date during the time period specified are to be saved. The time period is specified by a starting time and date and an ending time and date. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter. The time period is specified as follows:

((start-time start-date) (end-time end-date))

Element 1: Starting time and date

Element 1: Starting time

Use one of the following to specify the starting time. Documents must have been created after this time to be selected. Documents created before this time are not selected.

***AVAIL**

Documents and folders filed at any time are eligible for saving.

time Specify the starting time. When the starting time is used as a search value, the starting date must not be *BEGIN. The starting-time must be the same as the value specified for the **Last changed time (REFCHGTIME)** parameter when the REFCHGTIME parameter is specified. The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Element 2: Starting date

Use one of the following to specify the starting date. Documents must have been created on or after this date to be saved. Documents created before this date are not saved.

*BEGIN

Documents and folders are saved regardless of the creation date associated with the object.

*CURRENT

Only documents and folders filed on today's date, after the starting time (if specified), are selected.

date Specify a starting date for the document creation date time period. The date must be entered in the job date format. The start date must be the same as the value specified for the **Last changed date (REFCHGDATE)** parameter when the REFCHGDATE parameter is specified.

Element 2: Ending time and date

Element 1: Ending time

Use one of the following to specify the ending time. Documents must have been created before this time to be saved. Any documents created after the specified time are not saved.

*AVAIL

Documents and folders filed at any time are selected for saving.

time Specify the ending time. When the ending time is to be used as a search value, the ending date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 2: Ending date

Use one of the following to specify the ending date. Documents must have been created on or before this date to be saved. Documents created after this date are not saved.

*END Documents and folders filed on any date are selected. The ending time is not allowed when *END is specified.

date Specify the ending date for the document creation date time period. Documents created on or before this date are saved. The date must be specified in job date format.

Top

Document class (DOCCLS)

Specifies the class of documents to be saved. The class is assigned by the user when the document is created. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

Note: Although document classes are user-assigned, double-byte character set (DBCS) data cannot be specified on this parameter.

*ANY The document class is not used to select documents for saving.

character-value

Specify the document class, ranging from 1 through 16 characters, used to select documents for saving.

Top

Owner profile (OWNER)

Specifies the owner of the documents and folders to be saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*CURRENT

Documents and folders owned by the current requester are saved.

***ALL** This parameter is not used to select documents and folders for saving. You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority if *ALL is specified.

name Specify the name of the user profile that owns the documents and folders to be saved. All documents and folders owned by this user and that meet the other search values specified are saved. *ALLOBJ or *SAVSYS special authority is required if a user profile is selected other than the user profile of the user issuing this command.

Top

Last changed date (REFCHGDATE)

Specifies the date after which the folders that are created and the documents that are changed or created are to be saved. The change date is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object(DLO)** parameter.

*ANY No reference change date is specified. Documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

***SAVDLOALL**

Folders that have been created and documents created or changed since that last complete save operation are saved.

date Specify the date after which the created folders or the created or changed documents are saved.

Top

Last changed time (REFCHGTIME)

Specifies the time, relative to the date specified for the REFCHGDATE parameter, after which the folders that are created and the documents that are changed or created are to be saved. The change time is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*ANY No time is specified. The documents are saved regardless of the time they were created or changed. Folders are saved regardless of the time they were created.

time Specify the time after which the created folders or the created or changed documents are saved.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from

the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

System object name (SYSOBJNAM)

Specifies the system object names of the documents that are saved, when *SYSOBJNAM is specified for the **Document library object (DLO)** parameter.

Single values

*NONE

A system object name is not specified.

Other values (up to 300 repetitions)

name Specify the system object name of the document to be saved. A full ten characters must be specified.

Top

Document list (DOCL)

Specifies a list of documents to be saved. The document list must be in a folder. The name of the folder must be specified using the **Folder (FLR)** parameter. You must have use (*USE) authority to the folders containing the documents in the document list.

Note: A document list must be the result of a local search, not a remote search.

*NONE

No document list is saved.

character-value

Specify the document list to be saved.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

***END** The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

***UNLOAD**

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

Label (LABEL)

Specifies the name that identifies the data file on the tape used for the save. If the LABEL parameter is used the label must be specified on the restore command.

***GEN** The file label is created by the system.

character-value

Specify the data file identifier that is used as the label for the data file used for the save operation. A maximum of 17 characters can be specified.

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

***NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

***** The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Folders to omit (OMITFLR)

Specifies the names of one or more folders, or the generic names of each group of folders, to be excluded from the save operation.

Single values

***NONE**

No folders are excluded from the save operation.

Other values (up to 300 repetitions)

generic-folder-name

Specify a generic name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all folders with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete folder name.

folder-name

Specify the name of the folder to be excluded from the save operation.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

***NONE**

No output listing is created.

***PRINT**

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the name and library of the database file to which the information about the document library objects is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter. If the file does not exist, this command creates a database file in the specified library. If a new file is created, the system uses QAOJSAVO in QSYS with a format name of QJSDLO as a model.

Qualifier 1: File to receive output

name Specify the name of the database file to which output from the command is directed. If this file does not exist, it is created in the specified library.

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

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

*FIRST

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

*REPLACE

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Storage (STG)

Specifies whether system storage occupied by the document being saved is kept, deleted, or freed after the save operation ends.

Note: STG(*DELETE) and STG(*FREE) are not valid when any of the following are specified:

- DLO(*ALL) FLR(*ANY)
- DLO(*SEARCH) CHKFORMRK(*YES)

- DLO(*CHG)
- DLO(*MAIL)
- SAVACT(*YES)

***KEEP**

The storage occupied by the document remains unchanged after the save operation.

***DELETE**

The document object and all search terms are deleted from the system after the save operation.

***FREE** The document description and search terms remain on the system but the storage occupied by the document is deleted after the save operation. The document cannot be used until the document is restored to the system.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

***MEDIUM**

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

***HIGH**

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data specified for the **Document class (DOCCLS)** parameter. The character identifier is related to the display device used to enter the document class.

Single values

***SYSVAL**

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEV**

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

***NO** Document library objects in use are not saved. Document library objects cannot be updated while being used.

***YES** Document library objects can be changed during the save request.

Note: Some applications update document library objects directly. The data is supplied to the application rather than saving the updates in a temporary file and then updating the DLOs. DLOs that are being updated directly (typically, those being updated by PC-based applications) will not be saved. See the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information on using this parameter.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time for an object that is in use, before continuing the save operation. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object to become available before continuing the save operation.

***NOMAX**
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for each individual object before continuing the save operation.

Top

ASP number (ASP)

Specifies the number of the auxiliary storage pool (ASP) of the document library object (DLO) to be saved.

***ANY** The objects to be saved reside in any ASP. When DLO(*ALL) FLR(*ANY) are specified, all document library objects on the system are saved.

Note:When DLOs from multiple ASPs are saved, multiple tape media files are created. The beginning and ending sequence numbers of these media files will be required on the RSTDLO command to restore all ASPs.

1-32 Specify the number of an existing ASP that contains the document library objects to be saved. ASP 1 is the system ASP.

Note: Mail that has not been filed and documents that are not in a folder reside in the system ASP.

Top

Examples

Example 1: Performing a Complete Save Operation

```
SAVDLO DLO(*ALL) FLR(*ANY) DEV(TAP01)
```

This command saves all folders, documents, and mail to the tape device TAP01.

Example 2: Saving All Changes

```
SAVDLO DLO(*CHG) DEV(TAP01)
```

This command saves all documents created or changed since the last complete save operation, folders created since the last complete save operation, and all mail.

Example 3: Saving An Entire ASP

```
SAVDLO DLO(*ALL) DEV(TAP01) FLR(*ANY) ASP(2)
```

This command saves all folders and documents in user ASP 2.

Example 4: Saving Objects Changed After a Specific Date

```
SAVDLO DLO(*SEARCH) DEV(TAP01) OWNER(*ALL)  
REFCHGDATE('01/01/2002')
```

This command saves all documents changed or created after January 1, 2002. This command is useful for saving changes between backups of the documents. This command is similar to the Save Changed Objects (SAVCHGOBJ) used for other object types.

Example 5: Saving Documents and Folders Changed After a Specific Date

```
SAVDLO DLO(*SEARCH) DEV(TAP01) SRCHTYPE(*ALL) OWNER(*ALL)  
REFCHGDATE('01/01/2002')
```

This command saves all folders created since 01/01/2002 and all documents created or changed since 01/01/2002.

Example 6: Saving Documents Created After a Specific Date

```
SAVDLO DLO(*SEARCH) DEV(TAP01)
        CRTDATE((*AVAIL '01/01/2002')) OWNER(*ALL)
```

This command saves all documents created or changed since 01/01/2002.

Example 7: Saving Documents and Folders Created After a Specific Date

```
SAVDLO DLO(*SEARCH) DEV(TAP01) SRCHTYPE(*ALL)
        CRTDATE((*AVAIL '01/01/2002')) OWNER(*ALL)
```

This command saves all documents and folders created since 01/01/2002.

Example 8: Freeing System Storage During the Save Operation

```
SAVDLO DLO(DOCX) FLR(FOLDERA) DEV(TAP01) STG(*FREE)
```

This command saves the document named DOCX in folder FOLDERA to the tape device TAP01. As part of the save operation, the system storage that was occupied by the data portion of the document is freed.

Example 9: Saving Folders

```
SAVDLO DLO(*ALL) DEV(*SAVF) FLR(BILL TOM MARY) SAVF(SAVF1)
```

This command saves the folders BILL, TOM, and MARY, and all the documents in these folders.

Example 10: Saving Just the Folder Level of a Folder

```
SAVDLO DLO(*FLRLVL) FLR(DEC01) DEV(*SAVF) SAVF(MYLIB/SAVF1)
```

This command saves folder DEC01 in folder STATUS and all the documents in this folder to save file SAVF1 in library MYLIB. Subfolders in this folder are not saved.

Example 11: Saving Just Mail

```
SAVDLO DLO(*MAIL) DEV(*SAVF)
```

This command saves all distribution objects and all documents referred to by a mail log.

Top

Error messages

***ESCAPE Messages**

CPF3728

Device &1 specified with other devices.

CPF3733

&2 &1 in &3 previously damaged.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF3793
Machine or ASP storage limit reached.

CPF3812
Save file &1 in &2 in use.

CPF384D
Save or restore operation not allowed on ASP &1.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF8AC3
ASP &5 is not valid.

CPF8A47
Internal system objects in use.

CPF90B2
List of folder names not valid with DLO parameter.

CPF90C1
Document list &1 empty.

CPF90C2
Document list &1 not used.

CPF90D7
Documents not saved.

CPF90E1
Folder &1 cannot be used with TGTRLS(*PRV).

CPF90E5
Not authorized to document list &1.

CPF9006
User not enrolled in system distribution directory.

CPF902E
&1 document library objects saved. &10 not saved.

CPF903B
Too many objects for save file.

CPF903C
No document library objects saved.

CPF9030
Owner profile (&1) not found.

CPF9046
No documents found satisfying search specification in folder &1.

CPF9053
Not authorized to requested function.

CPF9056
Not authorized to requested function.

CPF906B
No document library objects saved.

CPF908A
Requester &1 not enrolled.

CPF9096
Cannot use CMDCHRID(*DEVVD), DOCCHRID(*DEVVD) in batch job.

CPF941B
Save data area &1 in use.

CPF9410
&1 document library objects saved. &10 not saved.

CPF9411
Changes for ASP &1 not saved.

CPF9417
CRTDATE value not valid with REFCHGDATE or REFCHGTIME.

CPF9418
SAVF parameter not valid.

CPF9419
LABEL(*GEN) required.

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.

CPF9831
Cannot assign device &1.

CPF9845
Error occurred while opening file &1.

CPF9846
Error while processing file &1 in library &2.

CPF9850
Override of printer file &1 not allowed.

CPF9851
Overflow value for file &1 in &2 too small.

CPF9860
Error occurred during output file processing.

CPF9899
Error occurred during processing of command.

Save Library (SAVLIB)

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

Parameters
Examples
Error messages

The Save Library (SAVLIB) command allows you to save a copy of one or more libraries.

When saving to a save file, only one library can be specified.

This command saves the entire library, including the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, and logical files, only the object definitions are saved, not the contents. However, logical file access paths can be saved by specifying *YES for the **Save access paths (ACCPTH)** parameter. The contents of spooled files on output queues can be saved by specifying *ALL for the **Spooled file data (SPLFDTA)** parameter. The contents of a save file can be saved by specifying *YES for the **Save file data (SAVFDTA)** parameter or using the Save Save File Data (SAVSAVFDTA) command. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter. The libraries and their objects are not affected in the system unless the command specifies that the storage is to be freed. However, unless *NO is specified for the **Update history (UPDHST)** parameter, the description of each library and each object is updated with the date, place, and time it was last saved. If a group of libraries is saved by specifying *NONSYS, *ALLUSR, or *IBM for the LIB parameter, the date, time, and place are updated in the history information for a data area in QSYS (data area QSAVLIBALL, QSAVALLUSR, or QSAVIBM).

The types of objects saved by this command are the same as those listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, with the addition of *DTADCT. Certain operating system objects that are not contained in user libraries (such as user profiles) are not saved by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDTA) commands.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- You must either have save system (*SAVSYS) special authority or the following object authorities:
 - read (*READ) authority for, or be the owner of, each library specified
 - object existence (*OBJEXIST) authority for each object in the library (for spooled file data, *OBJEXIST authority for the output queue)
- If you do not have *SAVSYS special authority, only those libraries and objects that you have authority for are saved.
- When saving to a tape or optical volume, you must have use (*USE) authority to the associated device description.
- When saving to a save file, you must have add (*ADD) and use (*USE) authorities to the save file.
- When using a media definition, you must have *USE authority to the media definition and execute (*EXECUTE) authority to the library where the media definition is located.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the output file library.
- No library being saved, or the objects in it, can be updated by a job that is running at the time the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.

- When the contents of a save file are saved, by specifying *YES for the **Save file data (SAVFDTA)** parameter, the save file must be restored before objects contained in it can be restored.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Single values: *NONSYS, *ALLUSR, *IBM, *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 2
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 3
SEQNBR	Sequence number	1-16777215, *END	Optional
LABEL	Label	<i>Character value</i> , *LIB	Optional
EXPDATE	File expiration date	<i>Date</i> , *PERM	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
STRLIB	Starting library	<i>Name</i> , *FIRST	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	
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
USEOPTBLK	Use optimum block	*YES, *NO	Optional
TGTRLS	Target release	*CURRENT, *PRV, V5R3M0, V5R4M0, V6R1M0	Optional
UPDHST	Update history	*YES, *NO	Optional
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	*NO, *YES	Optional
SAVACT	Save active	*NO, *LIB, *SYNCLIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, 120, *NOMAX	
	Element 2: Pending record changes	0-99999, *LOCKWAIT, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, *LOCKWAIT, *NOMAX	
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	<i>Name</i> , *NONE, *WRKSTN	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
SYNCID	Synchronization ID	<i>Name</i> , *NONE	Optional
ACCPATH	Save access paths	*SYSVAL, *NO, *YES	Optional
SAVFDTA	Save file data	*YES, *NO	Optional
SPLFDTA	Spooled file data	*NONE, *ALL	Optional
QDTA	Queue data	*NONE, *DTAQ	Optional

Keyword	Description	Choices	Notes
PVTAUT	Private authorities	<u>*NO</u> , *YES	Optional
STG	Storage	<u>*KEEP</u> , *FREE	Optional
DTACPR	Data compression	<u>*DEV</u> , *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	<u>*DEV</u> , *NO	Optional
OMITLIB	Libraries to omit	Single values: <u>*NONE</u> , *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *NONE</i> , *ALL	
	Qualifier 2: Library	<i>Generic name, name, *ALL</i>	
Element 2: Object type	<i>Character value, *ALL</i>		
ASPDEV	ASP device	<i>Name, *</i> , *SYSBAS, *CURASPGRP	Optional
OUTPUT	Output	<u>*NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i> , *CURLIB	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *FIRST</i>	
	Element 2: Replace or add records	<u>*REPLACE</u> , *ADD	
INFTYPE	Type of output information	<u>*OBJ</u> , *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	<i>Qualified object name</i>	Optional
	Qualifier 1: Command user space	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *LIBL</i> , *CURLIB	

Top

Library (LIB)

Specifies which libraries are saved.

Notes:

1. The system libraries QDOC, QDOCxxxx, QRCYxxxx, QRECOVERY, QRPLOBJ, QRPLxxxx, QSPL, QSPLxxxx, QSRV, QSYS, QSYSxxxx, and QTEMP cannot be saved with this command. xxxx is a basic user ASP number, and xxxxx is an independent ASP number.
2. If you specify *ALLUSR or *IBM on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved, the objects are not saved unless you use SAVACT. To ensure a complete save of all libraries, run this command with the system in a restricted state. For example, if SAVLIB LIB(*ALLUSR) is run when the subsystem QSNADS is active, the QAO* files are not saved in library QUSRSYS. To save the QAO* files, end the QSNADS subsystem before running SAVLIB LIB(*ALLUSR).
3. Doing a SAVLIB LIB(*IBM) and then doing a SAVLIB LIB(*ALLUSR) saves the same libraries as a SAVLIB LIB(*NONSYS), but requires two restore commands.
4. Libraries are saved alphabetically by ASP device name for each value specified. Libraries on independent ASPs are saved before libraries on the system and basic user ASPs. For values *NONSYS

and *ALLUSR, libraries QSYS2, QGPL, QUSRSYS, and QSYS2xxxxx (where xxxxx is an independent ASP number) are saved first, if they are located on the ASPs specified by the ASPDEV parameter.

This is a required parameter.

Single values

*NONSYS

All libraries except the following system libraries are saved. The xxxx and xxxxx represent ASP numbers.

QDOC	QRPLOBJ	QSRV
QDOCxxxx	QRPLxxxxx	QSYS
QRCYxxxxx	QSPL	QSYSxxxxx
QRECOVERY	QSPLxxxx	QTEMP

Note: All subsystems must be ended by the End Subsystem (ENDSBS) or End System (ENDSYS) command before this option is specified.

*ALLUSR

All user libraries are saved. All libraries with names that do not begin with the letter Q are saved 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 saved:

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

*IBM Saves all IBM-supplied libraries except for the following:

QDOC	QRCYxxxxx	QTEMP	QUSRPYMSVR
QDOCxxxx	QRECOVERY	QUSER38	QUSRDRARS
QDSNX	QRPLOBJ	QUSRADSM	QUSRSYS
QGPL	QRPLxxxxx	QUSRBRM	QUSRVI
QGPL38	QSPL	QUSRDIRCF	QUSRVxRxMx
QMGTC	QSPLxxxxx	QUSRDIRCL	
QMGTC2	QSRV	QUSRDIRDB	
QMPGDATA	QSRVAGT	QUSRIJS	
QMOMDATA	QSYS	QUSRINFSKR	
QMOMPROC	QSYSxxxxx	QUSRNOTES	
QPFRDATA	QSYS2	QUSROND	
QRCL	QSYS2xxxxx	QUSRPOSGS	
QRCLxxxxx	QS36F	QUSRPOSSA	

Note: 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.

The following libraries with names that do not begin with the letter Q are also saved:

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

*USRSPC

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be saved. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be saved. Only one library name can be saved when saving to a save file. A system library name cannot be specified.

Top

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

*SAVF The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

*MEDDFN

The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

*MOUNTED

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

*END The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

If *NONSYS, *ALLUSR, or *IBM is specified for the LIB parameter, the save operation for the set of libraries begins at the sequence number specified. The first file saved in this set is the QFILE file. The QFILE file contains the list of libraries saved.

Top

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

*LIB The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value

Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

*PERM

The file is protected permanently.

date Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

If an unrecoverable media error occurs during the save operation, this parameter can be used to restart the operation.

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

```
STRLIB(library-name) OMITLIB(library-name)
```

where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

***FIRST**

The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name Specify the name of the library with which to begin the save operation.

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Recovering your system book, SC41-5304, and the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

Qualifier 1: Media definition

name Specify the name of the media definition to be used.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Top

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

***** The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Top

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

***NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by

the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:

- If the program object was created at a release level more current than the targeted earlier release, you must:
 1. create the program object again specifying the targeted earlier release
 2. save the program object specifying the targeted earlier release
 3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
 1. save the program object specifying the targeted earlier release
 2. restore the program object on the target system.

For non-program objects:

You can:

1. save the object specifying the targeted earlier release
2. restore the object on the target system.

***CURRENT**

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Notes:

1. If LIB(*NONSYS), LIB(*ALLUSR), or LIB(*IBM) is specified, only the current release can be the target release.
2. Not all objects can be targeted to another release. To find out which objects are supported, see the chart in the Recovering your system book, SC41-5304.

Top

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journalized Changes (APYJRNCHG) command.

***YES** The last save date, time, and location is updated in each object saved.

***NO** The save history information contained in the description of each object saved is not updated.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

- *NO** The save operation for a library continues, saving only those objects that can be saved.
- *YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

- *NO** Objects that are in use are not saved. Objects cannot be updated while being saved.
- *LIB** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.
- *SYNCLIB**
Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.
- Note:** If you specify this value and you are saving many libraries, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.
- *SYSDFN**
Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

- 120** The system waits up to 120 seconds for each individual object lock before continuing the save operation.
- *NOMAX**
No maximum wait time exists.
- 0-99999**
Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

***NOCMTBDY**

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Top

Synchronization ID (SYNCID)

Specifies the name of the synchronized checkpoint in which this save while active operation will participate. The synchronized checkpoint must already be started by the Start Save Synchronization (STRSAVSYNC) command.

*NONE

The checkpoint for this save while active operation is not synchronized with any other save while active operation.

name Specify the name of the synchronized checkpoint. If you specify a name, you must also specify a value of *SYNCLIB for the **Save active (SAVACT)** parameter.

Note: If you specify a name, the value used for the **Save active wait time (SAVACTWAIT)** parameter **Element 2: Pending record changes** is the largest value specified among all of the participating save operations. However, if any participating save operation specifies *NOCMTBDY, then all participating save operations must specify *NOCMTBDY.

Top

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

***SYSVAL**

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Top

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

***NONE**

No spooled file data is saved.

***ALL** For each output queue that is saved, all available spooled file data on the output queue is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

***NONE**

Only the description of a queue is saved.

***DTAQ**

The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save private authorities with the objects that are saved. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the recovery of an object or a group of objects. It will not simplify the recovery of an entire system.

***NO** No private authorities are saved.

***YES** Private authorities are saved for each object that is saved.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

Top

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

***KEEP**

The storage occupied by the data portion of the objects being saved is not freed.

***FREE** The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

***MEDIUM**

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

***HIGH**

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

***NONE**

No libraries are excluded from the save operation.

***USRSPC**

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Single values

***USRSPC**

The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

*ALL All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

*NONE

No output listing is created.

*PRINT

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

*OUTFILE

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

Top

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASAVOBJ with format name QRSASV as a model.

Top

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Top

Type of information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

***OBJ** The list contains an entry for each object requested to be saved.

***ERR** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.

***LIB** The list contains a library entry for each library requested to be saved

***MBR** The list contains an entry for each object, database file member, and spooled file requested to be saved.

Top

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library where the user space is located.

Top

Examples

Example 1: Saving a Library on a Tape Device

```
SAVLIB LIB(JOE) DEV(TAP01)
```

This command saves the library named JOE on the tape that is in the tape device named TAP01. The storage occupied by JOE in the system is not freed, because the STG parameter default (*KEEP) was assumed.

Example 2: Saving on Multiple Volumes

```
SAVLIB LIB(QGPL) DEV(OPT01) VOL(ABC DEF GHI)
```

The general purpose library (QGPL) is saved on the optical volumes in the device named OPT01. The optical volumes used must have the volume names ABC, DEF, and GHI. If the save operation is not finished when volume ABC is full, a message is issued to the operator asking for volume DEF to be placed in the device.

Example 3: Freeing Storage when Saving Data

```
SAVLIB LIB(CUSTDATA) DEV(TAP01)  
VOL(CUSTNM CUSTAD) STG(*FREE)
```

The library named CUSTDATA is saved on volumes CUSTNM and CUSTAD, which are put in the tape device TAP01. The storage occupied by the specified members, modules, programs, service programs, SQL packages, and journal receivers in the CUSTDATA library is freed after it is saved.

Example 4: Saving on Multiple Devices Serially

```
SAVLIB LIB(USRLIB) DEV(TAP01 TAP02 TAP03)
VOL(USRA USRB USRC USRD) ENDOPT(*UNLOAD)
```

The library named USRLIB is saved on four tape volumes on three tape devices. The volume named USRA is put on the device named TAP01, the volume named USRB on the device named TAP02, the volume named USRC on the device named TAP03, and the volume named USRD on the device named TAP01. The volume named USRA is rewound, and must be unloaded by the operator when processing is complete so that the device named TAP01 can be used for the volume named USRD.

Example 5: Saving on Multiple Devices in Parallel

```
SAVLIB LIB(USRLIB) DEV(*MEDDFN) MEDDFN(LIBA/MEDDFNA)
```

The library named USRLIB is saved on the devices specified in the media definition named MEDDFNA in library LIBA. For information about creating and using a media definition, see the **Backup, Recovery, and Availability** topic in the Information Center.

Example 6: Saving a Library with a Media File Label

```
SAVLIB LIB(LIB1) DEV(TAP01) LABEL(MONDAYBACKUP)
```

This command uses the tape device named TAP01 to save the library named LIB1 on tape. The library is saved with a media file label of MONDAYBACKUP. This label must be specified when restoring the library or any of its objects.

Example 7: Specifying Where the Save Operation Starts

```
SAVLIB LIB(*NONSYS) DEV(TAP01 TAP02)
STRLIB(MIKESLIB) ASPDEV(*SYSBAS)
```

This command saves all libraries that are located on the system and basic user ASPs, beginning with the library named MIKESLIB on tape devices named TAP01 and TAP02. System libraries and libraries that are normally saved first (QSYS2, QGPL, and QUSRSYS) are not saved.

Example 8: Saving User Libraries in an Independent ASP Group

```
SAVLIB LIB(*ALLUSR) DEV(TAP01) ASPDEV(*CURASPGRP)
```

This command saves all user libraries on the independent ASPs that are in the ASP group for the current thread. It does not save libraries on the system ASP (ASP number 1) or on basic user ASPs (ASP numbers 2-32).

Example 9: Saving Spooled Files

```
SAVLIB LIB(MYLIB) DEV(TAP01) SPLFDTA(*ALL)
```

This command saves the library named MYLIB, including the spooled file data for any output queues in the library.

Error messages

*ESCAPE Messages

CPF3701

&1 objects saved from &3. &2 not saved.

CPF3703

&2 &1 in &3 not saved.

CPF3708

Save file &1 in &2 too small.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3730

Not authorized to &2 &1 in library &3.

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3735

Storage limit exceeded for user profile &1.

CPF3738

Device &1 used for save or restore is damaged.

CPF3749

Objects from library &2 not saved.

CPF3751

Some libraries not saved.

CPF3767

Device &1 not found.

CPF3768

Device &1 not valid for command.

CPF377D

Save ended because of read error on internal system resource.

CPF377E

Not enough storage for save-while-active request.

CPF377F

Save-while-active request prevented by pending record changes.

CPF3770

No objects saved or restored for library &1.

CPF3771

&1 objects saved from &3. &2 not saved.

CPF3777
&1 libraries saved, &6 partially saved, &2 not saved.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3785
Not all subsystems ended.

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF37AB
*NOCMTBDY not allowed with target release.

CPF37AC
Library not allowed with *NOCMTBDY.

CPF37B1
SPLFDTA not allowed with target release.

CPF37B4
User space &1 in &2 not valid.

CPF37B5
PVTAUT not allowed with target release.

CPF37B7
Not authorized to save private authorities.

CPF37B9
Synchronization ID &1 in use.

CPF37BC
Synchronization ID &1 ended. Wait time exceeded.

CPF37BD
Synchronization ID &1 ended. Save ended before checkpoint.

CPF37BE
Synchronization ID &1 not started.

CPF37C7
Synchronization ID &1 ended. SAVACTWAIT error.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF3818
Starting library &1 not found.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3871
No objects saved or restored; &3 objects not included.

CPF388B
Optical file path name not valid.

CPF3892
&2 &1 in &3 not saved.

CPF3894
Cancel reply received for message &1.

CPF38A2
ASP device &1 not correct.

CPF38A3
File &1 in &2 not valid with ASPDEV.

CPF38A4
ASP device &1 not correct.

CPF5729
Not able to allocate object &1.

CPF9809
Library &1 cannot be accessed.

CPF9812
File &1 in library &2 not found.

CPF9814
Device &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9825
Not authorized to device &1.

CPF9833
*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.

CPFB8ED
Device description &1 not correct for operation.

OPT1332

Optical volume not found in device &1.

STATUS Messages*CPF3770**

No objects saved or restored for library &1.

CPF3771

&1 objects saved from &3. &2 not saved.

CPF3871

No objects saved or restored; &3 objects not included.

[Top](#)

Save Licensed Program (SAVLICPGM)

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

Parameters
 Examples
 Error messages

The Save Licensed Program (SAVLICPGM) command saves a copy of all of the objects that make up a licensed program. It saves the licensed program in a form that can be restored by the Restore Licensed Program (RSTLICPGM) command.

Restrictions:

1. To use this command, you must have *ALLOBJ authority or have specific authority from the security officer.
2. A standard labeled tape volume must be on the tape device.
3. Some licensed programs can only be saved if the user is enrolled in the system distribution directory. See the publication for each licensed program for a description of this restriction.
4. This command does not save code or language objects for the base operating system.
5. At most one optical device, one save file, one virtual tape device or one tape media library device can be specified.

Top

Parameters

Keyword	Description	Choices	Notes
LICPGM	Product	Character value	Required, Positional 1
DEV	Device	Single values: *SAVF Other values (up to 4 repetitions): Name	Required, Positional 2
OPTION	Optional part to be saved	*BASE, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 01, 02, 03, 04, 05, 06, 07, 08, 09	Optional
RLS	Release	Character value, *ONLY	Optional
LNG	Language for licensed program	Character value, *PRIMARY, *ALL	Optional
OBJTYPE	Object type	*ALL, *PGM, *LNG	Optional
CHKSIG	Check signature	*SIGNED, *ALL, *NONE	Optional
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): Character value	Optional
SEQNBR	Sequence number	1-16777215, *END	Optional
EXPDTE	File expiration date	Date, *PERM	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	Optional
SAVF	Save file	Qualified object name	Optional
	Qualifier 1: Save file	Name	
	Qualifier 2: Library	Name, *LIBL, *CURLIB	
TGTRLS	Target release	*CURRENT, *PRV, V5R2M0, V5R3M0, V5R4M0	Optional

Keyword	Description	Choices	Notes
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional
DTACPR	Data compression	*DEV, *NO, *YES	Optional
LICACPRQD	License acceptance required	*NO, *YES	Optional

Top

Product (LICPGM)

Specifies the seven-character identifier of the licensed program that is saved.

This is a required parameter.

Top

Device (DEV)

Specifies the names of the tape or optical devices used for the save licensed program operation. Each device name must already be known on the system by a device description. Use the Work with Device Descriptions (WRKDEVD) command to display the names of the devices available on this system.

This is a required parameter.

The possible values are:

***SAVF** The product is saved in the save file specified on the save file (SAVF) parameter.

optical-device-name

Specify the name of the optical device to be used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the name of one or more tape devices to use for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume.

Top

Optional part to be saved (OPTION)

Specifies the optional parts of the licensed program given in the **Product** prompt (LICPGM parameter) that are saved.

The possible values are:

***BASE**

Only the base part of the licensed program is to be saved.

number-of-licensed-program-option

Specify the number of the optional part of the listed licensed program that is to be saved.

Top

Release (RLS)

Specifies which version, release, and modification level of the licensed program is saved.

The possible values are:

***ONLY**

Only one version, release, and modification level is installed for the licensed program 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.

Top

Language for licensed program (LNG)

Specifies which national language version (NLV) is used for the save operation.

Note: This parameter is ignored when OBJTYPE(*PGM) is specified.

The possible values are:

***PRIMARY**

The primary language is saved. The primary language is the language of the operating system.

Note: Use GO LICPGM with option 20 to display the primary language of the operating system.

***ALL** All languages are saved.

feature-code

Specify the NLV identifier for the language that is saved for the licensed program.

Top

Object type (OBJTYPE)

Specifies the type of licensed program objects being saved.

The possible values are:

***ALL** Program and language objects specified on the LNG parameter are saved.

***PGM** Only the program objects for the licensed program are saved.

***LNG** The objects associated with the NLV identified on the LNG parameter are saved.

Top

Check signature (CHKSIG)

Specifies if the digital signatures of objects being saved with the licensed program are to be checked.

The possible values are:

***SIGNED**

Objects with digital signatures are checked. Objects that are not digitally signed will not have their signatures verified. Any signed object with a signature that is not valid will be identified in a message in the job log and the save will fail.

***ALL** All objects that can be digitally signed are checked. Any object that can be signed but has no signature will be identified in a message in the job log. The save will not be terminated for unsigned objects. Any signed object with an invalid digital signature will be identified in a message in the job log and the save will fail.

***NONE**

Digital signatures of objects will not be checked.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the tape volumes on which the licensed program data is saved. The volumes must be placed in the same order as the volume identifiers are specified on this parameter. Up to 75 entries can be specified.

The possible values are:

***MOUNTED**

The volume currently placed in the device is used.

For a media library device, the volume to be used is the next cartridge in the the category mounted by the Set Tape Category (SETTAPCGY) command.

Note: This value cannot be specified when using an optical media library device.

volume-identifier

Specify the identifiers of one or more volumes in the order in which they are placed in the device and used to save the licensed program.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

***END** The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the tape created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

*REWIND

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

*LEAVE

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

*UNLOAD

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

Top

Save file (SAVF)

Specifies the qualified name of the existing save file to which the product is saved. The save file must be empty, unless *ALL is specified on the Clear prompt (CLEAR 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 searched. If no library is specified as the current library for the job, the QGPL library is used.

library-name

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

The possible values are:

save-file-name

Specify the name of the save file.

Top

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to restore and use the product. In the examples given for the *CURRENT and *PRV values, and when specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V4R1M0 is version 4 release 1 modification level 0.

Note: This value is limited by the minimum operating system release specified when the product was created.

The possible values are:

***CURRENT**

The product is to be restored to, and used on, the release of the operating system currently running on your system. The product can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The product is to be restored to the previous release with modification level 0 of the operating system. The product can also be restored to a system with any subsequent release of the operating system installed.

target-release

Specify the release in the format VxRxMx. The product can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Valid values depend on the current version, release, and modification level, and they change with each new release. Press F4 to see a list of valid target release values.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

***NONE**

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

***AFTER**

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

***REPLACE**

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

The possible values are:

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

***MEDIUM**

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

***HIGH**

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

License acceptance required (LICACPRQD)

Specifies whether or not the user will be prompted for license acceptance when the saved licensed program is restored. If the licensed program being saved does not have online software agreements, no prompt will be done at restore time. This parameter does not affect the license acceptance status of the licensed program currently installed.

The possible values are:

***NO** License acceptance will not be required when the saved licensed program is restored.

***YES** License acceptance will be required when the saved licensed program is restored, if the licensed program being saved has online software agreements.

Examples

Example 1: Tapes Cleared Automatically

```
SAVLICPGM LICPGM(5761WDS) DEV(TAP01) CLEAR(*ALL)
```

This command saves the *BASE option of the WebSphere Development Studio licensed program (5761WDS). It is saved on the tape that is in the TAP01 tape drive. Each uncleared tape is cleared automatically when it is encountered, and the save operation continues without operator intervention.

Example 2: Saving on Labeled Tape Volume

```
SAVLICPGM LICPGM(5761WDS) DEV(TAP01) VOL(ABCDE)
```

The *BASE option of the WebSphere Development Studio licensed program (5761WDS) is saved on the TAP01 tape drive, starting on the tape volume labeled ABCDE. If the save operation exceeds the storage capacity of one tape, a message requesting that another volume be placed in the TAP01 tape drive is shown to the operator.

Example 3: Saving on Multiple Volumes

```
SAVLICPGM LICPGM(5761SS1) OPTION(2) DEV(TAP01 TAP02)
```

The online help option of the i5/OS licensed program (5761SS1) is saved on tape drives TAP01 and TAP02 in alternating order. If the save operation exceeds the storage capacity of two tapes, a message requesting that another volume be put on TAP01 is shown to the operator. The tapes are rewound at the completion of the save operation.

Example 4: Saving to Save Files Cleared Automatically

```
SAVLICPGM LICPGM(5761WDS) DEV(*SAVF)  
          CLEAR(*ALL) SAVF(MYLIB/MYSAVF)
```

This command saves the base option of the WebSphere Development Studio licensed program (5761WDS). It is saved to the save file MYSAVF in the library MYLIB. The save file MYSAVF is cleared automatically and the save operation continues without operator intervention.

Example 5: Saving program objects to a save file

```
SAVLICPGM LICPGM(1MYPROD) OPTION(*BASE)  
          DEV(*SAVF) RLS(*ONLY)  
          OBJTYPE(*PGM) SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and does not have any folders. Only the program objects for 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 6: Saving all objects in one library to a save file


```
SAVLICPGM LICPGM(1MYPROD) OPTION(*BASE)
          DEV(*SAVF) RLS(*ONLY)
          LNG(2924) OBJTYPE(*ALL) SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of only one library and does not have any folders. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 7: Saving all objects in two libraries to a save file

```
SAVLICPGM LICPGM(1MYPROD) OPTION(*BASE)
          DEV(*SAVF) RLS(*ONLY)
          LNG(2924) OBJTYPE(*ALL) SAVF(MYLIB/MYSAVF)
```

The 1MYPROD product consists of two libraries and has one root folder with the *PGM objects, and one subfolder of that root folder with the *LNG objects. The program objects and language objects for the NLV 2924 of 1MYPROD product will be saved in the save file, MYSAVF, in MYLIB library.

Example 8: Saving for a previous release.

```
SAVLICPGM LICPGM(1MYPROD) DEV(TAP01) TGTRLS(*PRV)
```

The 1MYPROD product will be saved to TAP01 and may be restored on a system with the previous release of the operating system installed.

[Top](#)

Error messages

*ESCAPE Messages

CPF3D93

Objects for product &1 option &2 release &4 not saved.

[Top](#)

Save Object (SAVOBJ)

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

Parameters
Examples
Error messages

The Save Object (SAVOBJ) command saves a copy of a single object or a group of objects located in the same library. When *ALL is specified for the **Objects (OBJ)** parameter, objects can be saved from a list of libraries. When saving to a save file, only one library can be specified. The system saves the specified objects by writing a copy of each object on tape or optical media, or in a save file. The objects are not affected in the system unless the command specifies that the storage should be freed. However, the description of each object is changed with the date, time, and place when it was last saved, unless *NO is specified for the **Update history (UPDHST)** parameter.

For job queues, user queues, message queues, and logical files, only the object descriptions are saved, and the contents of the objects are not saved. However, logical file access paths can be saved by specifying *YES for the **Save access paths (ACCPH)** parameter. The contents of spooled files on output queues can be saved by specifying *ALL for the **Spooled file data (SPLFDTA)** parameter. The contents of a save file can be saved by specifying *YES for the **Save file data (SAVFDTA)** parameter or using the Save Save File Data (SAVSAVFDTA) command. The contents of a data queue can be saved by specifying *DTAQ for the **Queue data (QDTA)** parameter.

Note: This command ignores all file overrides currently in effect for the job, except for the output file.

Restrictions:

- You must either have save system (*SAVSYS) special authority or the following object authorities:
 - object existence (*OBJEXIST) authority for each object to be saved (for spooled file data, *OBJEXIST authority for the output queue)
 - execute (*EXECUTE) authority to each library objects are saved from
- If you do not have *SAVSYS special authority, only those objects that you have authority for are saved.
- When saving to a tape or optical device, you must have use (*USE) authority to the device description and device file. When saving to a save file, you must have object operational (*OBJOPR) and add (*ADD) authorities to the save file, and *EXECUTE authority to the library where the save file is located.
- When using a media definition, you must have *USE authority to the media definition and *EXECUTE authority to the media definition library.
- This command does not save the data dictionary for the library or its associated database files. To save them, the Save Library (SAVLIB) command should be used.
- When using the OUTFILE parameter to save to an existing database file, you must have *EXECUTE authority to the output file library.
- If tape is used, a standard-labeled volume must be placed in the device.
- No object being saved can be changed by a job that is running at the time the save operation occurs unless save-while-active is used.
- When the contents of a save file are being saved to the same save file by specifying SAVFDTA(*YES), only the description of the save file is saved.
- When the contents of a save file are saved by specifying SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
- When using the **Command user space (CMDUSRSPC)** parameter, you must have *USE authority to the user space and *EXECUTE authority to the user space library.

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
LIB	Library	Single values: *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
DEV	Device	Single values: *SAVF, *MEDDFN Other values (up to 4 repetitions): <i>Name</i>	Required, Positional 3
OBJTYPE	Object types	Single values: *ALL Other values (up to 300 repetitions): <i>Character value</i>	Optional, Positional 4
VOL	Volume identifier	Single values: *MOUNTED Other values (up to 75 repetitions): <i>Character value</i>	Optional, Positional 5
SEQNBR	Sequence number	1-16777215, *END	Optional
LABEL	Label	<i>Character value</i> , *LIB	Optional
EXPDATE	File expiration date	<i>Date</i> , *PERM	Optional
ENDOPT	End of media option	*REWIND, *LEAVE, *UNLOAD	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	
MEDDFN	Media definition	<i>Qualified object name</i>	Optional
	Qualifier 1: Media definition	<i>Name</i>	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
OPTFILE	Optical file	<i>Path name</i> , *	Optional
USEOPTBLK	Use optimum block	*YES, *NO	Optional
TGTRLS	Target release	*CURRENT, *PRV, V5R3M0, V5R4M0, V6R1M0	Optional
UPDHST	Update history	*YES, *NO	Optional
CLEAR	Clear	*NONE, *ALL, *AFTER, *REPLACE	Optional
PRECHK	Object pre-check	*NO, *YES	Optional
SAVACT	Save active	*NO, *LIB, *SYNCLIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, 120, *NOMAX	
	Element 2: Pending record changes	0-99999, *LOCKWAIT, *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, *LOCKWAIT, *NOMAX	
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	<i>Name</i> , *NONE, *WRKSTN	
	Qualifier 2: Library	<i>Name</i> , *LIBL, *CURLIB	
SYNCID	Synchronization ID	<i>Name</i> , *NONE	Optional
FILEMBR	File member	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: File	<i>Name</i> , *ALL	
	Element 2: Member	Single values: *ALL, *NONE Other values (up to 50 repetitions): <i>Generic name, name</i>	
ACCPH	Save access paths	*SYSVAL, *NO, *YES	Optional
SAVFDTA	Save file data	*YES, *NO	Optional

Keyword	Description	Choices	Notes
SPLFDTA	Spooled file data	* <u>NONE</u> , *ALL	Optional
QDTA	Queue data	* <u>NONE</u> , *DTAQ	Optional
PVTAUT	Private authorities	* <u>NO</u> , *YES	Optional
STG	Storage	* <u>KEEP</u> , *FREE	Optional
DTACPR	Data compression	* <u>DEV</u> , *NO, *YES, *LOW, *MEDIUM, *HIGH	Optional
COMPACT	Data compaction	* <u>DEV</u> , *NO	Optional
OMITLIB	Libraries to omit	Single values: * <u>NONE</u> , *USRSPC Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Single values: *USRSPC Other values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *<u>NONE</u>, *ALL</i>	
	Qualifier 2: Library	<i>Generic name, name, *<u>ALL</u></i>	
	Element 2: Object type	<i>Character value, *<u>ALL</u></i>	
ASPDEV	ASP device	<i>Name, *, *SYSBAS, *CURASPGRP</i>	Optional
OUTPUT	Output	* <u>NONE</u> , *PRINT, *OUTFILE	Optional
OUTFILE	File to receive output	<i>Qualified object name</i>	Optional
	Qualifier 1: File to receive output	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *<u>LIBL</u>, *CURLIB</i>	
OUTMBR	Output member options	<i>Element list</i>	Optional
	Element 1: Member to receive output	<i>Name, *<u>FIRST</u></i>	
	Element 2: Replace or add records	* <u>REPLACE</u> , *ADD	
INFTYPE	Type of output information	* <u>OBJ</u> , *LIB, *MBR, *ERR	Optional
CMDUSRSPC	Command user space	<i>Qualified object name</i>	Optional
	Qualifier 1: Command user space	<i>Name</i>	
	Qualifier 2: Library	<i>Name, *<u>LIBL</u>, *CURLIB</i>	

Top

Objects (OBJ)

Specifies the names of one or more objects or the generic name of each group of objects to be saved. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

***ALL** All the objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of objects in the specified library to be saved. A

generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name Specify one or more names of specific objects to be saved. Both generic names and specific names can be specified in the same command.

Top

Library (LIB)

Specifies which libraries contain the objects to be saved. If *ALL is specified for the **Objects (OBJ)** parameter, up to 300 library names can be specified.

This is a required parameter.

Single values

*USRSPC

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are saved. Only one library can be specified in the user space when saving to a save file. Special value *SPLF cannot be specified in the user space.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries that contain objects to be saved. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

Note: A generic library name cannot be specified when saving to a save file.

name Specify the name of the library that contains objects to be saved.

Note: Only one library can be specified when saving to a save file.

Top

Device (DEV)

Specifies the name of the device used for the save operation. The device name must already be known on the system by a device description.

This is a required parameter.

Single values

***SAVF** The save operation is done using the save file specified for the **Save file (SAVF)** parameter.

*MEDDFN

The save operation is done using the devices and media identified in the media definition specified for the **Media definition (MEDDFN)** parameter.

Other values

optical-device-name

Specify the name of the optical device used for the save operation.

tape-media-library-device-name

Specify the name of the tape media library device used for the save operation.

tape-device-name

Specify the names of one or more tape devices used for the save operation. If a virtual tape device is used, it must be the only device specified. If multiple tape devices are used, they must have compatible media formats and their names must be specified in the order in which they are used. Using more than one tape device permits one tape volume to be rewound and unloaded while another tape device processes the next tape volume. To use more than one device in parallel, a media definition must be specified.

Top

Object types (OBJTYPE)

Specifies the types of system objects to be saved.

Single values

***ALL** All object types that are specified by name and are in the specified library are saved. If *ALL is also specified for the **Objects (OBJ)** parameter, all the objects in the library that are of the types that can be saved are saved.

Other values (up to 300 repetitions)

object-type

Specify the value for each of the types of objects to be saved, such as command (*CMD), file (*FILE), or program (*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Volume identifier (VOL)

Specifies the volume identifiers of the volumes, or the cartridge identifiers of tapes in a tape media library device, on which the data is saved. The volumes must be placed in the device in the same order as specified for this parameter.

Single values

***MOUNTED**

The data is saved on the volumes placed in the device. 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.

Note: This value cannot be specified when using an optical media library device.

Other values (up to 75 repetitions)

character-value

Specify the identifiers of one or more volumes in the order in which they are placed in a device and used to save the data.

Top

Sequence number (SEQNBR)

Specifies, when tape is used, the sequence number to use as the starting point for the save operation.

***END** The save operation begins after the last sequence number on the first tape. If the first tape is full, an error message is issued and the operation ends.

1-16777215

Specify the sequence number of the file to be used for the save operation.

Top

Label (LABEL)

Specifies the name that identifies the data file on the tape volume that is to be used for the save operation. If this parameter is used on the save command, the same label must be specified on the restore command.

Note: You cannot specify *SAVLIB on this parameter, since it is a special value for the **Label (LABEL)** parameter of the restore command and would prevent you from restoring what you saved.

***LIB** The file label is created by the system using the name of the library specified for the **Library (LIB)** parameter.

character-value

Specify the data file identifier of the data file used for the save operation. A maximum of 17 characters can be used. This option is valid only for a single-library save operation.

Top

File expiration date (EXPDATE)

Specifies the expiration date of the file created by the save operation. If a date is specified, the file is protected and cannot be overwritten until the specified expiration date.

Notes:

1. This parameter is valid for tape and optical files.
2. Specifying this parameter does not protect against a later save operation specifying CLEAR(*ALL).

***PERM**

The file is protected permanently.

date Specify the date when protection for the file ends.

Top

End of media option (ENDOPT)

Specifies the operation that is automatically done on the tape or optical volume after the save operation ends. If more than one volume is used, this parameter applies only to the last volume used; all other volumes are unloaded when the end of the volume is reached.

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

***REWIND**

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

***LEAVE**

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

***UNLOAD**

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

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. The save file must be empty, unless *ALL is specified for the **Clear (CLEAR)** parameter.

Note: A value must be specified for this parameter if *SAVF is specified for the **Device (DEV)** parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Media definition (MEDDFN)

Specifies the media definition (*MEDDFN) object that identifies the devices and media used to contain the saved data. For information about creating and using a media definition, see the Recovering your system book, SC41-5304, and the Create Media Definition API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a media definition is specified, the VOL, SEQNBR, SAVF, and OPTFILE parameters cannot be specified. The volume identifiers and sequence numbers are specified in the media definition.

Qualifier 1: Media definition

name Specify the name of the media definition to be used.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library to be searched.

Optical file (OPTFILE)

Specifies the path name of the optical file that is used for the save operation, beginning with the root directory of the volume.

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

***** The system generates an optical file name in the root directory of the optical volume.

'optical-directory-path-name*'

The system generates an optical file name in the specified directory of the optical volume.

'optical-file-path-name'

Specify the path name of the optical file.

Use optimum block (USEOPTBLK)

Specifies whether or not the optimum block size is used for the save operation.

Note: Specifying USEOPTBLK(*YES) may result in a tape that can be duplicated only to a device that supports the same block size.

***YES** The optimum block size supported by the device is used for Save commands. If the block size that is used is larger than a block size that is supported by all device types, then:

- Performance may improve.
- The tape file that is created is only compatible with a device that supports the block size used. Commands such as Duplicate Tape (DUPTAP) do not duplicate files unless the files are being duplicated to a device which supports the same block size that was used.
- The value for the DTACPR parameter is ignored.

***NO** The optimum block size supported by the device is not used. Save commands use the default block size supported by all device types. The tape volume can be duplicated to any media format using the Duplicate Tape (DUPTAP) command.

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the object.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

To specify that an object be saved for distribution to a system at a different release level than the system on which the save operation is to occur, the procedure differs for program or non-program objects and by

the release level on which a program object is created. If, for example, you are saving an object for distribution to a target system running on an earlier release, you have the following choices:

For program objects:

- If the program object was created at a release level more current than the targeted earlier release, you must:
 1. create the program object again specifying the targeted earlier release
 2. save the program object specifying the targeted earlier release
 3. restore the program object on the target system.
- If the program object was created at the same release level as the target system, you can:
 1. save the program object specifying the targeted earlier release
 2. restore the program object on the target system.

For non-program objects:

You can:

1. save the object specifying the targeted earlier release
2. restore the object on the target system.

***CURRENT**

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

- *PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Update history (UPDHST)

Specifies whether the save history information of each saved object is changed with the date, time, and location of this save operation. The save history information for an object is displayed using the Display Object Description (DSPOBJD) command. The save history information is used to determine which journal entries are processed when RCVRNG(*LASTSAVE) and FROMENT(*LASTSAVE) or FROMENTLRG(*LASTSAVE) are used on the Apply Journalized Changes (APYJRNCHG) command.

- *YES** The last save date, time, and location is updated in each object saved.

- *NO** The save history information contained in the description of each object saved is not updated.

Note: UPDHST(*NO) should be used for a save operation that is not intended for recovery. For example, if the save data is sent, record by record, to another system and the save file immediately deleted, the save history information is probably not to be updated.

Top

Clear (CLEAR)

Specifies whether active data on the media is automatically cleared or replaced. Active data is any file on the media that has not expired. For saves to tape, clearing active data will make any files on the tape volume beyond the last file written by the save operation no longer accessible. For saves to optical, the files written by the save operation can be automatically replaced while other files on the volume remain active, or all active files can be automatically cleared. Clearing does not erase the data, it just makes the files no longer accessible.

Notes:

1. Clearing a tape does not initialize it. You should initialize tapes to a standard label format before the save command is issued by using the Initialize Tape (INZTAP) command and specifying a value for the NEWVOL parameter.
2. Clearing an optical volume does initialize it.
3. If a volume that is not initialized is encountered during the save operation, an inquiry message is sent and an operator can initialize the volume.

*NONE

None of the media is automatically cleared. If the save operation encounters active data on a tape volume or in a save file, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

***ALL** All of the media is automatically cleared.

If tapes are used and a sequence number is specified for the SEQNBR parameter, the first tape is cleared beginning at that sequence number. All tapes following that first tape are completely cleared. To clear the entire first tape, SEQNBR(1) must be specified.

*AFTER

All media after the first volume is automatically cleared. If the save operation encounters active data on the first tape volume, an inquiry message is sent, allowing the operator to either end the save operation or clear the media. If the save operation encounters the specified optical file on the first volume, an inquiry message is sent, allowing the operator to either end the save operation or replace the file.

Note: The *AFTER value is not valid for save files.

*REPLACE

Active data on the media is automatically replaced. Optical volumes are not initialized. Other media is automatically cleared in the same way as the *ALL value.

Top

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

***NO** The save operation for a library continues, saving only those objects that can be saved.

***YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation

continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB** Objects in a library can be saved while they are in use by another job. All of the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYNCLIB**

Objects in a library can be saved while they are in use by another job. All of the objects and all of the libraries in the save operation reach a checkpoint together and are saved in a consistent state in relationship to each other.

Note: If you specify this value and you are saving many libraries, it can take a long time to reach a checkpoint for all of the objects and libraries in the save operation.

***SYSDFN**

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object lock before continuing the save operation.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved

must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Top

Synchronization ID (SYNCID)

Specifies the name of the synchronized checkpoint in which this save while active operation will participate. The synchronized checkpoint must already be started by the Start Save Synchronization (STRSAVSYNC) command.

*NONE

The checkpoint for this save while active operation is not synchronized with any other save while active operation.

name Specify the name of the synchronized checkpoint. If you specify a name, you must also specify a value of *SYNCLIB for the **Save active (SAVACT)** parameter.

Note: If you specify a name, the value used for the **Save active wait time (SAVACTWAIT)** parameter **Element 2: Pending record changes** is the largest value specified among all of the participating save operations. However, if any participating save operation specifies *NOCMTBDY, then all participating save operations must specify *NOCMTBDY.

Top

File member (FILEMBR)

Specifies the database file members that are saved. This parameter is made up of two parts: the file name and the member name.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by its complete name, a generic name, or *ALL. The **Object types (OBJTYPE)** parameter value must be *ALL or include *FILE.

Note: This parameter cannot be specified when STG(*FREE) is specified.

Element 1: File

***ALL** The list of member name values that follows this value applies to all files specified for the OBJ parameter.

name Specify the name of the database file from which the listed members are to be saved. Up to 50 files can be specified, with a member list for each file.

Note: Generic names are not valid for the database file name, but are allowed for the member name.

Note: Duplicate file names are not allowed.

Element 2: Member

Single values

***ALL** All members are saved from the specified file.

***NONE**

No members are saved from the specified file. Only the file description is saved.

Other values (up to 50 repetitions)

generic-name

Specify the generic names of the members to be saved from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete member name.

Note: If generic member names are specified, the file must contain member names that match the generic names for the file to be saved. For example, if PAY* is specified as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved. If files specified by the FILEMBR parameter are not saved because members with the specified generic name cannot be found, a diagnostic message is sent, the save operation ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.

name Specify the names of the members to be saved from the given file.

Note: If specific member names are specified, the specified members must exist in the file for any part of the file to be saved or restored.

Top

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Top

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE

No spooled file data is saved.

***ALL** For each output queue that is saved, all available spooled file data on the output queue is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE

Only the description of a queue is saved.

*DTAQ

The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save private authorities with the objects that are saved. Saving private authorities will increase the amount of time it takes to save the objects, but it can simplify the recovery of an object or a group of objects. It will not simplify the recovery of an entire system.

*NO No private authorities are saved.

*YES Private authorities are saved for each object that is saved.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify this value.

Top

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP

The storage occupied by the data portion of the objects being saved is not freed.

*FREE The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

Note: If *DEV is specified for both this parameter and the **Data compaction (COMPACT)** parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed.

If *YES is specified for this parameter and *DEV is specified for the COMPACT parameter, both device data compaction and device data compression are performed if supported on the device.

***DEV** If the save is to tape and the target device supports compression, hardware compression is performed. Otherwise, no data compression is performed.

***NO** No data compression is performed.

***YES** If the save is to tape and the target device supports compression, hardware compression is performed. If compression is not supported, or if the save data is written to optical media or to a save file, software compression is performed. Low software compression is used for all devices except optical DVD, which uses medium software compression.

***LOW** If the save operation is to a save file or optical, software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

Note: This value is not valid for tape.

***MEDIUM**

If the save operation is to a save file or optical, software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

Note: This value is not valid for tape.

***HIGH**

If the save operation is to a save file or optical, software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Note: This value is not valid for tape.

Top

Data compaction (COMPACT)

Specifies whether device data compaction is performed.

***DEV** Device data compaction is performed if the data is saved to tape and all tape devices specified for the **Device (DEV)** parameter support the compaction feature.

Note: If *DEV is specified for both the **Data compression (DTACPR)** parameter and this parameter, only device data compaction is performed if device data compaction is supported on the device. Otherwise, data compression is performed if supported on the device.

If *YES is specified for the DTACPR parameter and *DEV is specified for this parameter, both device data compaction and device data compression are performed if supported on the device.

***NO** Device data compaction is not performed.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

***NONE**

No libraries are excluded from the save operation.

***USRSPC**

The libraries identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Single values

***USRSPC**

The objects identified in the user space specified for the **Command user space (CMDUSRSPC)** parameter are excluded from the save operation.

Other values (up to 300 repetitions)

Element 1: Object

Qualifier 1: Object

***NONE**

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

***ALL** The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

***ALL** All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS**

The system ASP and all basic user ASPs are included in the save operation.

***CURASGRP**

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Output (OUTPUT)

Specifies whether a list with information about the saved objects is created. The information can be printed with the job's spooled output or directed to a database file.

***NONE**

No output listing is created.

***PRINT**

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

***OUTFILE**

The output is directed to the database file specified for the **File to receive output (OUTFILE)** parameter.

Note: You must specify a database file name for the **File to receive output (OUTFILE)** parameter when OUTPUT(*OUTFILE) is specified.

File to receive output (OUTFILE)

Specifies the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library. If the file is created, the public authority for the file is the same as the create authority specified for the library in which the file is created. Use the Display Library Description (DSPLIBD) command to show the library's create authority.

Qualifier 1: File to receive output

name Specify the name of the database file to which the command output is directed.

Qualifier 2: Library

***LIBL** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file will be created in the QGPL library.

***CURLIB**

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

name Specify the name of the library to be searched.

Note: If a new file is created, the system uses the IBM-supplied file QASAVOBJ with format name QRSASV as a model.

Output member options (OUTMBR)

Specifies the name of the database file member to which the output is directed when *OUTFILE is specified for the **Output (OUTPUT)** parameter.

Element 1: Member to receive output

***FIRST**

The first member in the file receives the output. If OUTMBR(*FIRST) is specified and the member does not exist, the system creates a member with the name of the file specified for the **File to receive output (OUTFILE)** parameter.

name Specify the name of the file member that receives the output. If OUTMBR(member-name) is specified and the member does not exist, the system creates it.

If the member exists, you can add records to the end of the existing member or clear the existing member and add the records.

Element 2: Replace or add records

***REPLACE**

The existing records in the specified database file member are replaced by the new records.

***ADD** The new records are added to the existing information in the specified database file member.

Type of information (INFTYPE)

Specifies the type of information which is printed or directed to the database file.

- *OBJ** The list contains an entry for each object requested to be saved.
- *ERR** The list contains information about the command, an entry for each library, and an entry for each object that was not successfully saved.
- *LIB** The list contains a library entry for each library requested to be saved
- *MBR** The list contains an entry for each object, database file member, and spooled file requested to be saved.

Top

Command user space (CMDUSRSPC)

Specifies a user space containing the values for the parameters which have *USRSPC specified for this command. The user space allows up to 32767 list values for each parameter, while the command parameters only allow up to 300 list values. The user space must define the parameters in the format used by the Save Object List (QSRSAVO) API.

Qualifier 1: User space

name Specify the name of the user space containing the values for the parameters which have *USRSPC specified for this command.

Qualifier 2: Library

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

***CURLIB**

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

name Specify the name of the library where the user space is located.

Top

Examples

Example 1: Saving Program and File With Same Name

```
SAVOBJ OBJ(PETE) LIB(LIBX) DEV(TAP01)
```

This command saves the objects named PETE which are located in the LIBX library. If, for example, LIBX contains both a program and a file named PETE, both objects are saved. The storage occupied by the object is not freed because the STG parameter default (*KEEP) was assumed.

Example 2: Freeing System Storage

```
SAVOBJ OBJ(MSTRPAY PAY*) LIB(QGPL) DEV(TAP01) STG(*FREE)
```

The object named MSTRPAY, and all the objects whose names start with the characters PAY located in the general purpose library (QGPL), are saved. As part of the save operation, the system storage that was occupied by the data portion of the saved member, module, program, service program, SQL package, and journal receiver objects is freed.

Example 3: Saving File on Optical

```
SAVOBJ OBJ(FILEA) OBJTYPE(*FILE) LIB(LIBY) DEV(OPT01)
      VOL(TOM) CLEAR(*REPLACE)
```

The file named FILEA in the LIBY library is saved in a file with the library name LIBY on the optical volume that is identified by the volume identifier TOM. If the LIBY file already exists on the optical volume, it is automatically replaced when FILEA is saved.

Example 4: Saving Objects Supported on Previous Release

```
SAVOBJ OBJ(PAY*) LIB(LIB1) DEV(TAP01) TGTRLS(*PRV)
```

This command saves the objects beginning with the characters PAY from the LIB1 library in a format compatible with the previous release of the operating system. Only those objects supported on the previous release are saved.

Example 5: Saving Description and Data of File

```
SAVOBJ OBJ(SAVEFILE) LIB(MYLIB) OBJTYPE(*FILE) DEV(TAP01)
      SAVFDTA(*YES)
```

This command saves the file named SAVEFILE which is located in the library named MYLIB. Both the description and the data are saved for this save file.

Top

Error messages

*ESCAPE Messages

CPF3701

&1 objects saved from &3. &2 not saved.

CPF3702

&1 objects saved from &3. &2 not saved. &9 not included.

CPF3703

&2 &1 in &3 not saved.

CPF3708

Save file &1 in &2 too small.

CPF3709

Tape devices do not support same densities.

CPF3727

Duplicate device &1 specified on device name list.

CPF3728

Device &1 specified with other devices.

CPF3730
Not authorized to &2 &1 in library &3.

CPF3731
Cannot use &2 &1 in library &3.

CPF3733
&2 &1 in &3 previously damaged.

CPF3735
Storage limit exceeded for user profile &1.

CPF3738
Device &1 used for save or restore is damaged.

CPF3747
Object names cannot be specified with more than one library.

CPF3749
Objects from library &2 not saved.

CPF3767
Device &1 not found.

CPF3768
Device &1 not valid for command.

CPF377D
Save ended because of read error on internal system resource.

CPF377E
Not enough storage for save-while-active request.

CPF377F
Save-while-active request prevented by pending record changes.

CPF3770
No objects saved or restored for library &1.

CPF3771
&1 objects saved from &3. &2 not saved.

CPF3774
&1 objects saved from &3. &2 not saved. &8 not included.

CPF3778
Not all objects saved from all libraries.

CPF378A
Message queue not available.

CPF378C
SAVACTMSGQ(*WRKSTN) not valid for batch job.

CPF378E
Library &1 not saved.

CPF3781
Library &1 not found.

CPF3782
File &1 in &2 not a save file.

CPF3789
Only one library allowed with specified parameters.

CPF379E
Not enough storage available to save library &1.

CPF3793
Machine or ASP storage limit reached.

CPF3794
Save or restore operation ended unsuccessfully.

CPF3797
Objects from library &3 not saved. Save limit exceeded.

CPF37AB
*NOCMTBDY not allowed with target release.

CPF37AC
Library not allowed with *NOCMTBDY.

CPF37B1
SPLFDTA not allowed with target release.

CPF37B4
User space &1 in &2 not valid.

CPF37B5
PVTAUT not allowed with target release.

CPF37B7
Not authorized to save private authorities.

CPF37B9
Synchronization ID &1 in use.

CPF37BC
Synchronization ID &1 ended. Wait time exceeded.

CPF37BD
Synchronization ID &1 ended. Save ended before checkpoint.

CPF37BE
Synchronization ID &1 not started.

CPF37C7
Synchronization ID &1 ended. SAVACTWAIT error.

CPF380B
Save cannot be completed at this time.

CPF3812
Save file &1 in &2 in use.

CPF3815
Save file &1 in &2 too small for save operation.

CPF384E
USEOPTBLK(*YES) not valid for CD-ROM premastering.

CPF3867
Contents of FILEMBR parameter not correct.

CPF3868
FILEMBR specified but OBJTYPE must be *ALL or *FILE.

CPF3871
No objects saved or restored; &3 objects not included.

CPF388B

Optical file path name not valid.

CPF3892

&2 &1 in &3 not saved.

CPF3894

Cancel reply received for message &1.

CPF38A2

ASP device &1 not correct.

CPF38A3

File &1 in &2 not valid with ASPDEV.

CPF38A4

ASP device &1 not correct.

CPF5729

Not able to allocate object &1.

CPF9809

Library &1 cannot be accessed.

CPF9812

File &1 in library &2 not found.

CPF9814

Device &1 not found.

CPF9820

Not authorized to use library &1.

CPF9822

Not authorized to file &1 in library &2.

CPF9825

Not authorized to device &1.

CPF9833

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

CPFB8ED

Device description &1 not correct for operation.

OPT1332

Optical volume not found in device &1.

STATUS Messages*CPF3770**

No objects saved or restored for library &1.

CPF3771

&1 objects saved from &3. &2 not saved.

CPF3871

No objects saved or restored; &3 objects not included.

Top

Save Performance Collection (SAVPFCOL)

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

Parameters
Examples
Error messages

The Save Performance Collection (SAVPFCOL) command saves a copy of a single performance collection or a group of performance collections located in the same library.

Restrictions:

- This command is shipped with no public authority (*EXCLUDE).
- You must either have save system (*SAVSYS) special authority or the following object authorities:
 - object existence (*OBJEXIST) authority for each object to be saved.
 - execute (*EXECUTE) authority to each library objects are saved from.
- If you do not have *SAVSYS special authority, only those objects that you have authority for are saved.
- You must have *EXECUTE authority to the library where the save file is located.

Top

Parameters

Keyword	Description	Choices	Notes
COL	Collection	Single values: *SELECT, *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional, Positional 1
LIB	Library	<i>Name</i> , <u>QPFRDATA</u>	Optional
COLTYPE	Collection type	Single values: *ALL Other values (up to 10 repetitions): <i>Character value</i> , <u>*CSFILE</u>	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	
TGTRLS	Target release	<i>Character value</i> , *CURRENT, *PRV	Optional
DTACPR	Data compression	*NO, *YES, *LOW, *MEDIUM, *HIGH	Optional

Top

Collection (COL)

Specifies the names of one or more performance collections or the generic name of each group of performance collections to be saved. All the performance collections must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Collection type (COLTYPE)** parameter, all the collection types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

Single values

*SELECT

Lists all performance collections available in the specified library so you can select them to save.

Note: This value is valid only in an interactive environment.

***ALL** All the performance collections in the specified library are saved, depending on the values specified for the **Collection type (COLTYPE)** parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of performance collections in the specified library to be saved. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk is not specified with the name, the system assumes that the name is a complete performance collection name.

name Specify one or more names of specific performance collections to be saved. Both generic names and specific names can be specified in the same command.

Top

Library (LIB)

Specifies which library contains the performance collections to be saved.

QPFRRDATA

The performance collections to be saved are located in the IBM-supplied performance library, QPFRRDATA.

name Specify the name of the library where the performance collections are located.

Top

Collection type (COLTYPE)

Specifies the type of collection to be saved.

Single values

***ALL** All types of performance collections that are specified by name and are in the specified library are saved. This includes file-based collections and object-based collections. If ***ALL** is also specified for the **Collection (COL)** parameter, all the performance collections in the library that are of the types that can be saved are saved.

Other values (up to 10 repetitions)

*CSFILE

Only the Collection Services file-based collections specified in the **Collection (COL)** parameter are to be saved.

*CSMGTCOL

Only the Collection Services object-based collections specified in the **Collection (COL)** parameter are to be saved.

*DWFIL

Only the Disk Watcher file-based collections specified in the **Collection (COL)** parameter are to be saved.

*JWFIL

Only the Job Watcher file-based collections specified in the **Collection (COL)** parameter are to be saved.

*PEXFILE

Only the Performance Explorer file-based collections specified in the **Collection (COL)** parameter are to be saved.

***PEXMGTCOL**

Only the Performance Explorer object-based collections specified in the **Collection (COL)** parameter are to be saved.

type Specify the type of collections to be saved.

Valid values depend on the performance collections supported by the system. You can press F4 while prompting this command parameter to see a list of valid collection type values.

Top

Save file (SAVF)

Specifies the save file that is used to contain the saved data. This is a required parameter.

Qualifier 1: Save file

name Specify the name of save file to be used.

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

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

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to restore and use the performance collections.

When specifying the target-release value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**

The performance collections are to be restored to, and used on, the release of the operating system currently running on your system. The performance collections can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The performance collections are to be restored to the previous release with modification level 0 of the operating system. The performance collections can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Data compression (DTACPR)

Specifies whether data compression is used. If the save is running while other jobs on the system are active and software compression is used, the overall system performance may be affected.

***NO** No data compression is performed.

***YES** Software compression is performed. Low software compression is used.

***LOW** Software data compression is performed with the SNA algorithm. Low compression is usually faster and the compressed data is usually larger than if medium or high compression is used.

***MEDIUM**

Software data compression is performed with the TERSE algorithm. Medium compression is usually slower than low compression but faster than high compression. The compressed data is usually smaller than if low compression is used and larger than if high compression is used.

***HIGH**

Software data compression is performed with the LZ1 algorithm. High compression is usually slower and the compressed data is usually smaller than if low or medium compression is used.

Top

Examples

Example 1: Saving Performance Collections

```
SAVPFRCOL COL(TEST1 TEST2)
```

This command save performance collections TEST1 and TEST2 from the IBM-supplied performance library QPFRDATA. This command runs interactively while the user waits.

Top

Error messages

*ESCAPE Messages

CPFC763

Performance collection &2 of type &3 in library &1 does not exist.

CPF3730

Not authorized to &2 &1 in library &3.

CPF3731

Cannot use &2 &1 in library &3.

CPF3733

&2 &1 in &3 previously damaged.

CPF3735

Storage limit exceeded for user profile &1.

CPF3781

Library &1 not found.

CPF3812

Save file &1 in &2 in use.

CPF9820

Not authorized to use library &1.

Top

Save Restore (SAVRST)

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

Parameters
Examples
Error messages

The Save/Restore IFS (SAVRST) command saves and restores a copy of one or more objects, that can be used in the integrated file system (IFS).

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

The SAVRST command uses the current save and restore support for objects in libraries and for document library objects. As a result, there are restrictions when you use the SAVRST command on these objects.

Restrictions:

1. For name patterns in the root directory:
 - a. OBJ must be one of the following:
 - OBJ('//*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
 - OBJ('//*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT) ('/other values' *OMIT)
 2. For names involving objects in libraries:
 - a. OBJ must have only one name, except when omitting /QSYS.LIB from /*
 - b. OBJ must be one of the following:
 - OBJ('/QSYS.LIB/libname.LIB')
 - OBJ('/QSYS.LIB/libname.LIB/*')
 - OBJ('/QSYS.LIB/libname.LIB/*.type')
 - OBJ('/QSYS.LIB/libname.LIB/objname.type')
 - OBJ('/QSYS.LIB/libname.LIB/filename.FILE/*')
 - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/*.MBR')
 - OBJ('/QSYS.LIB/libname.LIB/filename. FILE/membername.MBR')
 - OBJ('/QSYS.LIB/*.type')
 - OBJ('/QSYS.LIB/objname.type')
 - OBJ('/QSYS.LIB/filename.FILE/*')
 - OBJ('/QSYS.LIB/filename.FILE/*.MBR')
 - OBJ('/QSYS.LIB/filename. FILE/membername.MBR')
 - c. The .type must be an object type supported by SAVOBJ and RSTOBJ
 - d. libname cannot be QSYS, QDOC, QDOCxxxx, QTEMP, QSPL, QSPLxxxx, QSRV, QRECOVERY, QRPLOBJ, or QSR if libname.LIB is the last component of the name
 - e. SUBTREE must be *ALL
 - f. For SAVRST:
 - CHGPERIOD end date and end time must be *ALL
 - CHGPERIOD must be default if a file member is specified
 - An object cannot be renamed
 - For database file members, OPTION(*NEW) only restores members for new files
3. For names involving document library objects:

- a. OBJ must have only one name, except when omitting /QDLS from /*
 - b. OBJ and SUBTREE must be one of the following:
 - OBJ('/QDLS/path/foldername') SUBTREE(*ALL)
 - OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)
 - c. For SAVRST:
 - The defaults must be taken on the PRECHK and SAVACTMSGQ parameters
 - CHGPERIOD must be default with OBJ('/QDLS/path/documentname') SUBTREE(*OBJ)
 - CHGPERIOD start date cannot be *LASTSAVE
 - CHGPERIOD end date and end time must be *ALL
 - SAVACT cannot be *SYNC
 - SAVACTMSGQ must be *NONE
 - ALWOBJDIF must be *NONE or *ALL
 - OPTION must be *ALL
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 1
OBJ	Objects	Values (up to 50 repetitions): <i>Element list</i>	Optional, Positional 2
	Element 1: Name	<i>Path name, _</i>	
	Element 2: Include or omit	*INCLUDE , *OMIT	
PATTERN	Name pattern	Values (up to 50 repetitions): <i>Element list</i>	Optional
	Element 1: Pattern	<i>Character value, _</i>	
	Element 2: Include or omit	*INCLUDE , *OMIT	
SUBTREE	Directory subtree	*ALL , *DIR, *OBJ, *NONE	Optional
RBMDFS	Rebuild mounted file system	*NONE , *UDFS	Optional
CHGPERIOD	Time period for last change	<i>Element list</i>	Optional
	Element 1: Start date	<i>Date, *ALL</i>	
	Element 2: Start time	<i>Time, *ALL</i>	
	Element 3: End date	<i>Date, *ALL</i>	
	Element 4: End time	<i>Time, *ALL</i>	
TGTRLS	Target release	<i>Simple name, *CURRENT, *PRV</i>	Optional
PRECHK	Object pre-check	*NO , *YES	Optional
SAVACT	Save active	*NO , *YES, *SYNC	Optional
SAVACTMSGQ	Save active message queue	<i>Path name, *NONE, *WRKSTN</i>	Optional
ASPDEV	ASP device	<i>Name, *ALLAVL, *, *SYSBAS, *CURASPGRP</i>	Optional
OPTION	Option	*ALL , *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE , *ALL Other values (up to 2 repetitions): *OWNER, *AUTL, *PGP	Optional

Keyword	Description	Choices	Notes
FRCOBJCVN	Force object conversion	Single values: <u>*SYSVAL</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	
SCAN	Scan objects	<i>Element list</i>	Optional
	Element 1: Scan during save	<u>*NO</u> , *YES	
	Element 2: Save failed objects	<u>*NOSAVFAILED</u> , *SAVFAILED	
PVTAUT	Private authorities	<u>*NO</u> , *YES	Optional
CRTPRNDIR	Create parent directories	<u>*NO</u> , *YES	Optional
PRNDIROWN	Parent directory owner	<i>Simple name</i> , <u>*PARENT</u>	Optional

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccc or nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Objects (OBJ)

Specifies the objects to be saved. You can specify an object name pattern for the path name to be used. When a path name is specified that could match many objects, you can specify a value for the **Name pattern (PATTERN)** parameter to subset the objects that are to be saved.

A maximum of 300 path names can be specified.

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

Additional information about object name patterns is in the Integrated file system topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

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

Element 1: Name

'* The objects in the current directory are saved.

path-name

Specify an object path name or a pattern that can match many names.

Element 2: Include or omit

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 determines whether the subtrees are included or omitted.

*INCLUDE

The objects that match the object name pattern are to be saved, unless overridden by an *OMIT specification.

*OMIT

The objects that match the object name pattern are not saved. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Note: The objects will be restored with the same name.

Top

Name pattern (PATTERN)

Specifies one or more object name patterns to be used to subset the objects to be saved. The **Objects (OBJ)** parameter defines the set of candidate objects. A maximum of 50 values can be specified for this parameter.

Element 1: Pattern

* All objects which qualify for the operation are included or omitted.

character-value

Specify an object name or a pattern that can match many names.

Element 2: Include or omit

Specifies whether names that match the pattern should be included or omitted from the operation.

Note: The SUBTREE parameter determines whether the subtrees are included or omitted.

*INCLUDE

Only objects which are included by the OBJ parameter and match the PATTERN parameter are included in the save, unless overridden by an *OMIT specification.

*OMIT

All objects which are included by the OBJ parameter are included in the save except those objects which match the PATTERN parameter. This overrides an *INCLUDE specification and is intended to be used to omit a subset of a previously selected pattern.

Top

Directory subtree (SUBTREE)

Specifies whether directory subtrees are included in the save operation.

- *ALL** The entire subtree of each directory that matches the object name pattern is included. The subtree includes all subdirectories and the objects within those subdirectories.
- *DIR** The objects in the first level of each directory that matches the object name pattern are included. The subdirectories of each matching directory are included, but the objects in the subdirectories are not included.
- *NONE**
No subtrees are included in the save operation. If a directory matches the object name pattern specified, the objects in the directory are included. If the directory has subdirectories, neither the subdirectories nor the objects in the subdirectories are included.
- *OBJ** Only the objects that exactly match the object name pattern will be processed. If the object name pattern specifies a directory, objects in the directory are not included.
- *STG** The objects that match the object name pattern are processed along with the storage for related objects. Objects that are saved using this value can only be restored using SUBTREE(*STG).

Top

Rebuild mounted file system (RBDMFS)

Specifies which mounted file systems should be rebuilt during the restore.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority to specify a value other than *NONE for this parameter.

***NONE**

Mounted file systems will not be rebuilt during the restore. Objects that were saved from a mounted file system will be restored to the file system that contains the directory being restored into.

***UDFS**

Mounted user-defined file systems will be rebuilt during the restore. A user-defined file system will be created if it does not exist and then will be mounted over the same directory it was mounted over during the save.

If a user-defined file system is created during the restore, the attributes will be set based on the saved user-defined file system including any user-defined file system specific attributes such as 'Case sensitivity' or 'Default file format'. If the user-defined file system exists before the restore, no attributes will be changed.

If there is an error creating or mounting a user-defined file system, none of the objects that were saved from the mounted user-defined file system will be restored.

Note: If it does not exist before the restore, the directory that is being mounted over will be created with attributes and authorities copied from the directory being restored into. This could cause problems when the user-defined file system is unmounted and then remounted, if the authorities are not sufficient to allow the mount to occur.

Top

Time period for last change (CHGPERIOD)

Specifies a date/time range. Objects that were last changed within that range will be saved.

Element 1: Start date

***ALL** No starting date is specified. All objects last changed prior to the ending date will be saved.

***LASTSAVE**

The objects that have changed since the last time they were saved with UPDHST(*YES) specified are saved. **Notes:**

1. If this value is specified, the value *ALL must be specified for all other elements of this parameter.
2. For local file systems, the system archive attribute is used. For remote file systems, the PC archive attribute is used.

date Specify the date after which objects that have changed are to be saved. The date must be specified in job date format.

Element 2: Start time

***ALL** All times of day are included in the range.

time Specify the time on the start date after which objects that have changed are to be saved.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the starting date is an explicit date.

Element 3: End date

***ALL** No ending date is specified. All objects changed since the starting date will be saved.

date Specify the date before which objects that have changed are to be saved. The date must be specified in the job date format.

Element 4: End time

***ALL** All times of day are included in the range.

time Specify a time on the end date before which objects that have changed are to be saved.

The time is specified in 24-hour format with or without a time separator as follows:

- With a time separator, specify a string of 5 or 8 digits, where the time separator for the job separates the hours, minutes, and seconds. If you issue this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command fails.
- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where **hh** = hours, **mm** = minutes, and **ss** = seconds. Valid values for **hh** range from 00 through 23. Valid values for **mm** and **ss** range from 00 through 59.

Note: Specifying an explicit time is valid only if the ending date is an explicit date.

Top

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Object pre-check (PRECHK)

Specifies whether the save operation ends if any of the selected objects cannot be saved.

***NO** The save operation does not end. Objects that can be saved are saved.

***YES** The save operation ends. Nothing is saved unless all of the selected objects can be saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state, this parameter is ignored and the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***YES** Objects can be saved and used at the same time. The object checkpoints can occur at different times.

***SYNC**

Objects can be saved and used at the same time. All of the object checkpoints occur at the same time.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing is complete.

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

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

***NONE**

No notification message is sent.

***WRKSTN**

The notification message is sent to the work station message queue.

path-name

Specify the path name of the message queue to be used.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation.

***DFT** The operation uses the ASPDEV value appropriate for the file system from which objects are being saved. For Integrated File System objects, *ALLAVL is used. For objects from the QSYS file system, the corresponding save command ASPDEV default is used.

***ALLAVL**

The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and all available independent ASPs.

***** The operation includes the system ASP, all basic user ASPs, and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS**

The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP**

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Top

Option (OPTION)

Specifies whether to restore objects that already exist on the system or objects that do not already exist on the system.

***ALL** All of the specified objects are restored, whether they already exist on the system or not.

***NEW** Objects are restored only if they do not already exist on the system.

***OLD** Objects are restored only if they already exist on the system.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.
Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- *ALL** All of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

Other values (up to 3 repetitions)

***AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

- *PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Scan objects (SCAN)

Specifies whether objects will be scanned while being saved when exit programs are registered with any of the integrated file system scan-related exit points and whether objects that previously failed a scan should be saved.

The integrated file system scan-related exit points are:

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

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

Element 1: Scan during save

***NO** Objects will not be scanned by the scan-related exit programs.

***YES** Objects will be scanned according to the rules described in the scan-related exit programs.

Element 2: Save failed objects

***NOSAVFAILED**

Objects that have either previously failed a scan or that fail a scan by a QIBM_QP0L_SCAN_OPEN exit program during this save will not be saved.

***SAVFAILED**

Objects that have either previously failed a scan or that fail a scan during this save will be saved.

Top

Private authorities (PVTAUT)

Specifies whether to save and restore private authorities with the objects that are saved and restored.

***NO** No private authorities are saved or restored.

***YES** Private authorities are saved and restored with the objects.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority on the system from which objects are being saved, and *ALLOBJ special authority on the restore system, to specify this value.

Top

Create parent directories (CRTPRNDIR)

Specifies whether parent directories of objects being restored should be created if they do not exist. For example, if object '/a/b/c/file1' is being restored then directories '/a', '/a/b' and '/a/b/c' must exist. This parameter only applies to "root" (/), QOpenSys and user-defined file systems, and will be ignored for all other file systems.

***NO** Parent directories will not be created if they do not exist. Diagnostic message CPD375B will be sent and the object will not be restored.

***YES** The restore will create parent directories if they do not exist. The directories created by the restore will have *EXCLUDE public authority and will be owned by the user profile specified for the **Parent directory owner (PRNDIROWN)** parameter. The other parent directory attributes will be set using the shipped default values for the **Create Directory (CRTDIR)** command parameters.

Parent directory owner (PRNDIROWN)

Specifies the name of an existing user profile that will own parent directories created by the restore. This parameter only applies to "root" (/), QOpenSys and user-defined file systems, and will be ignored for all other file systems.

Note: If a value is specified for this parameter, *YES must be specified for the **Create parent directories (CRTPRNDIR)** parameter.

*PARENT

The owner of a parent directory being created by the restore will be the same as the owner of the directory it is being created into. For example, if object '/a/b/c/file1' is being restored and directory '/a' exists but the '/b' and '/b/c' directories do not exist, the '/b' and '/b/c' directories are created with the same owner as the '/a' directory.

name Specify the name of a user profile to be the owner of any parent directories that are created by the restore.

Top

Examples

Example 1: Saving and Restoring a Member Object

```
SAVRST  RMTLOCNAME(SYSTEM1)
        OBJ(('QSYS.LIB/JTEMP.LIB/ZXC.FILE/QYYCPDGT.MBR'))
```

This command saves the QYYCPDGT member from file ZXC in library JTEMP and restores the object on the iSeries system at remote location SYSTEM1.

Example 2: Saving and Restoring a Directory

```
SAVRST  RMTLOCNAME(SYSTEM2) OBJ(('MYDIR')) SAVACT(*YES)
        SAVACTMSGQ('QSYS.LIB/SVRTEST.LIB/ZXC.MSGQ')
```

This command saves the MYDIR directory while active, and will use the ZXC message queue in library SVRTEST to save messages.

Top

Error messages

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Configuration (SAVRSTCFG)

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

Parameters
Examples
Error messages

The Save/Restore Configuration (SAVRSTCFG) command saves and restores configuration information without requiring a dedicated system. The information saved and restored includes the following:

- Configuration lists (*CFGL)
- Connection lists (*C>NNL)
- Class-of-service descriptions (*COSD)
- Internet Packet Exchange descriptions (*IPXD)
- Mode descriptions (*MODD)
- NetBIOS descriptions (*NTBD)

The user profile of the system default owner (QDFTOWN) becomes the default owner of any objects being restored in the system when the profile of the owner is not known to the system.

If an object already exists on the restore system, the public and private authorities of the existing object are kept. If the object does not exist in the library, all public authorities are restored, but any private authorities must be granted again.

Restrictions:

1. To use this command, the user must have *SAVSYS authority, or object existence authority for (or be the owner of) each object specified if the object already exists on the system.
2. This command is shipped with public *EXCLUDE authority.
3. The user must have either *ALLOBJ authority or authority to the command.
4. Changes made to the configuration while the SAVRSTCFG command is being run may not be reflected on the restore system, depending on when the changes occurred in relation to the send operation. If the operation cannot obtain a lock on a configuration object, a diagnostic message is issued; and the object is not saved and restored. The operation does not obtain the lock on a configuration object when a change or delete operation against the object does not complete before the default wait timer expires.
5. If any system resource manager objects are in use, a message is issued and none of them are saved and restored. The commands which can cause this to happen are Work with Hardware Products (WRKHDWPRD) and Work with Hardware Resources (WRKHDWRSC). No display commands can cause the system resource manager database to lock. All other known functions do read-only operations.
6. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL, *SRM Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1

Keyword	Description	Choices	Notes
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 2
OBJTYPE	Object types	Single values: *ALL Other values (up to 5 repetitions): *CFGL, *CNL, *COSD, *MODD, *NTBD	Optional
SRM	System Resource Management	*NONE , *TRA	Optional
PRECHK	Object pre-check	*NO , *YES	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE , *ALL Other values (up to 3 repetitions): *AUTL, *OWNER, *PGP	Optional

Top

Objects (OBJ)

Specifies the objects to be restored. Specify the name of each object, or the generic name of each group of objects to restore. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an asterisk (*) is not specified with the name, the system assumes that the name is a complete object name.

If the **Object types (OBJTYPE)** parameter has a value of *ALL, all the object types listed in the description of the OBJTYPE parameter are restored, if they have the specified names.

System resource management (SRM) objects cannot be restored individually or by specifying a generic name. To restore only SRM objects, specify *SRM for this parameter and a value for the **System Resource Management (SRM)** parameter.

This is a required parameter.

Single values

***ALL** All the device configuration objects are restored, depending on the values specified for the OBJTYPE parameter.

***SRM** The device configuration objects are not restored, but system resource management (SRM) objects are restored based on the SRM parameter value.

ATTENTION You must specify SRM(*NONE) on the RSTCFG, unless the system you are restoring to is the exact same hardware configuration that the original configuration was saved on, to prevent the restore of the SRM information. If the SRM information is restored, the configuration objects may become unusable.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of objects in the specified library to restore.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify one or more names of specific objects to restore. Both generic names and specific names can be specified in the same command. A maximum of 300 object names can be specified.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccc or nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Object types (OBJTYPE)

Specifies the types of i5/OS system objects that are saved and restored.

***ALL** All configuration object types that are specified by name are saved and restored. If OBJ(*ALL) is specified, all of the saved configuration objects are saved and restored. The following types can be specified:

***CFGL**

All configuration list object types are saved and restored.

***CNL**

All connection list object types are saved and restored.

***COSD**

All class-of-service description object types are saved and restored.

***IPXD** All Internetwork Packet Exchange object types are saved and restored.

***MODD**

All mode description object types are saved and restored.

***NTBD**

All NetBIOS description object types are saved and restored.

Top

System Resource Management (SRM)

Specifies the type of system resource management (SRM) information to be saved and restored. This parameter is valid only when *ALL or *SRM is specified on the OBJ parameter.

***NONE**

No SRM information is restored.

***TRA** All token-ring adapter information is restored.

Top

Object pre-check (PRECHK)

Specifies whether the save configuration operation ends if any of the objects satisfy the following conditions:

- The objects were previously found to be damaged.
- The objects are locked by another job.
- The user does not have authority to save the objects.

***NO** The save operation continues, saving only configuration and system resource management (SRM) objects that can be saved.

***YES** The save operation ends before any data is written to the media if any configuration objects or system resource manager objects cannot be saved.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- Authorization list: The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.

Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

***NONE**

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

***ALL** All of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

Other values (up to 3 repetitions)

***AUTL**

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Examples

Example 1: Saving and Restoring Generic Objects

```
SAVRSTCFG OBJ(ABCD*) RMTLOCNAME(SYSTEM1) OBJTYPE(*CFGL)
          ALWOBJDIF(*NONE)
```

This command saves and restores all objects whose names begin with ABCD and whose object type is configuration list (*CFGL). Only those objects that have no differences on SYSTEM1 will be restored.

Example 2: Saving and Restoring All Objects

```
SAVRSTCFG OBJ(*ALL) RMTLOCNAME(SYSTEM1)
```

This command saves and restores all configuration objects whose types match the five object types listed on the OBJTYPE parameter.

Top

Error messages

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Changed Objects (SAVRSTCHG)

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

Parameters
Examples
Error messages

The Save/Restore Changed Object (SAVRSTCHG) command saves and concurrently restores a copy of each changed object, or group of objects located in the same library, to another system. For database files, only the changed members are saved and restored. Objects or members changed since the specified date and time are saved and restored.

Objects changed since the specified date and time are saved and restored with the following exceptions:

- If OBJJRN(*NO) is specified, objects currently being journaled are not saved and restored, unless journaling was started after the specified date and time. This ensures that changes made to objects before journaling starts are not lost (because they were not journaled in a journal receiver).
- Freed objects (programs, files, journal receivers, and so forth) are not saved.
- User-defined messages, job and output queue definitions, logical file definitions, and data queue descriptions are saved and restored, but the contents of those objects are not saved and restored. Logical file access paths are saved and restored if ACCPTH(*YES) is specified.

Specified objects that were changed, and the libraries where they reside, remain locked during the save and restore operation.

To determine the date and time that an object was changed, run the Display Object Description (DSPOBJD) command with DETAIL(*FULL) specified. For database file members that were changed, run the Display File Description (DSPFD) command.

The types of objects that can be saved and restored by this command are listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>. The system saves and concurrently restores the changed objects.

Restrictions:

1. To use this command, the user must have the special authority *SAVSYS specified in the user profile by the SPCAUT parameter. Otherwise, the user must have object existence authority for each object specified, and execute authority to the specified library. If the user does not have the necessary authority to a specified object, all changed objects except that object are saved and restored.
2. No changed object that is being saved and restored can be changed by a job that is running when the save and restore operation occurs unless save-while-active is used.
3. When the contents of a save file are saved with SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
LIB	Saved library	Single values: *ALLUSR Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 3
OBJTYPE	Object types	Single values: <u>*ALL</u> Other values (up to 72 repetitions): *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRTIMG, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSPC, *VLDL, *WSCST	Optional
OBJJRN	Journalled objects	<u>*NO</u> , *YES	Optional
REFDATE	Reference date	<i>Date</i> , <u>*SAVLIB</u>	Optional
REFTIME	Reference time	<i>Time</i> , <u>*NONE</u>	Optional
STRLIB	Starting library	<i>Name</i> , <u>*FIRST</u>	Optional
OMITLIB	Libraries to omit	Single values: <u>*NONE</u> Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name</i> , <u>*ALL</u> , <u>*NONE</u>	
	Qualifier 2: Library	<i>Generic name, name</i> , <u>*ALL</u>	
	Element 2: Object type	<i>Character value</i> , <u>*ALL</u> , *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMORY, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TIMZON, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST	
ASPDEV	ASP device	<i>Name</i> , <u>*</u> , *SYSBAS, *CURASPGRP	Optional
TGTRLS	Target release	<i>Simple name</i> , <u>*CURRENT</u> , *PRV	Optional
PRECHK	Object pre-check	<u>*NO</u> , *YES	Optional
SAVACT	Save active	<u>*NO</u> , *LIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, <u>120</u> , *NOMAX	
	Element 2: Pending record changes	0-99999, <u>*LOCKWAIT</u> , *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, <u>*LOCKWAIT</u> , *NOMAX	

Keyword	Description	Choices	Notes
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	<i>Name, *NONE, *WRKSTN</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACCPATH	Save access paths	*SYSVAL , *NO, *YES	Optional
SAVFDTA	Save file data	*YES , *NO	Optional
QDTA	Queue data	*NONE , *DTAQ	Optional
PVTAUT	Private authorities	*NO , *YES	Optional
RSTLIB	Restore to library	<i>Name, *LIB</i>	Optional
RSTASPDEV	Restore to ASP device	<i>Name, *SAVASPDEV</i>	Optional
RSTASP	Restore to ASP number	1-32, *SAVASP	Optional
OPTION	Option	*ALL , *NEW, *OLD	Optional
MBROPT	Data base member option	*MATCH , *ALL, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE , *ALL Other values (up to 3 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
FRCOBJCVN	Force object conversion	Single values: *SYSVAL , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	*YES	
	Element 2: Objects to convert	*RQD , *ALL	

Top

Objects (OBJ)

Specifies the names of one or more objects, or the generic names of each group of objects, to check for changes and then to save those objects that have changed. All the objects must be in the library specified for the **Library (LIB)** parameter. If ***ALL** is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

***ALL** All changed objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of changed objects to save in the specified library. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name Specify the names of specific objects to save. Both generic names and specific names can be specified in the same command.

Top

Saved library (LIB)

Specifies the library that contains the changed objects to be saved and restored.

Single values

*ALLUSR

All user libraries are saved and restored. All libraries with names that do not begin with the letter Q are saved and restored 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 saved and restored:

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGPL	QSRVAGT	QUSRIS	QUSRVxRxMx
QGPL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

Note: 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.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified.

name Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format ccccccc or nnnnnnnn.ccccccc, where nnnnnnnn is the network identifier (ID) and ccccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Object types (OBJTYPE)

Specifies the types of system objects whose changes are saved. The object types saved are also the ones saved and restored by the Save Library (SAVLIB), Restore Object (RSTOBJ), and Restore Library (RSTLIB) commands. Data dictionaries and the associated files are saved only by using the SAVLIB command.

Single values

***ALL** Changes to all object types that are specified by name, and which are in the specified library, are saved.

Other values (up to 300 repetitions)

object-type

Specify the value for each of the types of objects that are saved, such as command (*CMD), file (*FILE), or program (*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Journalized objects (OBJJRN)

Specifies whether to save changed objects that are currently being journaled and that have been journaled since the date and time specified for the **Reference date (REFDATE)** and **Reference time (REFTIME)** parameters.

***NO** Objects being journaled are not saved. If journaling was started after the specified date and time, the changed objects or changed database file members are saved. The date and time of the last journal start operation can be shown by using the Display Object Description (DSPOBJD) command.

***YES** Objects whose changes are entered in a journal are saved.

Top

Reference date (REFDATE)

Specifies the reference date. Objects that have been changed since this date are saved.

***SAVLIB**

The objects that have been changed since the date of the last running of the Save Library (SAVLIB) command are saved. If the specified library was never saved, a message is issued and the library is not saved, but the operation continues.

date Specify the reference date; objects that have been changed since this date are saved. If you specify a date later than the date of the running of this command, a message is issued and the operation ends. The date must be specified in the job date format.

Top

Reference time (REFTIME)

Specifies the reference time. Objects that have been changed since this time on the specified date are saved.

*NONE

No explicit time is specified. Any objects changed since the date specified for the **Reference date (REFDATE)** parameter are saved.

time Specify the reference time; objects that have been changed since this time on the specified date are saved. If *SAVLIB is specified for the REFDATE parameter, no reference time can be specified. If you specify a time later than the time of the running of this command, a message is issued and the operation ends.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 6 digits (hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Starting library (STRLIB)

Specifies the library with which to begin the *ALLUSR save. This parameter is used to recover from ended or failed *ALLUSR save operations.

*FIRST

The save operation begins with the first library in alphabetical order.

name Specify the name of the library with which to begin the save.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE

No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes

for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

*ALL All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

***YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB** Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYSDFN**

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object lock before continuing the save operation.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

*LOCKWAIT

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

*NOCMTBDY

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Top

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

***NONE**
Only the description of a queue is saved.

***DTAQ**
The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save and restore private authorities with the objects that are saved and restored.

***NO** No private authorities are saved or restored.

***YES** Private authorities are saved and restored with the objects.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority on the system from which objects are being saved, and *ALLOBJ special authority on the restore system, to specify this value.

Top

Restore to library (RSTLIB)

Specifies the name of the library to which the objects are restored.

***LIB** The restore library is the same name as the value specified on the LIB parameter.

name Specify the name of the library to which the objects are restored.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

*SAVASPDEV

The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Top

Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Recovering your system book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

*SAVASP

The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

Option (OPTION)

Specifies how to handle restoring each object.

- *ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.
- *NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.
- *OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- *FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

*ALL All members in the saved file are restored.

*NEW Only new members (members not known to the system) are restored.

*OLD Only members already known to the system are restored.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- Authorization list: The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.

Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

*ALL All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of

the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

*FILELVL

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

*OWNER

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.

3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Examples

Example 1: Saving and Restoring Multiple Changed Objects

```
SAVRSTCHG OBJ(*ALL) RMTLOCNAME(SYSTEM1) LIB(*ALLUSR)
           OBJTYPE(*PGM) STRLIB(MORE) ALWOBJDIF(*NONE)
```

This command saves all changed objects whose type is *PGM and are located in user libraries beginning with the library named MORE. The objects are restored on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Changed Object

```
SAVRSTCHG OBJ(PETE) RMTLOCNAME(SYSTEM1) LIB(ROD)
           ALWOBJDIF(*ALL) OPTION(*NEW)
```

This command saves changed objects named PETE of all types supported, which is located in the library named ROD, and restores the object on the remote system named SYSTEM1 if the object does not already exist on the remote system.

Error messages

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Save Restore Doc/Lib Object (SAVRSTDLO)

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

Parameters
Examples
Error messages

The Save/Restore Document Library Object (SAVRSTDLO) command saves and restores the following objects to another system. The system must have a supported communication link with the system that the objects are being restored to.

- Documents
- Folders
- Distribution objects (mail)

Notes:

1. When a folder is saved and restored, the folder object is transferred along with the documents contained in that folder and the subfolders and documents in the subfolders and all successively nested folders and documents. Specific folders can be saved and restored individually using the DLO(*FLRLVL) parameter.
2. Distribution objects (mail) cannot be sent for individual users. Mail can be saved and restored for all users only.
3. SAVRSTDLO does not require a dedicated system; however, individual objects in use when the save and restore command is issued cannot be saved and restored. To ensure all document library objects are saved and restored, run this command when no office activity is occurring on the system.

Restrictions:

1. You must have *ALLOBJ or *SAVSYS special authority to use the following parameter combinations on this command:
 - DLO(*ALL) FLR(*ANY)
 - DLO(*CHG)
 - DLO(*MAIL)
 - DLO(*SEARCH) OWNER(*ALL)
 - DLO(*SEARCH) OWNER(*user-profile-name*)

where the *user profile name* specified is not the *user profile name* of the user issuing the SAVRSTDLO command.

2. Users that do not have *ALLOBJ or *SAVSYS special authority must:
 - Have *ALL authority for each document or folder to be sent.
 - Be enrolled as Document Interchange Architecture (DIA) users.
3. This command cannot be used while another job is running commands such as RCLDLO, SAVDLO, and RSTDLO because exclusive use of internal objects may have been obtained by these commands.
4. Determining document or folder ownership does not include checking group profiles if one is associated with the specified user profile.
5. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
DLO	Document library object	Single values: *ALL, *SEARCH, *CHG, *DOCL, *MAIL, *FLRLVL Other values (up to 300 repetitions): <i>Character value</i>	Required, Positional 1
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 2
FLR	Folder	Single values: *ANY, *NONE Other values (up to 300 repetitions): <i>Character value</i>	Optional
OMITFLR	Folders to omit	Single values: *NONE Other values (up to 300 repetitions): <i>Character value</i>	Optional
SRCHTYPE	Search type	*DOC, *ALL	Optional
CHKEXP	Expiration date	<i>Date</i> , *NO, *CURRENT	Optional
CRTDATE	Creation date	<i>Element list</i>	Optional
	Element 1: Starting time and date	<i>Element list</i>	
	Element 1: Starting time	<i>Time</i> , *AVAIL	
	Element 2: Starting date	<i>Date</i> , *BEGIN, *CURRENT	
	Element 2: Ending time and date	<i>Element list</i>	
	Element 1: Ending time	<i>Time</i> , *AVAIL	
	Element 2: Ending date	<i>Date</i> , *END	
DOCCLS	Document class	<i>Character value</i> , *ANY	Optional
OWNER	Owner profile	<i>Name</i> , *CURRENT, *ALL	Optional
REFCHGDATE	Last changed date	<i>Date</i> , *ANY, *SAVDLOALL	Optional
REFCHGTIME	Last changed time	<i>Time</i> , *ANY	Optional
DOCL	Document list	<i>Character value</i> , *NONE	Optional
TGTRLS	Target release	<i>Simple name</i> , *CURRENT, *PRV	Optional
STG	Storage	*KEEP, *DELETE, *FREE	Optional
CMDCHRID	Command character identifier	Single values: *SYSVAL, *DEV Other values: <i>Element list</i>	Optional
	Element 1: Graphic character set	<i>Integer</i>	
	Element 2: Code page	<i>Integer</i>	
SAVACT	Save active	*NO, *YES	Optional
SAVACTWAIT	Save active wait time	0-99999, 120, *NOMAX	Optional
ASP	Auxiliary storage pool ID	1-32, *ANY	Optional
NEWOBJ	Object name generation	*SAME, *NEW	Optional
ALWOBJDIF	Allow object differences	*NONE, *ALL	Optional
RSTASP	Restore to ASP number	1-32, *SAVASP	Optional

Top

Document library object (DLO)

Specifies the document library objects to save. To save a folder, DLO(*ALL) must be specified.

This is a required parameter.

Single values

***ALL** All document library objects further qualified by the FLR parameter are to be saved. Specifying DLO(*ALL) FLR(*ANY) saves all document library objects.

*SEARCH

All document library objects that meet the specified search values are saved. Search values are specified by using the following parameters:

- **Folder (FLR)** parameter.
- **Check for mark (CHKFORMRK)** parameter.
- **Expiration date (CHKEXP)** parameter.
- **Creation date (CRTDATE)** parameter.
- **Document class (DOCCLS)** parameter.
- **Owner profile (OWNER)** parameter.
- **Last changed date (REFCHGDATE)** parameter.
- **Last changed time (REFCHGTIME)** parameter.

Note: Folders are saved only if SRCTYPE(*ALL) is specified.

***CHG** All documents created or changed and all folders created since the last complete save operation and all mail is saved.

*SYSOBJNAM

The documents with the system object names specified for the **System object name (SYSOBJNAM)** parameter are saved.

*DOCL

The list of documents referred to in a document list specified for the **Document list (DOCL)** parameter is saved.

*MAIL

The distribution objects and documents referred to by a mail log are saved.

*FLRLVL

The folders specified for the **Folder (FLR)** parameter and documents in the folders are saved. Subfolders are not saved.

Other values (up to 300 repetitions)

document-name

Specify the user-assigned names of the documents that are saved. All documents specified must be in the same folder and that folder must be specified for the **Folder (FLR)** parameter.

Note: System object name is not supported.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccc or nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Folder (FLR)

Specifies the name of the folder to save.

Single values

***ANY** Document library objects can be saved from any folder. Consider the following when using the FLR parameter:

- FLR(*ANY) is not valid when one of the following is specified:
 - DLO(*DOCL)
 - DLO(*FLRLVL)
 - DLO(*document-name*)
- FLR(*ANY) is required when one of the following is specified:
 - DLO(*CHG)
 - DLO(*SYSOBJNAM)
 - DLO(*MAIL)
 - DLO(*SEARCH) SRCHTYPE(*ALL)
- When SAVDLO DLO(*ALL) FLR(*ANY) is specified, the following are saved:
 - All documents
 - All folders
 - All distribution objects (mail)

***NONE**

The documents saved are not in any folder. FLR(*NONE) is valid only when one of the following is specified:

- DLO(*ALL)
- DLO(*SEARCH) SRCHTYPE(*DOC)

Other values (up to 300 repetitions)

generic-folder-name

Specify a generic name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all objects with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete object name.

folder-name

Specify the user-assigned name of the folder in which the documents to be saved are located. The folder name can be a maximum of 63 characters in length.

- Folder objects specified here are saved only when DLO(*ALL) or DLO(*FLRLVL) is specified.
- FLR(*folder-name*) is not valid when one of the following is specified:

- DLO(*SYSOBJNAM)
- DLO(*MAIL)
- DLO(*SEARCH) SRCTYPE(*ALL)
- Only one folder name can be specified when one of the following is specified:
 - DLO(*DOCL)
 - DLO(*SEARCH) SRCTYPE(*DOC)
 - DLO(*document-name*)

Note: System object name is not supported.

Top

Folders to omit (OMITFLR)

Specifies the names of one or more folders, or the generic names of each group of folders, to be excluded from the save operation.

Single values

*NONE

No folders are excluded from the save operation.

Other values (up to 300 repetitions)

generic-folder-name

Specify a generic name. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all folders with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete folder name.

folder-name

Specify the name of the folder to be excluded from the save operation.

Note: System object name is not supported.

Top

Search type (SRCTYPE)

Specifies the type of objects for which to search. This parameter is valid only if *SEARCH is specified for the **Document library object (DLO)** parameter.

*DOC Only documents are to be searched and saved.

*ALL Documents and folders are to be searched and saved.

Top

Expiration date (CHKEXP)

Specifies that all documents with an expiration date before the specified date are to be saved. The expiration date is assigned by the user when a document is created to specify when the document is no longer needed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCTYPE)** parameter.

***NO** The expiration date is ignored.

***CURRENT**

All documents with an expiration date before today's date are saved.

date Specify a document expiration date. All documents with an expiration date before this date are saved.

Top

Creation date (CRTDATE)

Specifies that documents and folders that have a creation date during the time period specified are to be saved. The time period is specified by a starting time and date and an ending time and date. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter. The time period is specified as follows:

((start-time start-date) (end-time end-date))

Element 1: Starting time and date

Element 1: Starting time

Use one of the following to specify the starting time. Documents must have been created after this time to be selected. Documents created before this time are not selected.

***AVAIL**

Documents and folders filed at any time are eligible for saving.

time Specify the starting time. When the starting time is used as a search value, the starting date must not be *BEGIN. The starting-time must be the same as the value specified for the **Last changed time (REFCHGTIME)** parameter when the REFCHGTIME parameter is specified. The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Element 2: Starting date

Use one of the following to specify the starting date. Documents must have been created on or after this date to be saved. Documents created before this date are not saved.

***BEGIN**

Documents and folders are saved regardless of the creation date associated with the object.

***CURRENT**

Only documents and folders filed on today's date, after the starting time (if specified), are selected.

date Specify a starting date for the document creation date time period. The date must be

entered in the job date format. The start date must be the same as the value specified for the **Last changed date (REFCHGDATE)** parameter when the REFCHGDATE parameter is specified.

Element 2: Ending time and date

Element 1: Ending time

Use one of the following to specify the ending time. Documents must have been created before this time to be saved. Any documents created after the specified time are not saved.

*AVAIL

Documents and folders filed at any time are selected for saving.

time Specify the ending time. When the ending time is to be used as a search value, the ending date must not be *END. See the description of *starting-time* for details about how time can be specified.

Element 2: Ending date

Use one of the following to specify the ending date. Documents must have been created on or before this date to be saved. Documents created after this date are not saved.

*END Documents and folders filed on any date are selected. The ending time is not allowed when *END is specified.

date Specify the ending date for the document creation date time period. Documents created on or before this date are saved. The date must be specified in job date format.

Top

Document class (DOCCLS)

Specifies the class of documents to be saved. The class is assigned by the user when the document is created. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter and *DOC is specified for the **Search type (SRCHTYPE)** parameter.

Note: Although document classes are user-assigned, double-byte character set (DBCS) data cannot be specified on this parameter.

*ANY The document class is not used to select documents for saving.

character-value

Specify the document class, ranging from 1 through 16 characters, used to select documents for saving.

Top

Owner profile (OWNER)

Specifies the owner of the documents and folders to be saved. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*CURRENT

Documents and folders owned by the current requester are saved.

***ALL** This parameter is not used to select documents and folders for saving. You must have all object (*ALLOBJ) or save system (*SAVSYS) special authority if *ALL is specified.

name Specify the name of the user profile that owns the documents and folders to be saved. All documents and folders owned by this user and that meet the other search values specified are saved. *ALLOBJ or *SAVSYS special authority is required if a user profile is selected other than the user profile of the user issuing this command.

Top

Last changed date (REFCHGDATE)

Specifies the date after which the folders that are created and the documents that are changed or created are to be saved. The change date is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*ANY No reference change date is specified. Documents are saved regardless of the date they were created or changed. Folders are saved regardless of the date they were created.

***SAVDLOALL**

Folders that have been created and documents created or changed since that last complete save operation are saved.

date Specify the date after which the created folders or the created or changed documents are saved.

Top

Last changed time (REFCHGTIME)

Specifies the time, relative to the date specified for the REFCHGDATE parameter, after which the folders that are created and the documents that are changed or created are to be saved. The change time is updated when the document content or description is changed. This parameter is valid only when *SEARCH is specified for the **Document library object (DLO)** parameter.

*ANY No time is specified. The documents are saved regardless of the time they were created or changed. Folders are saved regardless of the time they were created.

time Specify the time after which the created folders or the created or changed documents are saved.

The time can be specified with or without a time separator:

- Without a time separator, specify a string of 4 or 6 digits (hhmm or hhmmss) where hh = hours, mm = minutes, and ss = seconds.
- With a time separator, specify a string of 5 or 8 digits where the time separator specified for your job is used to separate the hours, minutes, and seconds. If you enter this command from the command line, the string must be enclosed in apostrophes. If a time separator other than the separator specified for your job is used, this command will fail.

Top

Document list (DOCL)

Specifies a list of documents to be saved. The document list must be in a folder. The name of the folder must be specified using the **Folder (FLR)** parameter. You must have use (*USE) authority to the folders containing the documents in the document list.

Note: A document list must be the result of a local search, not a remote search.

*NONE

No document list is saved.

character-value

Specify the document list to be saved.

Top

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Storage (STG)

Specifies whether system storage occupied by the document being saved is kept, deleted, or freed after the save operation ends.

Note: STG(*DELETE) and STG(*FREE) are not valid when any of the following are specified:

- DLO(*ALL) FLR(*ANY)
- DLO(*SEARCH) CHKFORMRK(*YES)
- DLO(*CHG)
- DLO(*MAIL)
- SAVACT(*YES)

*KEEP

The storage occupied by the document remains unchanged after the save operation.

***DELETE**

The document object and all search terms are deleted from the system after the save operation.

***FREE** The document description and search terms remain on the system but the storage occupied by the document is deleted after the save operation. The document cannot be used until the document is restored to the system.

Top

Command character identifier (CMDCHRID)

Specifies the character identifier (graphic character set and code page) for the data specified for the **Document class (DOCCLS)** parameter. The character identifier is related to the display device used to enter the document class.

Single values

***SYSVAL**

The system determines the graphic character set and code page values for the command parameters from the QCHRID system value.

***DEVVD**

The system determines the graphic character set and code page values from the display device description where this command was entered. This option is valid only when entered from an interactive job. If this option is specified in a batch job, an error occurs.

Element 1: Graphic character set

1-32767

Specify the graphic character set to use.

Element 2: Code page

1-32767

Specify the code page to use.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

***NO** Document library objects in use are not saved. Document library objects cannot be updated while being used.

***YES** Document library objects can be changed during the save request.

Note: Some applications update document library objects directly. The data is supplied to the application rather than saving the updates in a temporary file and then updating the DLOs. DLOs that are being updated directly (typically, those being updated by PC-based applications) will not be saved. See the Backup and recovery topic collection in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/> for more information on using this parameter.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time for an object that is in use, before continuing the save operation. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object to become available before continuing the save operation.

***NOMAX**
No maximum wait time exists.

0-99999
Specify the number of seconds to wait for each individual object before continuing the save operation.

Top

ASP number (ASP)

Specifies the number of the auxiliary storage pool (ASP) of the document library object (DLO) to be saved.

***ANY** The objects to be saved reside in any ASP. When DLO(*ALL) FLR(*ANY) are specified, all document library objects on the system are saved.

Note:When DLOs from multiple ASPs are saved, multiple tape media files are created. The beginning and ending sequence numbers of these media files will be required on the RSTDLO command to restore all ASPs.

1-32 Specify the number of an existing ASP that contains the document library objects to be saved. ASP 1 is the system ASP.

Note: Mail that has not been filed and documents that are not in a folder reside in the system ASP.

Top

Object name generation (NEWOBJ)

Specifies whether a new library-assigned name and system object name are generated for the folders and documents that are restored.

***SAME**
The library-assigned name and system object name do not change.

***NEW** A new library-assigned name and system object name are generated for each document or folder that is restored.

Top

Allow object differences (ALWOBJDIF)

Specifies whether the following differences encountered during a restore operation are allowed.

- Ownership—the owner of the object on the system is different than the owner of the object from the save operation.
- System object name—the system object name on the system does not match the system object name on the media.

- Authorization list linking—the object is being restored to a system different from the one on which it was saved.

The ALWOBJDIF parameter can be used to allow an object to be restored whose owner or object name on the system is different than on the media used for the restore operation. By specifying the *ALL special value, an object with a different name is restored to the name on the media, while an object with a different owner keeps the owner name from the system instead of the media.

Note: To use this parameter, you need all object (*ALLOBJ) authority.

***NONE**

None of the differences described above are allowed on the restore operation. For authorization list cases, the object is restored, but the object is not linked to the authorization list, and public authority is set to *EXCLUDE. For other cases, a diagnostic message is sent for the object, and the object is not restored.

***ALL** All of the differences described above are allowed for the restore operation. An informational message is sent, and the object is restored.

Notes:

- If the owners of the object do not match, the object is restored, but it will keep the ownership and authorities of the object on the system before the restore operation.
- If *ALL is specified on this parameter, *NEW cannot be specified for the **Object name generation (NEWOBJ)** parameter.
- If you are restoring objects to a system different from the one on which they were saved and the objects are secured by an authorization list, specifying *ALL automatically links the objects to the authorization list. If the authorization list does not exist on the new system, a message that includes the name of the missing list is issued.

Top

Restore to ASP number (RSTASP)

Specifies the number of the auxiliary storage pool (ASP) on media in which restored documents and folders are to be placed.

***SAVASP**

The documents and folders are placed in the same ASP from which they were saved.

1-32 Specify the number of the ASP in which restored documents and folders are placed.

Top

Examples

Example 1: Saving and Restoring All Document Library Objects

```
SAVRSTDLO DLO(*ALL) RMTLOCNAME(SYSTEM1) FLR(*ANY)
          SAVACT(*YES) NEWOBJ(*NEW)
```

This command saves all document library objects located in any folder and restores the objects to a remote system named SYSTEM1. The objects can be changed during the save and restore operation and new names will be given to the libraries in which the objects are restored.

Example 2: Saving and Restoring Specific Document Library Objects

SAVRSTDLO DLO(*SEARCH) RMTLOCNAME(SYSTEM2) FLR(FLR2)
OWNER(USER8) SRCHTYPE(*DOC)

This command searches for a folder named FLR2 and checks to see if the documents owned by user USER8. The folder and the documents in the folder are saved and restored on a remote system named SYSTEM2.

Top

Error messages

*ESCAPE Messages

CPCAD82

&1 document library objects saved and restored.

CPFAD8C

An error occurred during the SAVRSTDLO operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

Top

Save Restore Library (SAVRSTLIB)

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

Parameters
Examples
Error messages

The Save/Restore Library (SAVRSTLIB) command allows the user to save and restore a copy of one or more libraries to another system. The system must have a supported communication link with the restoring system.

Documents and folders contained in the QDOC library can be saved and restored by using the Save/Restore Document Library Object (SAVRSTDLO) command.

The SAVRSTLIB command saves and restores the entire library; this includes the library description, the object descriptions, and the contents of the objects in the library. For job queues, message queues, and logical files, only the object definitions are saved and restored, not the contents. Logical file access paths can be saved and restored by specifying ACCPTH(*YES). The contents of a save file can be saved and restored by specifying SAVFDTA(*YES). The contents of spooled files on output queues can be saved and restored by specifying SPLFDTA(*ALL). The contents of a data queue can be saved and restored by specifying QDTA(*DTAQ).

The libraries and their objects are not affected on the system unless the command specifies that the storage is to be freed.

The types of objects saved and restored by this command are the same as those listed in the OBJTYPE parameter description in "Commonly used parameters: Expanded descriptions" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>, with the addition of *DTADCT. Certain I5/OS system objects that are not contained in user libraries (such as user profiles) are not saved and restored by this command. They can be saved by the Save System (SAVSYS) or Save Security Data (SAVSECDDTA) commands, and restored by using the Restore User Profile (RSTUSRPRF) command.

Restrictions:

1. To use this command, the user must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter, or the user must have:
 - Read authority for, or be the owner of, each library specified.
 - Object existence authority for each object in the library.

If the user does not have the correct authorities for all of the libraries and objects specified, only those for which the user does have authority are saved and restored.

2. No library that is being saved and restored, or the objects in the library being saved and restored, can be updated by a job that is running at the time the save and restore operation occurs unless save-while-active (SAVACT) is used.
3. When the contents of a save file are saved and restored by specifying SAVFDTA(*YES), the save file must be restored before objects contained in it can be restored.
4. Both systems intended to participate in the save and restore operation must be connected to the same APPN network or, if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
LIB	Library	Single values: *NONSYS, *ALLUSR, *IBM Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 2
STRLIB	Starting library	<i>Name, *FIRST</i>	Optional
OMITLIB	Libraries to omit	Single values: *NONE Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *ALL, *NONE</i>	
	Qualifier 2: Library	<i>Generic name, name, *ALL</i>	
	Element 2: Object type	<i>Character value, *ALL, *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNNTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *USRQ, *USRSPC, *VLDL, *WSCST</i>	
ASPDEV	ASP device	<i>Name, *, *SYSBAS, *CURASPGRP</i>	Optional
TGTRLS	Target release	<i>Simple name, *CURRENT, *PRV</i>	Optional
PRECHK	Object pre-check	<i>*NO, *YES</i>	Optional
SAVACT	Save active	<i>*NO, *LIB, *SYSDFN</i>	Optional
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, 120 , *NOMAX	
	Element 2: Pending record changes	0-99999, * LOCKWAIT , *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, * LOCKWAIT , *NOMAX	
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	<i>Name, *NONE, *WRKSTN</i>	
	Qualifier 2: Library	<i>Name, *LIBL, *CURLIB</i>	
ACCPATH	Save access paths	<i>*SYSVAL, *NO, *YES</i>	Optional
SAVFDTA	Save file data	<i>*YES, *NO</i>	Optional
SPLFDTA	Spooled file data	<i>*NONE, *ALL</i>	Optional
QDTA	Queue data	<i>*NONE, *DTAQ</i>	Optional
PVTAUT	Private authorities	<i>*NO, *YES</i>	Optional
STG	Storage	<i>*KEEP, *FREE</i>	Optional
OPTION	Option	<i>*ALL, *NEW, *OLD, *FREE</i>	Optional
MBROPT	Data base member option	<i>*MATCH, *ALL, *NEW, *OLD</i>	Optional
ALWOBJDIF	Allow object differences	Single values: *NONE, *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
RSTLIB	Restore to library	<i>Name, *LIB</i>	Optional

Keyword	Description	Choices	Notes
RSTASPDEV	Restore to ASP device	Name, <u>*SAVASPDEV</u>	Optional
RSTASP	Restore to ASP number	1-32, <u>*SAVASP</u>	Optional
FRCOBJCVN	Force object conversion	Single values: <u>*SYSVAL</u> , *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	<u>*YES</u>	
	Element 2: Objects to convert	<u>*RQD</u> , *ALL	

Top

Library (LIB)

Specifies which libraries to save and restore.

Notes:

1. If using *NONSYS, or *IBM keywords some libraries will not be saved and restored, because they are being used for this operation.
2. If the user specifies *ALLUSR on this parameter, this command should be run when the specified libraries are not being used. If objects in a library are in use while the library is being saved and restored, the objects are not saved and restored. To ensure a complete save and restore of all libraries, run the SAVLIB command with the system in a restricted state. For example, if SAVRSTLIB LIB(*ALLUSR) is run when the subsystem QSNADS is active, the QAO* files are not saved and restored in library QUSRSYS. To save and restore the *QAO files, end the QSNADS subsystem before running SAVRSTLIB LIB(*ALLUSR). Some subsystems cannot be ended when using the SAVRSTLIB command. The libraries associated with these subsystems will not be saved and restored using the SAVRSTLIB command.

QSOC if using the optical bus transport with the SAVRSTLIB command

QCMN

if using the communications transport with the SAVRSTLIB command

*NONSYS

All user-created libraries, the QGPL and QUSRSYS libraries, and licensed program libraries such as QRPGL and QIDU are saved. All subsystems must be ended by the End Subsystem (ENDSBS) or End System (ENDSYS) command before this option is specified. When *NONSYS is specified, the libraries are saved in alphabetical order on the media.

*ALLUSR

All user libraries are saved and restored. All libraries with names that do not begin with the letter Q are saved and restored 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 saved and restored:

```
QDSNX      QRCLxxxxx  QUSRDIRDB  QUSRVI
QGPL       QSRVAGT   QUSRIJS    QUSRVxRxMx
QGPL38     QSYS2     QUSRINFSKR
QMGTC      QSYS2xxxxx QUSRNOTES
QMGTC2     QS36F     QUSROND
```

QMPGDATA	QUSER38	QUSRPOSGS
QMOMDATA	QUSRADSM	QUSRPOSSA
QMOMPROC	QUSRBRM	QUSRPYMSVR
QPFRDATA	QUSRDIRCF	QUSRDRARS
QRCL	QUSRDIRCL	QUSRSYS

Note: 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.

***IBM** Saves all system (IBM) libraries except for the following Q libraries:

QDOC	QRCYxxxxx	QTEMP	QUSRPYMSVR
QDOCxxxx	QRECOVERY	QUSER38	QUSRDRARS
QDSNX	QRPLOBJ	QUSRADSM	QUSRSYS
QGGL	QRPLxxxxx	QUSRBRM	QUSRVI
QGGL38	QSPL	QUSRDIRCF	QUSRVxRxMx
QMGTC	QSPLxxxx	QUSRDIRCL	
QMGTC2	QSRV	QUSRDRDB	
QMPGDATA	QSRVAGT	QUSRIJS	
QMOMDATA	QSYS	QUSRINFSCR	
QMOMPROC	QSYSxxxxx	QUSRNOTES	
QPFRDATA	QSYS2	QUSROND	
QRCL	QSYS2xxxxx	QUSRPOSGS	
QRCLxxxxx	QS36F	QUSRPOSSA	

Note: 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.

The following libraries with names that do not begin with the letter Q are also saved:

#CGULIB	#DSULIB	#SEULIB
#COBLIB	#RPLIB	
#DFULIB	#SDALIB	

generic-name

Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. A maximum of 300 generic library names can be specified.

name Specify the names of the library to be saved and restored. A maximum of 300 library names can be specified.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format cccccc or nnnnnnnn.cccccc, where nnnnnnnn is the network identifier (ID) and cccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Starting library (STRLIB)

Specifies the library with which to begin the *NONSYS, *IBM, or *ALLUSR save operation.

Note: This parameter is valid only if *NONSYS, *IBM, or *ALLUSR is specified on the **Library (LIB)** parameter.

*FIRST

The save operation begins with the first library in alphabetical order.

name Specify the name of the library with which to begin the save operation.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE

No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

***NONE**

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

***ALL** The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

***ALL** All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

***SYSBAS**

The system ASP and all basic user ASPs are included in the save operation.

***CURASPGRP**

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

***PRV** The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Top

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

***NO** The save operation for a library continues, saving only those objects that can be saved.

***YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB** Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYSDFN**

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object lock before continuing the save operation.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value ***NOCMTBDY** is specified.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

***NOCMTBDY**

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify ***NOCMTBDY**.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

*LOCKWAIT

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

*NOMAX

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

*SYSVAL

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Top

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

*NONE

No spooled file data is saved.

***ALL** For each output queue that is saved, all available spooled file data on the output queue is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

*NONE

Only the description of a queue is saved.

*DTAQ

The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save and restore private authorities with the objects that are saved and restored.

*NO No private authorities are saved or restored.

*YES Private authorities are saved and restored with the objects.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority on the system from which objects are being saved, and *ALLOBJ special authority on the restore system, to specify this value.

Top

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

*KEEP

The storage occupied by the data portion of the objects being saved is not freed.

*FREE The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Top

Option (OPTION)

Specifies how to handle restoring each object.

*ALL All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.

*NEW Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are

new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.

***OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.

***FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If ***MATCH** is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when ***ALL** is specified for the **Allow object differences (ALWOBJDIF)** parameter.

***ALL** All members in the saved file are restored.

***NEW** Only new members (members not known to the system) are restored.

***OLD** Only members already known to the system are restored.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (***ALLOBJ**) special authority to specify any value other than ***NONE** for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- Authorization list: The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.

Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.

- File level id: The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.

- Member level id: The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- Ownership: The owner of an object on the system is different than the owner of an object from the save operation.
- Primary Group: The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

*FILELVL

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

*OWNER

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

*PGP

Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Restore to library (RSTLIB)

Specifies whether the library contents are restored to the same library from which they were saved, or to a different library. If a different library is specified, you cannot specify *NONSYS, *ALLUSR, or *IBM for the **Library (LIB)** parameter.

***LIB** The library contents are restored to the same library or libraries from which they were saved.

name Specify the name of the library where the saved library contents are being restored. If *NONSYS, *ALLUSR, or *IBM is specified on the LIB parameter, a library name cannot be specified on this parameter.

Note: If an SQL database is restored to a library other than the one in which it was saved, the journals are not restored.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

***SAVASPDEV**

The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Top

Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Recovering your system book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

***SAVASP**

The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Examples

Example 1: Saving and Restoring All User Libraries

```
SAVRSTLIB  LIB(*ALLUSR)  RMTLOCNAME(SYSTEM1)
           STRLIB(GFM1)  OPTION(*OLD)
```

This command saves all user libraries beginning with the GFM1 library and restores them on a remote system named SYSTEM1. Only the objects in the library having a saved version are restored.

Example 2: Saving and Restoring Specific Libraries

```
SAVRSTLIB LIB(GRUNBOK TIMON VASEK) RMTLOCNAME(SYSTEM1)
```

This command saves the following libraries GRUNBOK, TIMON and VASEK and restores them on a remote system named SYSTEM1.

[Top](#)

Error messages

*ESCAPE Messages

CPCAD81

&1 libraries saved and restored.

CPFAD8B

An error occurred during the SAVRSTLIB operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

[Top](#)

Save Restore Object (SAVRSTOBJ)

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

Parameters
 Examples
 Error messages

The Save/Restore Object (SAVRSTOBJ) command saves and restores a single object, or a group of objects located in the same library, to another system. The objects can be restored to the same library or a different library. The types of objects that can be saved and restored are the same set of objects allowed on the Save Object (SAVOBJ) command. Objects on the system from which the objects are being saved are not affected by the SAVRSTOBJ command.

For job queues, message queues, and logical files, only the object descriptions are saved and restored, not the contents. Logical file access paths can be saved and restored by specifying ACCPTH(*YES). The contents of a save file can be saved and restored by specifying SAVFDTA(*YES). The contents of spooled files on output queues can be saved and restored by specifying SPLFDTA(*ALL). The contents of a data queue can be saved and restored by specifying QDTA(*DTAQ).

Note: This command ignores all file overrides currently in effect for the job.

Restrictions:

1. To use this command, you must have either the special authority *SAVSYS specified in the user profile by the SPCAUT parameter or have (a) object existence authority for each object specified and (b) read authority for the specified library. If you do not have the necessary authority to a specified object, all objects except that one are saved and restored.
2. Both systems intended to participate in the save and restore operation must be connected to the same APPN network, or if the OptiConnect for I5/OS option is to be used, both systems must be joined by the OptiConnect for I5/OS hardware and software.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Objects	Single values: *ALL Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 1
LIB	Saved library	Single values: *ALLUSR Other values (up to 300 repetitions): <i>Generic name, name</i>	Required, Positional 2
RMTLOCNAME	Remote location name	<i>Name</i>	Required, Positional 3
OBJTYPE	Object types	Single values: *ALL Other values (up to 69 repetitions): *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *IMGCLG, *JOB, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PRTIMG, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSFC, *VLDL, *WSCST	Optional

Keyword	Description	Choices	Notes
STRLIB	Starting library	Name, <u>*FIRST</u>	Optional
OMITLIB	Libraries to omit	Single values: <u>*NONE</u> Other values (up to 300 repetitions): <i>Generic name, name</i>	Optional
OMITOBJ	Objects to omit	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: Object	<i>Qualified object name</i>	
	Qualifier 1: Object	<i>Generic name, name, *ALL, *NONE</i>	
	Qualifier 2: Library	<i>Generic name, name, *ALL</i>	
	Element 2: Object type	<i>Character value, *ALL, *ALRTBL, *BNDDIR, *CHTFMT, *CLD, *CLS, *CMD, *CRG, *CRQD, *CSI, *CSPMAP, *CSPTBL, *DTAARA, *DTAQ, *EDTD, *EXITRG, *FCT, *FILE, *FNTRSC, *FNTTBL, *FORMDF, *FTR, *GSS, *IGCDCT, *IGCSRT, *IGCTBL, *JOBQ, *JOBQ, *JOBSCD, *JRN, *JRNRCV, *LOCALE, *MEDDFN, *MENU, *MGTCOL, *MODULE, *MSGF, *MSGQ, *M36, *M36CFG, *NODGRP, *NODL, *ORTBL, *OUTQ, *OVL, *PAGDFN, *PAGSEG, *PDFMAP, *PDG, *PGM, *PNLGRP, *PRDAVL, *PSFCFG, *QMFORM, *QMQR, *QRYDFN, *RCT, *SBSD, *SCHIDX, *SPADCT, *SQLPKG, *SQLUDT, *SRVPGM, *SSND, *SVRSTG, *S36, *TBL, *USRIDX, *TIMZON, *USRQ, *USRSPC, *VLDL, *WSCST</i>	
ASPDEV	ASP device	Name, <u>*</u> , *SYSBAS, *CURASPGRP	Optional
TGTRLS	Target release	Simple name, <u>*CURRENT</u> , *PRV	Optional
PRECHK	Object pre-check	<u>*NO</u> , *YES	Optional
SAVACT	Save active	<u>*NO</u> , *LIB, *SYSDFN	Optional
SAVACTWAIT	Save active wait time	<i>Element list</i>	Optional
	Element 1: Object locks	0-99999, 120 , *NOMAX	
	Element 2: Pending record changes	0-99999, <u>*LOCKWAIT</u> , *NOCMTBDY, *NOMAX	
	Element 3: Other pending changes	0-99999, <u>*LOCKWAIT</u> , *NOMAX	
SAVACTMSGQ	Save active message queue	<i>Qualified object name</i>	Optional
	Qualifier 1: Save active message queue	Name, <u>*NONE</u> , *WRKSTN	
	Qualifier 2: Library	Name, <u>*LIBL</u> , *CURLIB	
FILEMBR	File member	Values (up to 300 repetitions): <i>Element list</i>	Optional
	Element 1: File	Name, <u>*ALL</u>	
	Element 2: Member	Single values: <u>*ALL</u> , *NONE Other values (up to 50 repetitions): <i>Generic name, name</i>	
ACCPH	Save access paths	<u>*SYSVAL</u> , *NO, *YES	Optional
SAVFDTA	Save file data	<u>*YES</u> , *NO	Optional
SPLFDTA	Spooled file data	<u>*NONE</u> , *ALL	Optional
QDTA	Queue data	<u>*NONE</u> , *DTAQ	Optional
PVTAUT	Private authorities	<u>*NO</u> , *YES	Optional
STG	Storage	<u>*KEEP</u> , *FREE	Optional
OPTION	Option	<u>*ALL</u> , *NEW, *OLD, *FREE	Optional
MBROPT	Data base member option	<u>*MATCH</u> , *ALL, *NEW, *OLD	Optional
ALWOBJDIF	Allow object differences	Single values: <u>*NONE</u> , *ALL Other values (up to 4 repetitions): *AUTL, *FILELVL, *OWNER, *PGP	Optional
RSTLIB	Restore to library	Name, <u>*LIB</u>	Optional
RSTASPDEV	Restore to ASP device	Name, <u>*SAVASPDEV</u>	Optional
RSTASP	Restore to ASP number	1-32, <u>*SAVASP</u>	Optional

Keyword	Description	Choices	Notes
FRCOBJCVN	Force object conversion	Single values: *SYSVAL, *NO Other values: <i>Element list</i>	Optional
	Element 1: Convert during restore	*YES	
	Element 2: Objects to convert	*RQD, *ALL	

Top

Objects (OBJ)

Specifies the names of one or more objects or the generic name of each group of objects to be saved. All the objects must be in the library specified for the **Library (LIB)** parameter. If *ALL is specified or defaulted for the **Object types (OBJTYPE)** parameter, all the object types listed in the description of that parameter are saved, provided they are in the specified library and have the specified names.

This is a required parameter.

Single values

***ALL** All the objects in the specified libraries are saved, depending on the values specified for the OBJTYPE parameter.

Other values (up to 300 repetitions)

generic-name

Specify one or more generic names of groups of objects in the specified library to be saved. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete object name.

name Specify one or more names of specific objects to be saved. Both generic names and specific names can be specified in the same command.

Top

Saved library (LIB)

Specifies the library containing the objects to be saved and restored.

Single values

*ALLUSR

All user libraries are saved and restored. All libraries with names that do not begin with the letter Q are saved and restored 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 saved and restored:

QDSNX	QRCLxxxxx	QUSRDIRDB	QUSRVI
QGGL	QSRVAGT	QUSRIJS	QUSRVxRxMx
QGGL38	QSYS2	QUSRINFSKR	
QMGTC	QSYS2xxxxx	QUSRNOTES	
QMGTC2	QS36F	QUSROND	
QMPGDATA	QUSER38	QUSRPOSGS	
QMOMDATA	QUSRADSM	QUSRPOSSA	
QMOMPROC	QUSRBRM	QUSRPYMSVR	
QPFRDATA	QUSRDIRCF	QUSRDRARS	
QRCL	QUSRDIRCL	QUSRSYS	

Note: 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.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the library. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name. Up to 300 generic library values can be specified.

name Specify the name of the library to be saved and restored. Up to 300 library names can be specified.

Top

Remote location (RMTLOCNAME)

Specifies the remote location to connect with. Specify the remote location name using the format ccccccc or nnnnnnnn.ccccccc, where nnnnnnnn is the network identifier (ID) and ccccccc is the remote location name.

remote-location-name

Specify the remote location name associated with the system to which you want to restore objects. The local network ID (LCLNETID) network attribute is used as the value of the network identifier.

network-ID.location-name

Specify the network identifier and the remote location name associated with the system to which you want to restore objects.

Top

Object types (OBJTYPE)

Specifies the types of system objects to be saved.

Single values

***ALL** All object types that are specified by name and are in the specified library are saved. If *ALL is also specified for the **Objects (OBJ)** parameter, all the objects in the library that are of the types that can be saved are saved.

Other values (up to 300 repetitions)

object-type

Specify the value for each of the types of objects to be saved, such as command (*CMD), file (*FILE), or program (*PGM).

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

Top

Starting library (STRLIB)

Specifies the library with which to begin the save operation.

If an unrecoverable media error occurs during the save operation, this parameter can be used to restart the operation.

The basic steps for restarting a save operation are:

1. Check the job log to determine the library where the previous save operation failed. Find the last library saved, which is indicated by a successful completion message.
2. Load the next tape and ensure the tape is initialized.
3. Add the following to your original save command:

```
STRLIB(library-name) OMITLIB(library-name)
```

where the *library-name* for the STRLIB and OMITLIB parameters is the last library successfully saved. This starts the save operation on the library after the last successfully saved library.

To restore the libraries, you will need to perform a separate restore operation for each save operation that was performed.

*FIRST

The save operation begins with the first library value specified for the **Library (LIB)** parameter. If the first value is a generic name or special value, the save operation begins with the first library that matches this value.

name Specify the name of the library with which to begin the save operation.

Top

Libraries to omit (OMITLIB)

Specifies the names of one or more libraries, or the generic names of each group of libraries, to be excluded from the save operation.

Single values

*NONE

No libraries are excluded from the save operation.

Other values (up to 300 repetitions)

generic-name

Specify the generic name of the libraries to be excluded. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. The asterisk (*) substitutes

for any valid characters. A generic name specifies all libraries with names that begin with the generic prefix, for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete library name.

name Specify the name of the library to be excluded from the save operation.

Top

Objects to omit (OMITOBJ)

Specifies the objects to be excluded from the operation. Up to 300 objects or generic object values can be specified.

Element 1: Object

Qualifier 1: Object

*NONE

No objects are excluded from the operation.

***ALL** All objects of the specified object type are excluded from the operation.

generic-name

Specify the generic name of the objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the object to be excluded from the operation.

Qualifier 2: Library

*ALL The specified objects are excluded from all libraries that are part of the operation.

generic-name

Specify the generic name of the libraries that contain objects to be excluded.

Note: A generic name is specified as a character string that contains one or more characters followed by an asterisk (*). If a generic name is specified, then all objects that have names with the same prefix as the generic object name are selected.

name Specify the name of the library that contains the object to be excluded from the operation.

Element 2: Object type

*ALL All object types are excluded from the operation, depending on the value specified for the object name.

character-value

Specify the object type of the objects to be excluded from the operation.

To see a complete list of object types when prompting this command, position the cursor on the field for this parameter and press F4 (Prompt). For a description of the object types, see "Object types" in the CL topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

ASP device (ASPDEV)

Specifies the auxiliary storage pool (ASP) device to be included in the save operation. This parameter is used to subset the list of objects which qualify for the SAV based on the OBJ parameter.

*
- The operation includes the system ASP (ASP number 1), all basic user ASPs (ASP numbers 2-32), and, if the current thread has an ASP group, all independent ASPs in the ASP group.

*SYSBAS

The system ASP and all basic user ASPs are included in the save operation.

*CURASPGRP

If the current thread has an ASP group, all independent ASPs in the ASP group are included in the save operation.

name Specify the name of the ASP device to be included in the save operation.

Target release (TGTRLS)

Specifies the release level of the operating system on which you intend to use the object being saved.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

*CURRENT

The object is to be restored to, and used on, the release of the operating system currently running on your system. The object can also be restored to a system with any subsequent release of the operating system installed.

*PRV The object is to be restored to the previous release with modification level 0 of the operating system. The object can also be restored to a system with any subsequent release of the operating system installed.

character-value

Specify the release in the format VxRxMx. The object can be restored to a system with the specified release or with any subsequent release of the operating system installed.

Object pre-check (PRECHK)

Specifies whether the save operation for a library ends if any of the following are true:

1. The objects do not exist
2. The library or the objects were previously found to be damaged
3. The library or the objects are locked by another job
4. The requester of the save operation does not have authority for the library or to save the objects.

*NO The save operation for a library continues, saving only those objects that can be saved.

***YES** If, after all specified objects are checked, one or more objects cannot be saved, the save operation for a library ends before any data is written. If multiple libraries are specified, the save operation continues with the next library. However, if PRECHK(*YES) and SAVACT(*SYNCLIB) are specified and an object in any library to be saved does not meet the preliminary check conditions, the save operation ends and no objects are saved.

Top

Save active (SAVACT)

Specifies whether an object can be updated while it is being saved.

Note: If your system is in a restricted state and the SAVACT parameter is specified, the save operation is performed as if SAVACT(*NO) was specified.

***NO** Objects that are in use are not saved. Objects cannot be updated while being saved.

***LIB** Objects in a library can be saved while they are in use by another job. All the objects in a library reach a checkpoint together and are saved in a consistent state in relationship to each other.

***SYSDFN**

Objects in a library can be saved while they are in use by another job. Objects in a library may reach checkpoints at different times and may not be in a consistent state in relationship to each other.

Top

Save active wait time (SAVACTWAIT)

Specifies the amount of time to wait for an object that is in use, or for transactions with pending changes to reach a commit boundary, before continuing the save operation.

Element 1: Object locks

For each object that is in use, specifies the amount of time to wait for the object to become available. If an object remains in use for the specified time, the object is not saved.

120 The system waits up to 120 seconds for each individual object lock before continuing the save operation.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for each individual object lock before continuing the save operation.

Element 2: Pending record changes

For each group of objects that are checkpointed together, specifies the amount of time to wait for transactions with pending record changes to reach a commit boundary. The **Save active (SAVACT)** parameter determines which objects are checkpointed together. If 0 is specified, all objects being saved must be at commit boundaries. If any other value is specified, all objects that are journaled to the same journals as the objects being saved must reach commit boundaries. If a commit boundary is not reached in the specified time, the save operation is ended, unless the value *NOCMTBDY is specified.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for transactions with pending record changes to reach a commit boundary.

***NOCMTBDY**

The system will save objects without requiring transactions with pending record changes to reach a commit boundary. Therefore, objects may be saved with partial transactions.

If you restore an object that was saved with partial transactions, you cannot use the object until you apply or remove journal changes (APYJRNCHG or RMVJRNCHG command) to reach commit boundaries. You will need all journal receivers that contain information about the partial transactions to apply or remove the changes. Until you apply or remove the changes, any future save of that object will include the partial transactions, even if you do not specify *NOCMTBDY.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for transactions with pending record changes to reach a commit boundary.

Element 3: Other pending changes

For each library, specifies the amount of time to wait for transactions with other pending changes to reach a commit boundary. Other pending changes include the following:

- Data Definition Language (DDL) object level changes for that library.
- Any API commitment resource that was added without the option to allow normal save processing. For more information, see the Add Commitment Resource (QTNADDCR) API in the APIs topic collection in the Programming category in the i5/OS Information Center at <http://www.ibm.com/systems/i/infocenter/>.

If a commit boundary is not reached for a library in the specified time, the library is not saved.

***LOCKWAIT**

The system waits up to the value specified for Element 1 for the types of transactions that are listed above to reach a commit boundary.

***NOMAX**

No maximum wait time exists.

0-99999

Specify the number of seconds to wait for the types of transactions that are listed above to reach a commit boundary.

If 0 is specified, and only one name is specified for the **Objects (OBJ)** parameter, and *FILE is the only value specified for the **Object types (OBJTYPE)** parameter, the system will save the object without requiring the types of transactions that are listed above to reach a commit boundary.

Top

Save active message queue (SAVACTMSGQ)

Specifies the message queue that the save operation uses to notify the user that the checkpoint processing for a library is complete. A separate message is sent for each library to be saved when the *SYSDFN or *LIB value is specified for the **Save active (SAVACT)** parameter. When *SYNCLIB is specified for the SAVACT parameter, one message is sent for all libraries in the save operation.

This parameter can be used to save the objects at a known, consistent boundary to avoid additional recovery procedures following a restore operation. Applications can be stopped until the checkpoint processing complete message is received.

Single values

*NONE

No notification message is sent.

*WRKSTN

The notification message is sent to the work station message queue. This value is not valid in batch mode.

Qualifier 1: Save active message queue

name Specify the name of the message queue to be used.

Qualifier 2: Library

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

*CURLIB

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

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

Top

File member (FILEMBR)

Specifies the database file members that are saved. This parameter is made up of two parts: the file name and the member name.

Each database file specified here must also be specified for the **Objects (OBJ)** parameter, by its complete name, a generic name, or *ALL. The **Object types (OBJTYPE)** parameter value must be *ALL or include *FILE.

Note: This parameter cannot be specified when STG(*FREE) is specified.

Element 1: File

*ALL The list of member name values that follows this value applies to all files specified for the OBJ parameter.

name Specify the name of the database file from which the listed members are to be saved. Up to 50 files can be specified, with a member list for each file.

Note: Generic names are not valid for the database file name, but are allowed for the member name.

Note: Duplicate file names are not allowed.

Element 2: Member

Single values

*ALL All members are saved from the specified file.

*NONE

No members are saved from the specified file. Only the file description is saved.

Other values (up to 50 repetitions)

generic-name

Specify the generic names of the members to be saved from the specified file. A generic name is a character string that contains one or more characters followed by an asterisk (*). If an * is not specified with the name, the system assumes that the name is a complete member name.

Note: If generic member names are specified, the file must contain member names that match the generic names for the file to be saved. For example, if PAY* is specified as a generic member name, and the system is unable to find a member whose name starts with PAY, the file is not saved. If files specified by the FILEMBR parameter are not saved because members with the specified generic name cannot be found, a diagnostic message is sent, the save operation ends, and an escape message is sent specifying the number of files not saved. If at least one of the files processed for the FILEMBR parameter contains a member with the specified generic name, the diagnostic message is not sent, and the number of files not saved is in the final completion message.

name Specify the names of the members to be saved from the given file.

Note: If specific member names are specified, the specified members must exist in the file for any part of the file to be saved or restored.

Top

Save access paths (ACCPH)

Specifies whether the logical file access paths that are dependent on the physical files being saved are also saved. The access paths are saved only in the case of the following:

- All members on which the access paths are built are included in this save operation.
- The access paths are not invalid or damaged at the time of the save.

The system checks to ensure the integrity of the access paths. Any discrepancies found by the system will result in the access paths being rebuilt.

Informational messages are sent indicating the number of logical file access paths saved with each physical file. All physical files on which an access path is built must be in the same library. This parameter does not save logical file objects; it only controls the saving of the access paths. More information on the restoring of saved access paths is in the Recovering your system book, SC41-5304.

ATTENTION: If the based-on physical files and the logical files are in different libraries, the access paths are saved. However, if the logical files and the based-on physical files are in different libraries and the logical files or physical files do not exist at restore time (such as during disaster recovery or the files were deleted) the access paths are not restored. They are rebuilt. For the fastest possible restore operation for logical files, the logical files and the based-on physical files must be in the same library and must be saved at the same time.

***SYSVAL**

The QSAVACCPH system value determines whether to save the logical file access paths that are dependent on the physical files that are being saved.

***NO** Only those objects specified on the command are saved. No logical file access paths are saved.

***YES** The specified physical files and all eligible logical file access paths over them are saved.

Note: Specifying this value does not save the logical files.

Top

Save file data (SAVFDTA)

Specifies, for save file objects, whether the description of a save file, or both the description and the contents of a save file, are saved.

***YES** The description and contents of a save file are saved.

***NO** Only the description of a save file is saved.

Top

Spooled file data (SPLFDTA)

Specifies whether to save spooled file data and attributes for output queues that are saved.

***NONE**
No spooled file data is saved.

***ALL** For each output queue that is saved, all available spooled file data on the output queue is saved.

Top

Queue data (QDTA)

Specifies, for queue objects, whether the description of a queue, or both the description and the contents of a queue, are saved.

***NONE**
Only the description of a queue is saved.

***DTAQ**
The description and contents of a standard data queue are saved. Only the description of a Distributed Data Management (DDM) data queue is saved.

Top

Private authorities (PVTAUT)

Specifies whether to save and restore private authorities with the objects that are saved and restored.

***NO** No private authorities are saved or restored.

***YES** Private authorities are saved and restored with the objects.

Note: You must have save system (*SAVSYS) or all object (*ALLOBJ) special authority on the system from which objects are being saved, and *ALLOBJ special authority on the restore system, to specify this value.

Top

Storage (STG)

Specifies whether the system storage that is occupied by the data portion of the specified members (except for save files), modules, programs, service programs, Structured Query Language (SQL) packages, and journal receivers in the library being saved is freed as part of the save operation. Only the data portion of the objects is freed, not the descriptions of the objects.

***KEEP**
The storage occupied by the data portion of the objects being saved is not freed.

***FREE** The storage occupied by the data portion of the specified objects being saved is freed as part of the save operation. The storage for all the objects in a library is freed only after all the objects in that library are saved successfully.

Note: To prevent the possible abnormal end of a program, the program being saved must not be running in the system when *FREE is specified.

Top

Option (OPTION)

Specifies how to handle restoring each object.

- *ALL** All the objects in the saved library are restored to the library. Objects in the saved library replace the current versions in the system library. Objects not having a current version are added to the system library. Objects presently in the library, but not on the media, remain in the library.
- *NEW** Only the objects in the saved library that do not exist in the current version of the system library are added to the library. Only objects not known to the system library are restored; known objects are not restored. This option restores objects that were deleted after they were saved or that are new to this library. If any saved objects have a version already in the system library, they are not restored, and an informational message is sent for each one, but the restore operation continues.
- *OLD** Only the objects in the library having a saved version are restored; that is, the version of each object currently in the library is replaced by the saved version. Only objects known to the library are restored. If any saved objects are no longer part of the online version of the library, they are not added to the library; an informational message is sent for each one, but the restore continues.
- *FREE** The saved objects are restored only if they exist in the system library with their space freed. The saved version of each object is restored on the system in its previously freed space. This option restores objects that had their space freed when they were saved. If any saved objects are no longer part of the current version of the library, or if the space is not free for any object, the object is not restored and an informational message is sent for each one. The restore operation continues, and all of the freed objects are restored.

Top

Data base member option (MBROPT)

Specifies, for database files that exist on the system, which members are restored. If *MATCH is used, the member list in the saved file must match, member for member, the current version on the system. All members are restored for files that do not exist, if the file is restored.

*MATCH

The saved members are restored if the lists of the members where they exist match, member for member, the lists of the current system version. MBROPT(*MATCH) is not valid when *ALL is specified for the **Allow object differences (ALWOBJDIF)** parameter.

- *ALL** All members in the saved file are restored.
- *NEW** Only new members (members not known to the system) are restored.
- *OLD** Only members already known to the system are restored.

Top

Allow object differences (ALWOBJDIF)

Specifies whether differences are allowed between the saved objects and the restored objects.

Notes:

1. You must have all object (*ALLOBJ) special authority to specify any value other than *NONE for this parameter.
2. If differences are found, the final message for the restore operation is an escape message rather than the normal completion message.

The types of differences include:

- **Authorization list:** The saved object had an authorization list, and either the object exists on the system but does not have the same authorization list, or the object does not exist and it is being restored to a different system than the save system.
Note: This parameter has no effect when the saved object did not have an authorization list. If the object exists, it is restored with the authorization list of the existing object. If it does not exist, it is restored with no authorization list.
- **File level id:** The creation date and time of the database file on the system does not match the creation date and time of the file that was saved.
- **Member level id:** The creation date and time of the database file member on the system does not match the creation date and time of the member that was saved.
- **Ownership:** The owner of an object on the system is different than the owner of an object from the save operation.
- **Primary Group:** The primary group of an object on the system is different than the primary group of an object from the save operation.

Single values

*NONE

None of the differences listed above are allowed on the restore operation. See the description of each individual value to determine how differences are handled.

- ***ALL** All of the differences listed above are allowed on the restore operation. File level id and member level id differences are handled differently than the *FILELVL value. If there is a file level difference and *ALL is specified for the **Data base member option (MBROPT)** parameter, the existing version of the file is renamed and the saved version of the file is restored. If there is a member level difference, the existing version of the member is renamed and the saved version of the member is restored. This value will restore the saved data, but the result may not be correct. You will need to choose whether the restored data or the renamed data is correct, and you will need to make the necessary corrections to the database. For other differences, see the description of each individual value to determine how differences are handled.

Other values (up to 4 repetitions)

*AUTL

Authorization list differences are allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is restored with the authorization list of the existing object. If the saved object had an authorization list and the object does not exist and it is being restored to a different system than the save system, the object is restored and it is linked to the authorization list. If the authorization list does not exist, the public authority is set to *EXCLUDE.

If this value is not specified, authorization list differences are not allowed. If the saved object had an authorization list and the object exists on the system but does not have the same authorization list, the object is not restored. If the saved object had an authorization list and the object does not

exist and it is being restored to a different system than the save system, the object is restored, but it is not linked to the authorization list, and the public authority is set to *EXCLUDE.

***FILELVL**

File level id and member level id differences are allowed. An attempt will be made to restore existing physical files even though the physical file on the save media may have a different file level id or member level id than the physical file on the system. The physical file data will only be restored for those physical files whose format level identifiers on the save media match the format level identifiers of the corresponding physical file on the system.

If this value is not specified, file level id and member level id differences are not allowed. If an object already exists on the system with a different file level id or member level id than the saved object, the object is not restored.

***OWNER**

Ownership differences are allowed. If an object already exists on the system with a different owner than the saved object, the object is restored with the owner of the object on the system.

If this value is not specified, ownership differences are not allowed. If an object already exists on the system with a different owner than the saved object, the object is not restored.

***PGP** Primary group differences are allowed. If an object already exists on the system with a different primary group than the saved object, the object is restored with the primary group of the object on the system.

If this value is not specified, primary group differences are not allowed. If an object already exists on the system with a different primary group than the saved object, the object is not restored.

Top

Restore to library (RSTLIB)

Specifies the library in which the objects are to be restored.

***LIB** The name of the library that was specified on the LIB parameter is used.

name Specify the name of the library to which you want to restore the objects.

Top

Restore to ASP device (RSTASPDEV)

Specifies the auxiliary storage pool (ASP) device to which the data is to be restored.

Note: You can specify either the RSTASPDEV parameter or the RSTASP parameter, but not both.

***SAVASPDEV**

The data is restored to the same ASP from which it was saved.

name Specify the name of the ASP device to be used.

Top

Restore to ASP number (RSTASP)

Specifies whether objects are restored to the auxiliary storage pool (ASP) from which they were saved or to the system ASP (ASP number 1) or to a basic user ASP (ASP numbers 2 through 32).

Some objects cannot be restored to user ASPs. More information about object types which can be restored to user ASPs is in the Recovering your system book, SC41-5304. If the library exists in, or is being restored to the system ASP, journals, journal receivers, and save files can be restored to basic user ASPs. All other object types will be restored to the ASP of the library.

ATTENTION: System or product libraries (libraries that begin with a Q or #) must not be created in or restored to a user ASP. Doing so can cause unpredictable results.

*SAVASP

The objects are restored to the ASP from which they were saved.

1-32 Specifies the ASP number. When the specified ASP is 1, the specified objects are restored to the system ASP, and when the specified ASP is 2 through 32, the objects are restored to the basic user ASP specified.

Top

Force object conversion (FRCOBJCVN)

Specifies whether to convert user objects to the format required for use in the current version of the operating system, or to be compatible with the current machine, when the objects are restored.

Notes:

1. This parameter applies only to user objects of the *MODULE, *PGM, *SRVPGM, and *SQLPKG object types.
2. An object must have creation data (either observable or unobservable) to be converted.
3. If an object needs to be converted (because it is formatted for an earlier version of the operating system or is incompatible with the current machine), but is not converted during this restore operation, the object is automatically converted the first time it is used.

Single values

*SYSVAL

The objects are converted based on the value of the QFRCCVNRST system value.

***NO** The objects are not converted during the restore operation.

Note: If FRCOBJCVN(*NO) is specified, then the QFRCCVNRST system value must have a value of either "0" or "1".

Element 1: Convert during restore

***YES** The objects are converted during the restore operation.

Notes:

1. If FRCOBJCVN(*YES *RQD) is specified, then the QFRCCVNRST system value must have a value of "0", "1", or "2". FRCOBJCVN(*YES *RQD) will override a QFRCCVNRST value of "0" or "1". If FRCOBJCVN(*YES *ALL) is specified, then QFRCCVNRST can have any valid value and FRCOBJCVN(*YES *ALL) overrides the QFRCCVNRST system value.
2. Specifying this value increases the time of the restore operation, but avoids the need to convert the objects when they are first used.

Element 2: Objects to convert

***RQD** The objects are converted only if they require conversion to be used by the current operating system or to be compatible with the current machine. If the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

***ALL** All objects are converted regardless of their current format and machine compatibility, including

compatible objects already in the current format. However, if the objects do not have all creation data (either observable or unobservable), the objects cannot be converted and will not be restored.

Top

Examples

Example 1: Saving and Restoring Generic Objects

```
SAVRSTOBJ OBJ(ABCD*) LIB(ACE) RMTLOCNAME(SYSTEM1)
          OBJTYPE(*PGM) ALWOBJDIF(*NONE)
```

This command saves the objects beginning with the characters ABCD located in the library named ACE and restores them on the remote system named SYSTEM1.

Example 2: Saving and Restoring a Specific Object

```
SAVRSTOBJ OBJ(FLETCH) LIB(CHASE) RMTLOCNAME(SYSTEM1)
          ALWOBJDIF(*ALL) OPTION(*NEW)
```

This command saves the object named FLETCH located in the library named CHASE and restores it on the remote system named SYSTEM1 if it does not already exist on the remote system.

Top

Error messages

*ESCAPE Messages

CPCAD80

&1 objects saved and restored.

CPFAD8D

An error occurred during the &1 operation.

CPFAD80

Unable to establish connection from &1 to &2.

CPFAD81

User profile &1 not found on remote location &2.

CPFAD82

Remote location &1 not found.

CPFAD83

Remote location &1 cannot be source location.

CPFAD84

ObjectConnect internal error, function code &1, return code &2.

CPFAD86

Location name &1 unable to close &2.

CPFAD88

Unable to establish connection from &1 to &2.

CPFAD93

APPC failure. Failure code is &3.

CPF389C

ObjectConnect internal error, function code &1, return code &2.

[Top](#)

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
Distributed Relational Database Architecture
Domino
DRDA
IBM
Infoprint
InfoWindow
i5/OS
iSeries
Integrated Language Environment
Lotus
LPDA
OfficeVision
Print Services Facility
RPG/400
System i
System x
SystemView
System/36
TCS
Tivoli
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.



Printed in USA